

**WT1501, WT1502,
WT1503, WT1504
Precision Power Analyzer
Communication Interface**

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

Thank you for purchasing the WT1501, WT1502, WT1503, or WT1504 Precision Power Analyzer.

This Communication Interface User's Manual explains the following interface features and commands.

- Ethernet interface
- USB interface (wireless LAN adapter)
- USB interface (USB communication)
- Communication commands

To ensure correct use, please read this manual thoroughly before operation.

After reading this manual, keep it in a safe place.

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

Document No.	Error Message
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.

Trademarks

- Microsoft, MS-DOS, Windows, and Windows 11 are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are either registered trademarks or trademarks of Adobe Inc.
- 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.

Ethernet Interface and USB Interface

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

- 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/library/>

To use a wireless network, you need to connect a wireless LAN adapter to this instrument. Please use your own wireless LAN adapter.

For the recommended wireless LAN adapters that can be used with this instrument, contact your nearest YOKOGAWA dealer.

Revisions

- 1st Edition: July 2026

How to Use This Manual

Structure of the Manual

This user manual consists of the following chapters: Chapter 1 through Chapter 7, as well as the appendix.

Chapter 1 Ethernet interface

Describes the features and specifications of the Ethernet interface.

Chapter 2 USB Interface (When Using a Wireless LAN Adapter)

Describes the function of connecting a wireless LAN adapter to this instrument and controlling it via WLAN from a terminal.

Chapter 3 USB Interface (USB Communication)

Describes the functions and specifications of the USB interface for controlling this instrument from a PC.

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.

Appendix

Describes error messages and provides other information.

Index

Conventions Used in This Manual

Prefixes k and K

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

- k** Denotes 1000. Example: 100 kHz
- K** Denotes 1024. Example: 720 KB (file size)

Displayed characters

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

Notes

The notes in this manual are indicated with the following symbols.

Note Provides important information for handling this instrument.

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	Meaning	Example	Example of Input
< >	A defined value	ELEMeNt<x> <x> = 1 to 4	ELEMENT2
{ }	Select an option in { }	SQFormula {TYPE1 TYPE2 TYPE3}	SQFORMULA TYPE1
	Exclusive OR		
[]	Can be omitted	NUMeric[:NORMal]:VALue?	NUMERIC:VALUE?

Contents

How to Use This Manual	iii
Conventions Used in This Manual	iv
Chapter 1 Ethernet Interface	
1.1 Part Names and Functions	1-1
1.2 Ethernet Interface Functions and Specifications	1-2
1.3 Connection through the Ethernet Interface	1-4
1.4 Configuring the Instrument's Ethernet Settings	1-5
Chapter 2 USB Interface (when using a wireless LAN adapter)	
2.1 Part Names and Functions	2-1
2.2 USB Interface (when using a wireless LAN adapter) Functions and Specifications	2-2
2.3 Connecting the Wireless LAN Adapter	2-3
2.4 Configuring the Instrument (when using a wireless LAN adapter)	2-4
Chapter 3 USB Interface	
3.1 Part Names and Functions	3-1
3.2 USB Interface Functions and Specifications	3-2
3.3 Connection through the USB Interface	3-3
3.4 Configuring the Instrument's USB Settings	3-4
Chapter 4 Before Programming	
4.1 Messages	4-1
4.2 Commands	4-4
4.3 Responses	4-7
4.4 Data	4-8
4.5 Synchronizing with the Controller	4-11
Chapter 5 Commands	
5.1 List of Commands	5-1
5.2 AUX Group	5-23
5.3 CAN Group	5-26
5.4 COMMunicate Group	5-29
5.5 CURSor Group	5-31
5.6 DISPlay Group	5-33
• Function options (<Function>) list	5-45
5.7 FILE Group	5-50
5.8 HARMonics Group	5-54
5.9 HISTory Group	5-56
5.10 HOLD Group	5-58
5.11 IMAGe Group	5-59
5.12 INPut Group	5-61
5.13 INTEGrate Group	5-84
5.14 MEASure Group	5-87
5.15 MOTor Group	5-94
5.16 NUMeric Group	5-104
• Numeric data format	5-109
• List of preset numeric data output items	5-111
5.17 STATus Group	5-114

Contents

5.18	STORe Group	5-115
5.19	SYSTem Group	5-119
5.20	UDP Group.....	5-126
5.21	UPDate Group.....	5-127
5.22	WAVEform Group	5-129
5.23	Common Command Group	5-131

Chapter 6 Status Reports

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

Chapter 7 Modbus/TCP Communication

7.1	Overview of Modbus/TCP Communication	7-1
7.2	Communication with Client Devices	7-2
7.3	Register Functions and Uses	7-3

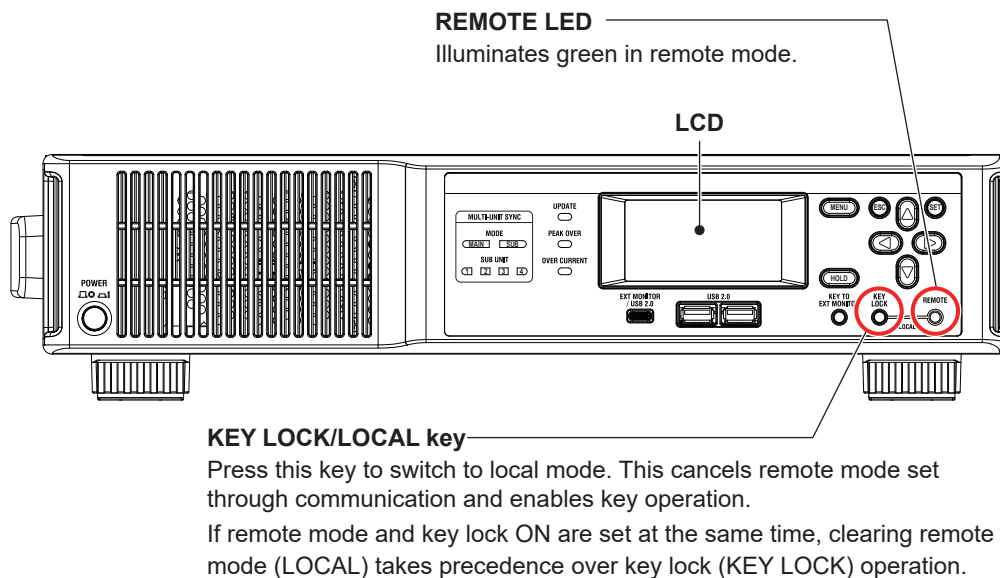
Appendix

Appendix 1	Error Messages	App-1
Appendix 2	History Function	App-4
Appendix 3	Port Numbers	App-8

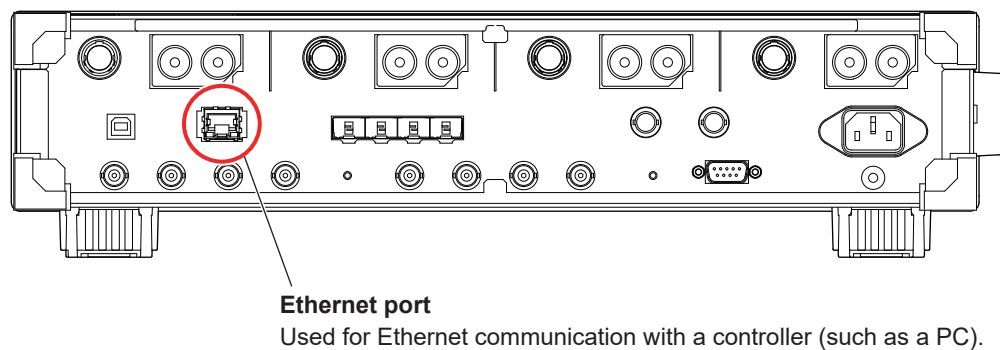
Index

1.1 Part Names and Functions

Front panel



Rear panel



1.2 Ethernet Interface Functions and Specifications

Ethernet interface functions

Receive functions

You can make the same settings as those made through key operation when an external monitor or Web server is connected.

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

Send functions

The instrument can output measured and computed data.

It can output panel setup parameters and status bytes.

It can output error codes that have occurred.

Ethernet interface specifications

Electrical and mechanical specifications	IEEE 802.3 compliant, Auto-MDIX
Simultaneous connections	1
Communication protocol	TCP/IP (VXI-11), HiSLIP, RawSocket*
Connector type	RJ-45

* The port number for Raw Socket is 10002, and the terminator is LF (0Ah).

Note

You cannot use the Ethernet interface simultaneously with other interfaces (USB interface).

Switching between remote and local modes

Switching from local to remote mode

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

- The REMOTE LED lights green.
- All keys except the **KEY LOCK** 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 **KEY LOCK**, 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 LED turns off.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

If remote mode and key lock ON are set at the same time, clearing remote mode (LOCAL) takes precedence over key lock (KEY LOCK) operation.

- On the first KEY LOCK operation, remote mode is cleared, and the REMOTE LED turns off.
- On the second KEY LOCK operation, key lock turns OFF, and the KEY LOCK LED turns off.

RawSocket communication sample program

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

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

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

    /* sockaddr_in structure */
    struct sockaddr_in dstAddr;

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

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

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

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

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

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

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

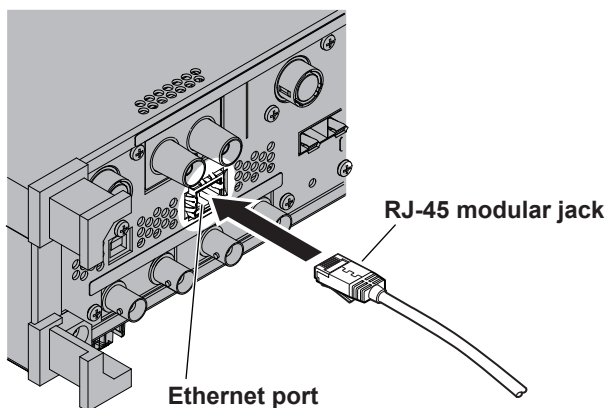
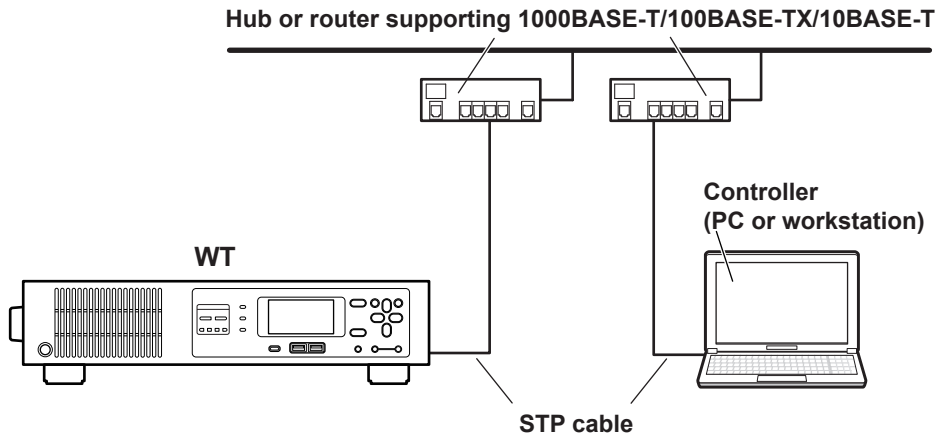
    /* Windows Socket API closure */
    closesocket(dstSocket);
    WSACleanup();

    return 0;
}
```

1.3 Connection through the Ethernet Interface

Connection procedure

Connect the STP (Shielded Twisted-Pair) cable connected to a hub or similar device to the Ethernet port on the rear panel of this instrument.



Notes on connection

- When connecting this instrument to a PC, be sure to use a straight-through cable through a hub or router. Operation is not guaranteed for a one-to-one connection using a crossover cable.
- Use a cable that is compatible with your network environment (transmission speed).

Note

For details on how to connect the instrument to a network, see section 14.1, "Connecting the Instrument to a Network" of the *User's Manual* (IM WT1501-01EN).

1.4 Configuring the Instrument's Ethernet Settings

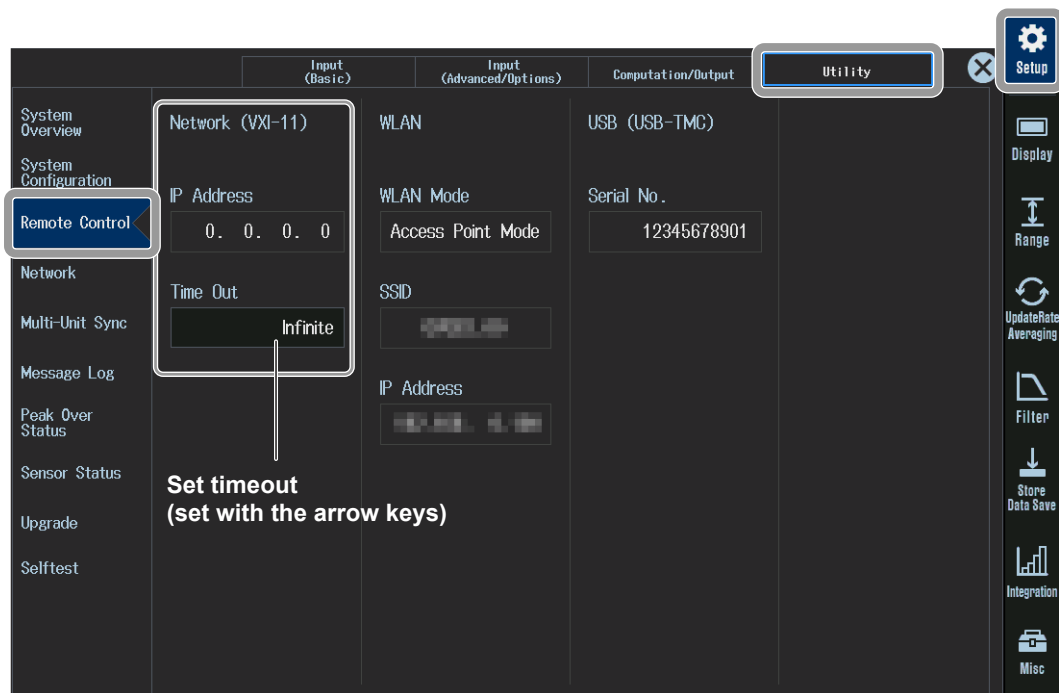
This section explains the following settings for remote control through the Ethernet interface.

- Setting the network connection timeout

When operating from an external monitor or using the Web server function

Utility settings overview (Utility)_Remote control settings (Remote Control)

Setup menu icon  > **Utility** > **Remote Control** > Network (VXI-11)



Note

- Use only one communication interface: Network, WLAN, or USB. If commands are sent simultaneously using another communication interface, the commands will not execute properly.
- For setup menu operations, see chapter 3 of the User's Manual (IM WT1501-01EN).
- The setup menu can also be displayed by pressing **MENU** on the front panel.
- The utility settings list screen can also be displayed by selecting **Utility** with the arrow keys and pressing **SET**.
- When the instrument is in remote mode, the REMOTE LED on the front panel illuminates green. In remote mode, all keys except KEY LOCK are disabled.

Setting the timeout period

If the instrument is not accessed for a specified period of time (timeout), the instrument closes the network connection.

Setting range: Infinite, 1 to 120 s

The default value is Infinite.

Setting TCP/IP

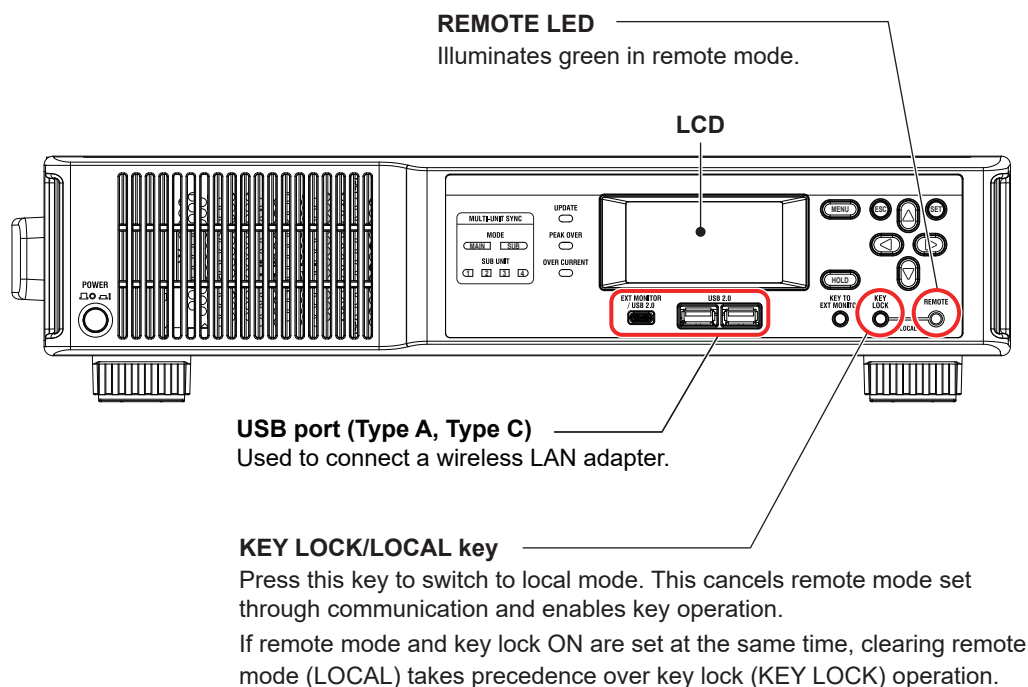
To use the Ethernet interface function, the following TCP/IP settings are required.

- IP address
- Subnet mask
- Default gateway

For details on operating from the instrument's LCD and configuring these settings, see section 14.2, "Setting TCP/IP" of the User's Manual (IM WT1501-01EN).

2.1 Part Names and Functions

Front panel



2.2 USB Interface (when using a wireless LAN adapter) Functions and Specifications

USB interface (when using a wireless LAN adapter) functions

Receive functions

You can make the same settings as those made through key operation when an external monitor or Web server is connected.

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

Send functions

The instrument can output measured and computed data.

It can output panel setup parameters and status bytes.

It can output error codes that have occurred.

USB Interface (when using a wireless LAN adapter) Specifications

Electrical and mechanical specifications	USB Rev. 2.0 compliant
Connector	Type-C, Type-A connector (receptacle)
Number of ports	Type-C:1, Type-A:2
Power Supply	Self-powered
System requirements:	A PC with a standard USB port, running Windows 11. A separate device driver is required to enable the connection with the PC.

Note

- Communication speed depends on the operating environment. Use Ethernet if necessary.
- You cannot use the USB interface simultaneously with other interfaces (Ethernet interface).

Switching between remote and local modes

Switching from local to remote mode

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

- The REMOTE LED lights green.
- All keys except the **KEY LOCK** 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 **KEY LOCK**, 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 LED turns off.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

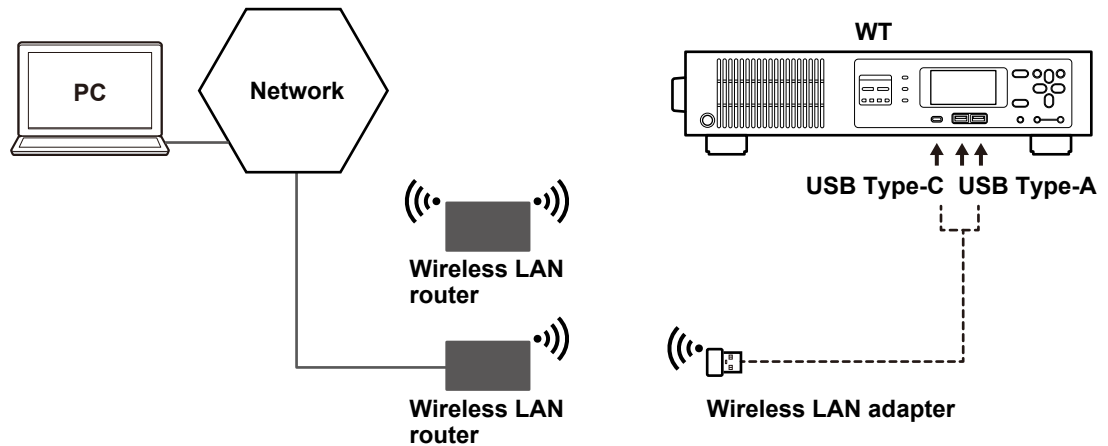
If remote mode and key lock ON are set at the same time, clearing remote mode (LOCAL) takes precedence over key lock (KEY LOCK) operation.

- On the first KEY LOCK operation, remote mode is cleared, and the REMOTE LED turns off.
- On the second KEY LOCK operation, key lock turns OFF, and the KEY LOCK LED turns off.

2.3 Connecting the Wireless LAN Adapter

Connection procedure

Connect the wireless LAN adapter to a Type-C or Type-A connector.



Notes on connection

- Insert the wireless LAN adapter firmly all the way into the USB port.
- After turning on the instrument, do not remove or insert the wireless LAN adapter until operation is possible (approximately 20 seconds).

2.4 Configuring the Instrument (when using a wireless LAN adapter)

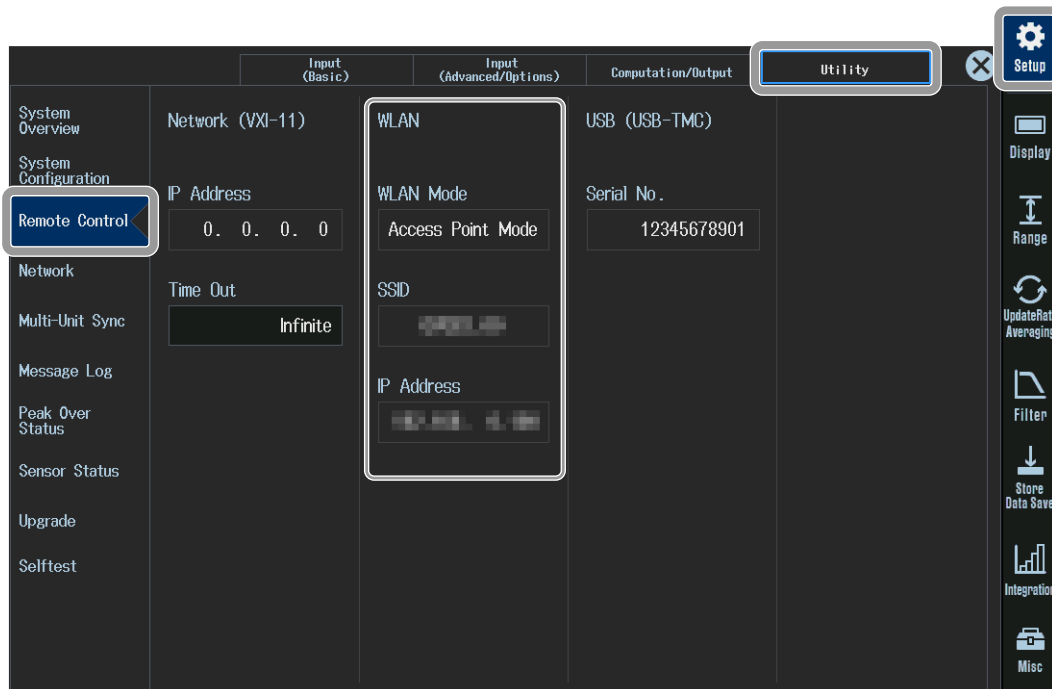
This section explains the settings for remote control through the USB interface when using a wireless LAN adapter.

- Checking the WLAN settings

When operating from an external monitor or using the Web server function

Utility settings overview (Utility)_Remote control settings (Remote Control)

You can check each setting by selecting **Setup** menu icon  > **Utility** > **Remote Control** > WLAN.



Note

- Use only one communication interface: Network, WLAN, or USB. If commands are sent simultaneously using another communication interface, the commands will not execute properly.
- For setup menu operations, see chapter 3 of the User's Manual (IM WT1501-01EN).
- The setup menu can also be displayed by pressing **MENU** on the front panel.
- The utility settings list screen can also be displayed by selecting **Utility** with the arrow keys and pressing **SET**.
- When the instrument is in remote mode, the REMOTE LED on the front panel illuminates green. In remote mode, all keys except KEY LOCK are disabled.

Setting WLAN

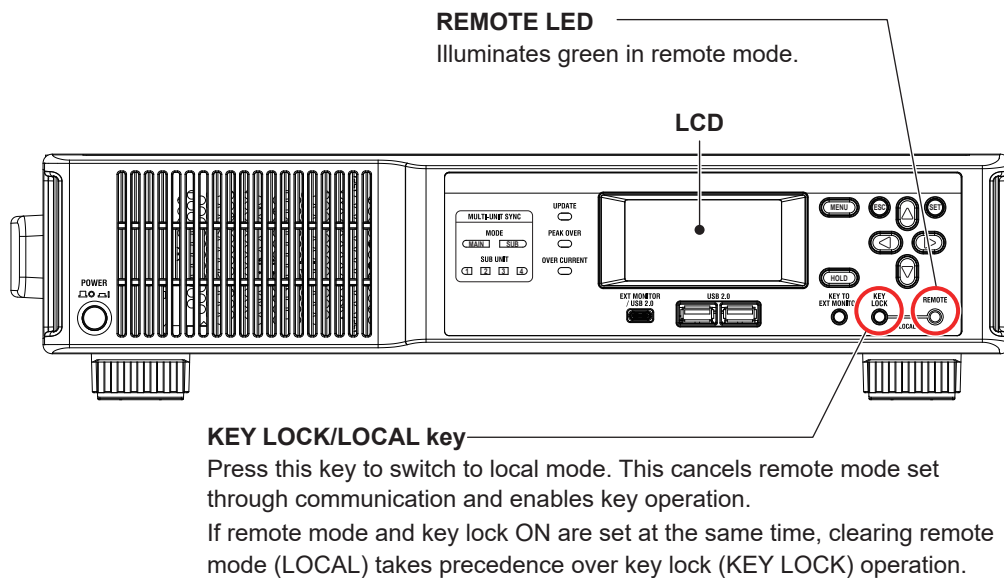
To use the USB interface function with a wireless LAN adapter, the following WLAN settings are required.

- WLAN mode
- SSID
- IP address

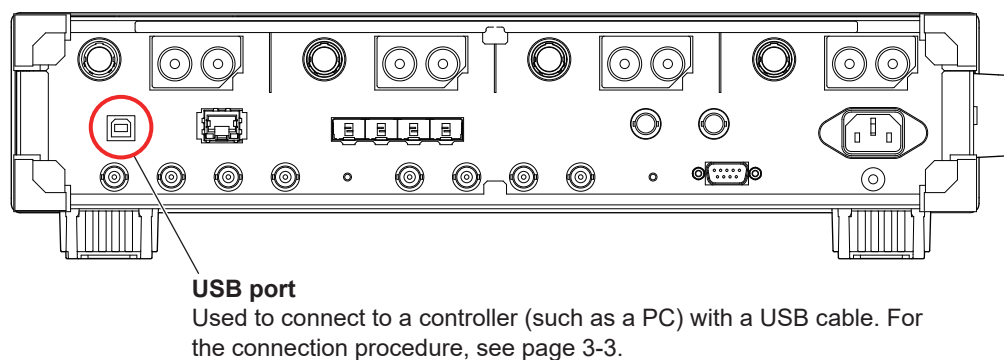
For operation from the instrument's LCD and details on how to configure these settings, see section 14.7, "Setting WLAN," of the User's Manual (IM WT1501-01EN).

3.1 Part Names and Functions

Front panel



Rear panel



3.2 USB Interface Functions and Specifications

USB interface functions

Receive functions

You can make the same settings as those made through key operation when an external monitor or Web server is connected.

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

Send functions

The instrument can output measured and computed data.

It can output panel setup parameters and status bytes.

It can output error codes that have occurred.

USB interface specifications

Electrical and mechanical specifications	USB Rev. 2.0 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 11. A separate device driver is required to enable the connection with the PC.

Note

You cannot use the USB interface simultaneously with other interfaces (Ethernet interface).

Switching between remote and local modes

Switching from local to remote mode

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

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

- The REMOTE LED lights green.
- All keys except the **KEY LOCK** 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 **KEY LOCK**, 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 LED turns off.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

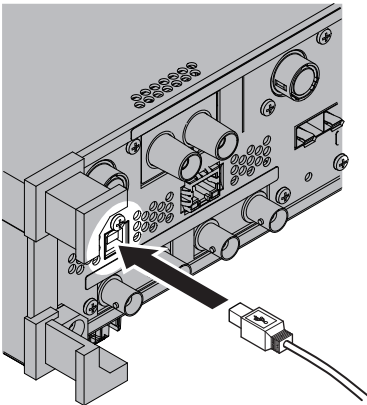
If remote mode and key lock ON are set at the same time, clearing remote mode (LOCAL) takes precedence over key lock (KEY LOCK) operation.

- On the first KEY LOCK operation, remote mode is cleared, and the REMOTE LED turns off.
- On the second KEY LOCK operation, key lock turns OFF, and the KEY LOCK LED turns off.

3.3 Connection through the USB Interface

Notes on connection

- Securely insert the USB cable all the way into the USB port.
- When connecting multiple instruments through USB hubs, connect this instrument to the USB hub closest to the controller.
- After turning on the instrument, do not connect or disconnect the USB cable until operation is possible (approx. 20 to 30 seconds). Doing so may damage the instrument.



3.4 Configuring the Instrument's USB Settings

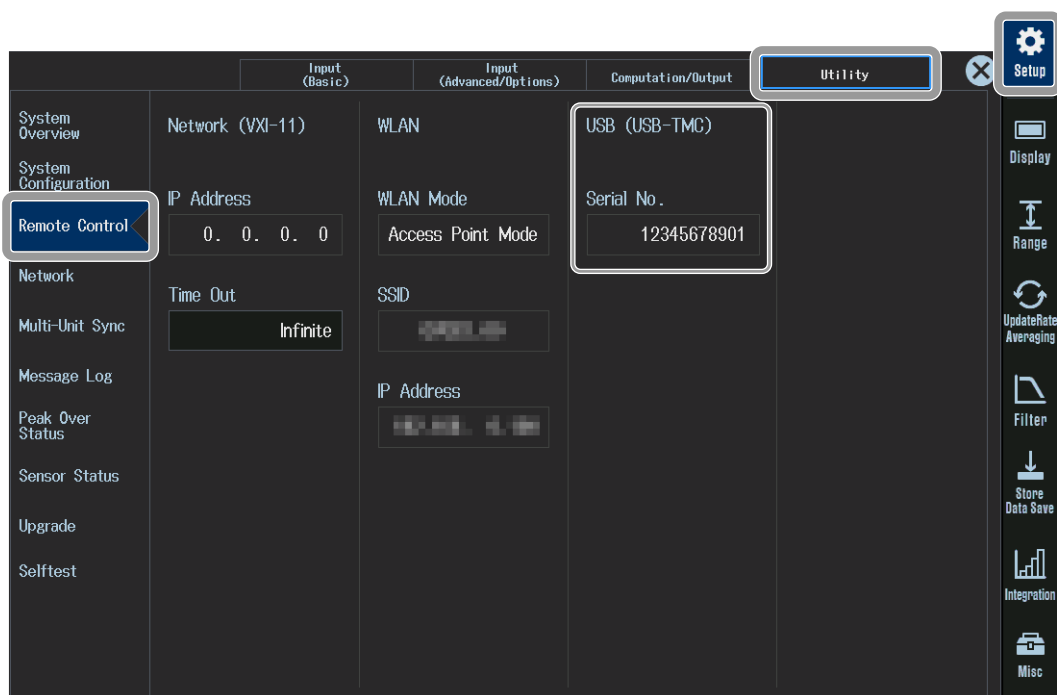
This section explains the following settings for remote control through the USB interface.

- Checking the instrument number used for USB TMC communication

When operating from an external monitor or using the Web server function

Utility settings overview (Utility)_Remote control settings (Remote Control)

You can check the instrument number by selecting **Setup** menu icon  > **Utility** > **Remote Control** > **USB (USB-TMC)**.



Note

- Use only one communication interface: Network, WLAN, or USB. If commands are sent simultaneously using another communication interface, the commands will not execute properly.
- Install the YOKOGAWA USB TMC (Test and Measurement Class) driver on the PC. For information about how to obtain the YOKOGAWA USB TMC driver, contact your nearest YOKOGAWA dealer, or access the USB driver download page from the following YOKOGAWA website and download the USB TMC driver.
<https://tmi.yokogawa.com/library/>
- Do not use any USB TMC driver or software provided by other companies.
- For setup menu operations, see chapter 3 of the User's Manual (IM WT1501-01EN).
- The setup menu can also be displayed by pressing **MENU** on the front panel.
- The utility settings list screen can also be displayed by selecting **Utility** with the arrow keys and pressing **SET**.
- When the instrument is in remote mode, the REMOTE LED on the front panel illuminates green. In remote mode, all keys except KEY LOCK are disabled.

For operation from the instrument's LCD and details on how to configure these settings, see section 18.5, "Remote Control," of the User's Manual (IM WT1501-01EN).

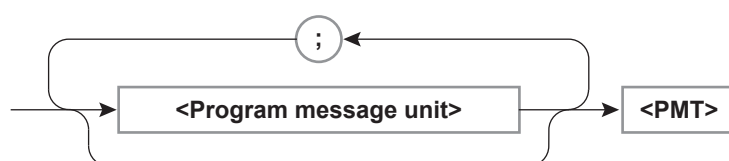
4.1 Messages

Messages

Information is transmitted and received between the controller and this instrument in units called messages. Messages sent from the controller to this instrument are called “program messages,” and messages that the controller receives from this instrument are called “response messages.” If a program message contains a command that requests a response (called a “query”), this instrument sends a response message after receiving the program message. The response to one program message is always one response message.

Program messages

The program message syntax is shown below.



<Program message unit>

A program message consists of one or more program message units connected together. Each program message unit corresponds to one command. This instrument executes commands in the order in which it receives them.

Program message units are separated with a semicolon (;).

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

Example : `INPut:CFACtor 3;INDependent OFF<PMT>`

Unit
Unit

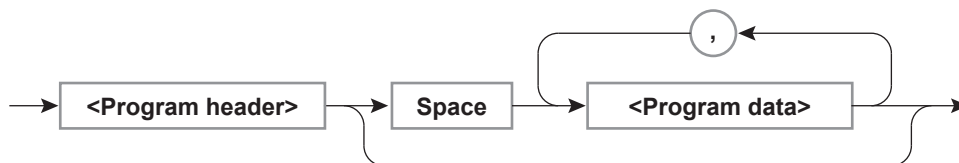
<PMT>

This is the program message terminator. There are the following three types.

NL (new line)	A single character with ASCII code 0AH, the same as LF (line feed)
^END	The END message defined in IEEE 488.1 The data byte transmitted at the same time as the END message becomes the last data in the program message.
NL^END	NL with an END message appended NL is not included in the program message.

Program message unit syntax

The program message unit syntax is shown below.

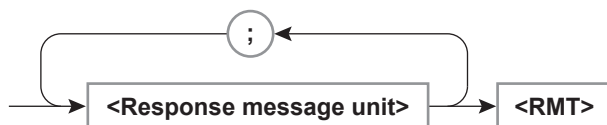


<Program header>	The program header indicates the type of command. For details, see page 4-4.
<Program data>	Add program data when conditions or other information required to execute the command are needed. When adding program data, separate the header and data with a space (ASCII code 20H). When there are multiple data items, separate them with commas (.). For details, see page 4-8.

Example : INPut:CFACtor 3<PMT>

Response messages

The program message syntax is shown below.



<Response message unit>

A response message consists of one or more response message units connected together. Each response message unit corresponds to one response. Response message units are separated with semicolons (;). For details on response message syntax, see the next page.

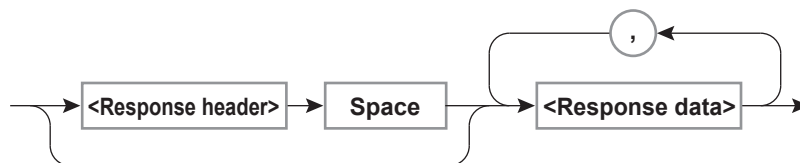
Example : INPUT:CFACOR 3 ; INDEPENDENT 0<RMT>
Unit Unit

<RMT>

This is the response message terminator, which is `NL^END`.

Response message unit syntax

The response message unit syntax is shown below.



<Response header>	A response header may be added before the response data. The header and data are separated by a single space. For details, see page 4-7.
<Response data>	Response data contains the content of the response. When there are multiple data items, they are separated with commas (.). For details, see page 4-7.

Example `100.00E-03<RMT> :DISPLAY:MODE WAVE<RMT>`

Data
Header
Data

If a program message contains multiple queries, the order of the responses follows the order of the queries. Many queries return one response message unit, but some return multiple units. The response to the first query is the first unit, but the *n*th response is not necessarily the *n*th unit. To retrieve responses reliably, send separate program messages.

Notes on transmitting and receiving messages

- When you send a program message that does not contain a query, you can send the next program message at any time.
- When you send a program message that contains a query, you must receive the response message before sending the next program message. If you send the next program message without receiving the response message, or after receiving only part of it, an error occurs. Response messages that were not received are discarded.
- If the controller attempts to receive a response message when there is none, an error occurs. If the controller attempts to receive a response message before it finishes sending the program message, an error occurs.
- When you send a program message containing multiple units, if it contains an incomplete program unit, this instrument picks out the program message units that appear to be complete and attempts to execute them, but execution may not always succeed. Even if such a message contains a query, a response may not always be returned.

Deadlock state

This instrument can store messages of at least 1024 bytes in both the transmit and receive buffers. The number of bytes may increase or decrease depending on the operating state. If both of these buffers become full at the same time during transmission and reception, this instrument becomes inoperable. This is called a deadlock state. When this occurs, you can recover from the inoperable state by discarding the response message.

Deadlock does not occur if the program message, including <PMT>, is 1024 bytes or less.

Deadlock also does not occur with program messages that do not contain queries.

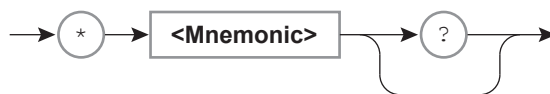
4.2 Commands

Commands

There are three types of commands (program headers) that are sent from the controller to this instrument, as shown below. Each type uses a different program header syntax.

Common command headers

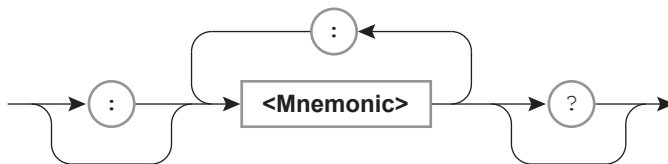
Commands specified in IEEE 488.2-1992 are called common commands. The common command header syntax is shown below. An asterisk (*) must always be added at the beginning.



Example of a common command: ***CLS**

Compound headers

Instrument-specific commands other than common commands are classified by function and organized hierarchically. The compound header syntax is shown below. When writing a lower hierarchy level, always add a colon (:).



Example of a compound header: **:DISPlay:MODE**

Simple headers

These are functionally independent commands that do not have a hierarchy. The header syntax is shown below.



Example of a simple header: **:HOLD**

Note

A <mnemonic> is a character string consisting of letters and numbers.

Writing commands consecutively

Groups

A group is a set of commands that share a common compound header with hierarchical headers. Smaller groups may also exist within a group.

Example Integration-related group (partial list)

- | | |
|---------------------------------------|-------------------------------------|
| • <code>:INTEGrate?</code> | • <code>:INTEGrate:RTIME:END</code> |
| • <code>:INTEGrate:MODE</code> | • <code>:INTEGrate:ACAL</code> |
| • <code>:INTEGrate:TIMer</code> | • <code>:INTEGrate:START</code> |
| • <code>:INTEGrate:RTIME?</code> | • <code>:INTEGrate:STOP</code> |
| • <code>:INTEGrate:RTIME:START</code> | • <code>:INTEGrate:RESet</code> |

When writing commands in the same group consecutively

This instrument stores the hierarchy level of the command being executed and parses the next command that is sent by assuming that it belongs to the same hierarchy level. Therefore, for commands in the same group, you can omit the common header portion.

Example `:INTEGrate:MODE NORMAl;ACAL ON<PMT>`

When writing commands in different groups consecutively

When writing a command from a different group after another command, add a colon (:) to the beginning of the header. It cannot be omitted.

Example `:INTEGrate:MODE NORMAl;:DISPlay:MODE NUMeric<PMT>`

When writing a simple header consecutively

When writing a simple header after another command, add a colon (:) to the beginning of the simple header. It cannot be omitted.

Example `:INTEGrate:MODE NORMAl;:HOLD ON<PMT>`

When writing a common command consecutively

Common commands defined in IEEE 488.2-1992 are unrelated to the hierarchy. You do not need to add a colon (:).

Example `:INTEGrate:MODE NORMAl;*CLS;ACAL ON<PMT>`

When commands are separated with <PMT>

Separating commands with a terminator means sending two program messages. Therefore, even when continuing with commands in the same group, you cannot omit the common header.

Example `:INTEGrate:MODE NORMAl<PMT>:INTEGrate:ACAL ON<PMT>`

Higher-level queries

A query formed by adding a question mark (?) to a higher-level command in a group is called a higher-level query. When you execute the highest-level query, you can receive all the settings that can be configured in that group at once. Some groups with three or more hierarchy levels output all lower hierarchy levels.

Example :INTEGrate?<PMT> -> :INTEGRATE:MODE NORMAL;TIMER 0,0,0;ACAL 0<RMT>

The response to a higher-level query can be sent as is to this instrument as a program message. When sent, it reproduces the settings that were in effect when the higher-level query was executed. However, some higher-level queries do not return setting information that is not currently in use. Note that the response does not necessarily output all information in the group.

Header interpretation rules

This instrument interprets received headers according to the following rules.

- The instrument does not distinguish between uppercase and lowercase letters in mnemonics.

Example CURSor can also be written as cursor or Cursor.

- The lowercase portion can be omitted.

Example CURSor can also be written as CURSO or CURS.

- A question mark (?) at the end of a header indicates that the header is a query. The question mark (?) cannot be omitted.

Example The shortest abbreviation of CURSor? is CURS?.

- If <x> (a number) at the end of a mnemonic is omitted, it is interpreted as x = 1.

Example For ELEMeNt<x>, ELEM means ELEMeNt1.

- The portions enclosed in brackets ([]) can be omitted.

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

However, for a higher-level query, the final portion cannot be omitted.

Example SCALing? and SCALing:STATe? are different queries.

4.3 Responses

Responses

When the controller sends a query, which is a command with a question mark (?), this instrument returns a response message to that query. There are two response formats.

Responses with header and data

Responses that can be used as program messages as is are returned with the command header added.

Example `:DISPlay:MODE?<PMT> -> :DISPLAY:MODE WAVE<RMT>`

Data-only responses

Responses that cannot be used as program messages as is (query-only commands) are returned as data only, without a header. However, some query-only commands return responses with a header.

Example `[ :INPut ] :POVer?<PMT> -> 0<RMT>`

Returning responses without headers

You can force responses that are normally returned with “header + data” to be returned without a header. Use the `:COMMUnicate:HEADer` command for this.

Abbreviations

Normally, response headers are returned with the lowercase portion omitted. You can also return them in full spelling, without omitting these portions. To return headers in full spelling, use the `:COMMUnicate:VERBoSe` command.

When abbreviated headers are returned, the portions enclosed in brackets ([]) are also omitted.

4.4 Data

Data refers to conditions or numeric values written after a header, separated by a space. Data is classified as follows.

Category	Meaning	See page
<Decimal>	A numeric value expressed as a decimal number Example: Setting the VT ratio ->[:INPut]:SCALing:VT:ELEMeNt1 100	4-8
<Voltage>, <Current>, <Time>, <Frequency>	A numeric value with a physical dimension Example: Setting the voltage range ->[:INPut]:VOLTage:RANGE:ELEMeNt1 100V	4-9
<Register>	A register value expressed in binary, octal, decimal, or hexadecimal Example: Extended event register value ->:STATUS:EESE #HFE	4-9
<Character data>	A specified character string (mnemonic). Select from the strings enclosed in braces ({ }). Example: Selecting the trigger mode ->:DISPlay:WAVE:TRIGger:MODE {AUTO NORMAl OFF}	4-10
<Boolean>	Represents ON or OFF. Set using ON, OFF, or a numeric value Example: Turning data hold on ->:HOLD ON	4-10
<String data>	An arbitrary character string Example: User-defined function ->:MEASure:FUNCTioN1:EXPRession "URMS(E1)"	4-10
<Filename>	Represents a file name Example: Save file name ->:FILE:SAVE:SETup[:EXECute] "CASE1"	–
<Block data>	Data with arbitrary 8-bit values Example: Response for captured waveform data -> #40012ABCDEF GHIJKL	4-10

<Decimal>

A <Decimal> is a numeric value expressed as a decimal number, as shown in the table below. It is written in the NR format 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 of <NR1> to <NR3>	

- When this instrument receives a decimal number sent from the controller, it accepts any of the <NR1> to <NR3> formats. This is represented by <NRf>.
- For response messages returned from this instrument to the controller, whether <NR1>, <NR2>, or <NR3> is used is determined for each query. The format does not change according to the size of the value.
- In the <NR3> format, the plus sign (+) after E can be omitted. The minus sign (–) cannot be omitted.
- If you specify a value outside the setting range, the value is set to the closest value within the available setting range.
- If you specify a value with more precision than supported, the value is rounded.

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

<Voltage>, <Current>, <Time>, and <Frequency> are <Decimal> data with physical dimensions. A <Multiplier> and a <Unit> can be appended to the <NRf> format described above.

Use one of the following syntaxes.

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

The available <Multipliers> and <Units> are shown in the table below.

<Multiplier>			<Unit>		
Symbol	Name	Multiplier	Symbol	Name	Meaning
EX	Exa	10 ¹⁸	V	Volt	Voltage
PE	Peta	10 ¹⁵	A	Ampere	Current
T	Tera	10 ¹²	S	Second	Time
G	Giga	10 ⁹	HZ	Hertz	Frequency
MA	Mega	10 ⁶	MHZ	Megahertz	Frequency
K	Kilo	10 ³			
M	Milli	10 ⁻³			
U	Micro	10 ⁻⁶			
N	Nano	10 ⁻⁹			
P	Pico	10 ⁻¹²			
F	Femto	10 ⁻¹⁵			

- <Multiplier> and <Unit> are not case sensitive.
- Use U to represent micro, not μ .
- Use MA to represent mega so that it can be distinguished from milli, which is represented by M. However, for current, MA is interpreted as milliamperes. Megahertz is represented by MHZ. Therefore, for frequency, M (milli) cannot be used as a multiplier.
- If both <Multiplier> and <Unit> are omitted, the basic units (V, A, S, and HZ) are used.
- Response messages are always in <NR3> format. They are returned in the basic units without a <Multiplier> or <Unit>.

<Register>

<Register> is integer data that can be expressed not only as a <Decimal> but also as a <Hexadecimal>, <Octal>, or <Binary> value. Use this data type when each bit of the numeric value has a meaning.

Use one of the following syntaxes.

Syntax	Example
<NRf>	1
#H<hexadecimal number consisting of 0 to 9 and A to F>	#H0F
#Q<octal number consisting of 0 to 7>	#Q777
#B<binary number consisting of 0 or 1>	#B001100

- <Register> is not case sensitive.
- Response messages are always returned in <NR1> format.

<Character data>

<Character data> is data consisting of specified characters (mnemonics). It is mainly used to express choices. Select and write one of the strings enclosed in braces ({}). The data is interpreted in the same way as described in "Header interpretation rules" on page 4-6.

Syntax	Example
{ AUTO NORMa l}	AUTO

- For response messages, as with headers, you can use `:COMMunicate:VERBo` to select whether the response is returned in full spelling or abbreviated form.
- The `:COMMunicate:HEADer` setting does not affect <Character data>.

<Boolean>

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

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

- When <NRf> is used, the value is rounded to an integer. If the rounded value is 0, it is OFF. If the rounded value is not 0, it is ON.
- Response messages are always returned as 1 for ON and 0 for OFF.

<String data>

<String data> is not a specified character string like <Character data>, but an arbitrary character string. Use the following syntax, enclosing the string in single quotation marks (') or double quotation marks (").

Syntax	Example
< String data >	' ABC ' " IEEE488.2-1992 "

- When a double quotation mark (") occurs as a character in a string enclosed in double quotation marks (""), represent it with two double quotation marks ("""). The same applies to single quotation marks (').
- Response messages are always returned enclosed in double quotation marks (").
- Because <String data> can be an arbitrary character string, if the final single quotation mark (') or double quotation mark (") is missing, this instrument interprets the remaining program message units as part of <String data>, and errors may not be detected correctly.

<Block data>

<Block data> is data with arbitrary 8-bit values. On this instrument, it is used only in response messages. The syntax is shown below.

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

#N: Indicates <Block data>. N indicates the number of ASCII characters (digits) that represent the number of data bytes that follows.

<N-digit decimal number>: Represents the number of data bytes. Example: 00000010 = 10 bytes.

<Data byte sequence>: Represents the actual data. Example: ABCDEFGHIJ.

The data can be any 8-bit value (0 to 255). Therefore, ASCII code 0AH, which represents NL, may also appear as data, so care is required on the controller side.

4.5 Synchronizing with the Controller

Overlapped commands and sequential commands

There are two types of commands: overlapped commands and sequential commands. With an overlapped command, operation by the next command may start before operation by the previously sent command is complete.

For example, when specifying a voltage range and querying the result, if you send the following program message, the response always returns the latest setting value (in this case, 100 V).

```
:INPut:VOLTage:RANGe:ELEMenT1 100V;ELEMenT1?<PMT>
```

This is because `:INPut:VOLTage:RANGe:ELEMenT1` makes the next command wait until its own processing is complete. This type of command is called a sequential command.

In contrast, for example, when you execute file loading and want to query the resulting voltage range, if you send the following program message, `:INPut:VOLTage:RANGe:ELEMenT1?` is executed before file loading is complete, so the returned voltage range is the value from before file loading.

```
:FILE:LOAD:SETup "FILE1";:INPut:VOLTage:RANGe:ELEMenT1?
```

Executing the next command before a command's own processing is complete, as with `:FILE:LOAD:SETup`, is called overlapped operation. A command that performs overlapped operation is called an overlapped command.

In such cases, you can avoid overlapped operation by using one of the methods shown below.

How to synchronize with overlapped commands

Using the *WAI command

The `*WAI` command makes commands following `*WAI` wait until the overlapped command finishes.

```
Example :COMMunicate:OPSE #H0040;:FILE:LOAD:SETup "FILE1"  
;*WAI;:INPut:VOLTage:RANGe:ELEMenT1?<PMT>
```

`:COMMunicate:OPSE` is a command that selects the target of `*WAI`. Here, only media access is specified as the target.

Because `*WAI` is executed immediately before `:INPut:VOLTage:RANGe:ELEMenT1?`, `:INPut:VOLTage:RANGe:ELEMenT1?` is not executed until file loading is complete.

Using the :COMMunicate:OVERlap command

The :COMMunicate:OVERlap command allows or prohibits overlapped operation.

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

:COMMunicate:OVERlap #HFFBF allows overlapped operation other than media access. Because file loading is not allowed to perform overlapped operation, the next command, :FILE:LOAD:SETup, operates in the same way as a sequential command. Therefore, :INPut:VOLTagE:RANGe:ELEMent1? is not executed until file loading is complete.

Using the *OPC command

The *OPC command sets the OPC bit, bit 0 of the standard event register (see page 6-5), to 1 when overlapped operation is complete.

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 to occur.)

:INPut:VOLTagE:RANGe:ELEMent1?<PMT>

:COMMunicate:OPSE is a command that selects the target of *OPC. Here, only media access is specified as the target.

*ESE 1 and *SRE 32 indicate that a service request is generated only 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 occurs.

Using the *OPC? query

The *OPC? query generates a response when overlapped operation is complete.

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

:COMMunicate:OPSE is a command that selects the target of *OPC?. Here, only media access is specified as the target.

Because *OPC? does not create a response until overlapped operation is complete, file loading is complete when the response to *OPC? has been read.

Note

Most commands are sequential commands. Overlapped commands are explicitly identified as overlapped commands in chapter 5. All other commands are sequential commands.

Synchronizing for commands other than overlapped commands

Even with sequential commands, synchronization may be necessary to query measurement data correctly. For example, suppose you want to query the latest numeric data each time measurement data is updated. If you send `:NUMeric[:NORMal]:VALue?` at arbitrary timing, the instrument returns the current measurement data regardless of whether measurement data update has completed. As a result, you may receive the same data as the previous data. In this case, you need to use one of the following methods to synchronize with the timing at which measurement data update is complete.

Using the `:STATus:CONDition?` query

`:STATus:CONDition?` is a command that queries the contents of the condition register (see page 6-7). You can determine whether measurement data is being updated by reading bit 0 of the condition register. If bit 0 of the condition register is 1, measurement data is being updated. If it is 0, measurement data can be queried.

Using the extended event register

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

Example `:STATus:FILTer1 FALL;:STATus:EES 1;EESR?;*SRE 8<PMT>`

(Read the response to `:STATus:EESR?`)

Loop

(Wait for a service request to occur.)

`:NUMeric[:NORMal]:VALue?<PMT>`

(Read the response to `:NUMeric[:NORMal]:VALue?`)

`:STATus:EESR?<PMT>`

(Read the response to `:STATus:EESR?`)

(Return to Loop)

`:STATus:FILTer1 FALL` indicates that the transition filter is set so that bit 0 (FILTer1) of the extended event register is set to 1 when bit 0 of the condition register changes from 1 to 0.

`:STATus:EES 1` is a command that reflects only bit 0 of the extended event register in the status byte.

`:STATus:EESR?` is executed to clear the extended event register.

`*SRE 8` is a command that causes a service request to occur only due to the extended event register.

`:NUMeric[:NORMal]:VALue?` is not executed until a service request occurs.

Using the COMMunicate:WAIT command

:COMMunicate:WAIT is a command that waits 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 method described above.

:COMMunicate:WAIT 1 indicates waiting until bit 0 of the extended event register is set to 1.

:NUMeric[:NORMal]:VALue? is not executed until bit 0 of the extended event register becomes 1.

5.1 Command List

Command	Explanation	Page
AUX Group		
:AUX<x>?	Queries all auxiliary input settings.	5-23
:AUX<x>:AUTO	Sets or queries auto voltage range on/off for auxiliary signals (analog input type).	5-23
:AUX<x>:FILTer?	Queries all auxiliary input filter settings.	5-23
:AUX<x>:FILTer[:LINE]	Sets or queries the auxiliary input line filter.	5-23
:AUX<x>:FILTer:NOISe	Sets or queries the auxiliary input pulse noise filter.	5-23
:AUX<x>:LSCale?	Queries all auxiliary input linear scaling settings.	5-23
:AUX<x>:LSCale:AVALue	Sets or queries slope A of the auxiliary input linear scale.	5-23
:AUX<x>:LSCale:BVALue	Sets or queries offset value B of the auxiliary input linear scale.	5-24
:AUX<x>:LSCale:CALCulate?	Queries all settings for calculating the parameters of the auxiliary input linear scale.	5-24
:AUX<x>:LSCale:CALCulate:{P1X P1Y P2X P2Y}	Sets or queries the data {Point1X Point1Y Point2X Point2Y} used to calculate the parameters of the auxiliary input linear scale.	5-24
:AUX<x>:LSCale:CALCulate:EXECute	Executes the calculation of the parameters of the auxiliary input linear scale.	5-24
:AUX<x>:NAME	Sets or queries the auxiliary input name.	5-24
:AUX<x>:PRANge	Sets or queries the range for auxiliary signals (pulse input type).	5-24
:AUX<x>:RANGe	Sets or queries the voltage range for auxiliary signals (analog input type).	5-24
:AUX<x>:SCALing	Sets or queries the scaling factor for auxiliary input.	5-25
:AUX<x>:TYPE	Sets or queries the auxiliary input type.	5-25
:AUX<x>:UNIT	Sets or queries the unit added to the auxiliary input.	5-25
CAN Group		
:CAN?	Queries all settings for the CAN data output function.	5-26
:CAN:CONDition?	Queries the CAN bus signal port status.	5-26
:CAN:FD:SETup:BRATe	Sets or queries the bit rate of the arbitration phase for CAN bus signals (CAN FD).	5-26
:CAN:FD:SETup:DBRate	Sets or queries the bit rate of the data phase for CAN bus signals (CAN FD).	5-26
:CAN:FD:SETup:DSPoInt	Sets or queries the sample point of the data phase for CAN bus signals (CAN FD).	5-26
:CAN:FD:SETup:SPOInt	Sets or queries the sample point of the arbitration phase for CAN bus signals (CAN FD).	5-26
:CAN:FD:SETup:TERMinator	Sets or queries the port terminator for CAN bus signals (CAN FD).	5-26
:CAN:FILE?	Queries all settings for saving DBC files.	5-26
:CAN:FILE:ANAMing	Sets or queries the function for automatically generating DBC file names when saving.	5-26
:CAN:FILE:CDIRectory	Changes the save destination directory for DBC files.	5-26
:CAN:FILE:DRIVE	Sets the save destination drive for DBC files.	5-27

5.1 Command List

Command	Explanation	Page
:CAN:FILE:FREE?	Queries the free space (byte) on the save destination drive for stored numeric data.	5-27
:CAN:FILE:NAME	Sets or queries the save file name for DBC files.	5-27
:CAN:FILE:PATH?	Queries the DBC file save destination as an absolute path.	5-27
:CAN:FILE:SAVE:DBC:ABORT	Aborts saving the DBC file.	5-27
:CAN:FILE:SAVE:DBC:EXECUTE	Saves the DBC file.	5-27
:CAN:MFormat	Sets or queries the message format for CAN bus signals.	5-27
:CAN:NUMeric:ESElect?	Queries all settings for transmit items of CAN bus signals for numeric data (element selection method).	5-27
:CAN:NUMeric:ESElect:ALL	Sets the output elements for all functions collectively when numeric data is transmitted as CAN bus signals.	5-27
:CAN:NUMeric:ESElect:<Function>	Sets or queries the output elements for each function when numeric data is transmitted as CAN bus signals.	5-27
:CAN:NUMeric:ESElect:PRESet<x>	Presets the output elements for functions to a predefined pattern when numeric data is transmitted as CAN bus signals.	5-27
:CAN:NUMeric[:NORMal]:UNIT	Sets or queries the target unit when numeric data is transmitted as CAN bus signals.	5-28
:CAN:NUMeric:MID	Sets or queries the message ID for CAN bus signals.	5-28
:CAN:OUT:COUNT	Sets or queries the number of times CAN bus signal data frames are transmitted.	5-28
:CAN:PROTocol	Sets or queries the protocol of the CAN bus signal port.	5-28
:CAN:SETup:BRATe	Sets or queries the bit rate for CAN bus signals.	5-28
:CAN:SETup:SPOint	Sets or queries the sample point for CAN bus signals.	5-28
:CAN:SETup:TERMinator	Sets or queries the port terminator for CAN bus signals.	5-28
:CAN:START	Starts CAN bus signal output for numeric data.	5-28
:CAN:STOP	Stops CAN bus signal output for numeric data.	5-28

COMMunicate Group

:COMMunicate?	Queries all communication settings.	5-29
:COMMunicate:HEADer	Sets or queries whether responses to queries are returned with a header (example: :DISPLAY:CONFIG NUMERIC) or without a header (example: NUMERIC).	5-29
:COMMunicate:LOCKout	Sets or clears local lockout.	5-29
:COMMunicate:OPSE	Sets or queries the overlap commands that are subject to *OPC, *OPC?, and *WAI.	5-29
:COMMunicate:OPSR?	Queries the value of the operation pending status register.	5-29
:COMMunicate:OVERlap	Sets or queries which commands operate as overlap commands.	5-29
:COMMunicate:REMOte	Sets remote/local. The instrument is in remote mode when this is ON.	5-29
:COMMunicate:VERBOse	Sets or queries whether responses to queries are returned in full form (example: :INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03) or abbreviated form (example: :VOLT:RANG:ELEM1 1.000E+03).	5-30
:COMMunicate:WAIT	Waits until one of the specified extended events occurs.	5-30
:COMMunicate:WAIT?	Creates a response when one of the specified extended events occurs.	5-30

CURSor Group

:CURSor?	Queries all cursor measurement settings.	5-31
----------	--	------

Command	Explanation	Page
:CURSor:BAR<x>?	Queries all settings for cursor measurement in the bar graph display.	5-31
:CURSor:BAR<x>:LINKage	Sets or queries the on/off status of cursor position linkage in the bar graph display.	5-31
:CURSor:BAR<x>:POSition<x>	Sets or queries the cursor position in the bar graph display.	5-31
:CURSor:BAR<x>[:STATe]	Sets or queries the on/off status of cursor display in the bar graph display.	5-31
:CURSor:BAR<x>:{Y<x> DY}?	Queries the cursor measurement values in the bar graph display.	5-31
:CURSor:WAVE<x>?	Queries all settings for cursor measurement in the waveform display.	5-31
:CURSor:WAVE<x>:LINKage	Sets or queries the on/off status of cursor position linkage in the waveform display.	5-31
:CURSor:WAVE<x>:PATH	Sets or queries the cursor path in the waveform display.	5-31
:CURSor:WAVE<x>:POSition<x>	Sets or queries the cursor position in the waveform display.	5-32
:CURSor:WAVE<x>[:STATe]	Sets or queries the on/off status of cursor display in the waveform display.	5-32
:CURSor:WAVE<x>:TRACe<x>	Sets or queries the trace for the cursor in the waveform display.	5-32
:CURSor:WAVE<x>:{X<x> DX PERDt Y<x> DY}?	Queries cursor measurement values in the waveform display.	5-32

DISPlay Group

:DISPlay?	Queries all display settings.	5-33
:DISPlay:BAR<x>?	Queries all bar graph display settings.	5-33
:DISPlay:BAR<x>:FORMat	Sets or queries the bar graph display format (number of screen divisions).	5-33
:DISPlay:BAR<x>:ITEM<x>?	Queries all settings for each bar graph.	5-33
:DISPlay:BAR<x>:ITEM<x>[:FUNCTION]	Sets or queries the bar graph display item (function and element).	5-33
:DISPlay:BAR<x>:ITEM<x>:SCALing?	Queries all bar graph scaling settings.	5-33
:DISPlay:BAR<x>:ITEM<x>:SCALing:MODE	Sets or queries the bar graph scaling mode.	5-33
:DISPlay:BAR<x>:ITEM<x>:SCALing:VALue	Sets or queries the bar graph manual scaling upper limit.	5-34
:DISPlay:BAR<x>:ITEM<x>:SCALing:VERTical	Sets or queries the bar graph vertical scale type.	5-34
:DISPlay:BAR<x>:ITEM<x>:SCALing:XAXis	Sets or queries the bar graph X-axis position.	5-34
:DISPlay:BAR<x>:ORDer	Sets or queries the starting and ending harmonic orders displayed in the bar graph.	5-34
:DISPlay:CONFIg	Sets or queries the display mode.	5-35
:DISPlay:NUMeric<x>?	Queries all numeric display settings.	5-35
:DISPlay:NUMeric<x>:NORMal?	Queries all numeric display settings.	5-35
:DISPlay:NUMeric<x>[:NORMal]:ALL?	Queries all numeric display (All items display) settings.	5-35
:DISPlay:NUMeric<x>[:NORMal]:ALL:ORDer	Sets or queries the harmonic order displayed on the harmonic measurement function display page of numeric display (All items display).	5-35
:DISPlay:NUMeric<x>[:NORMal]:ALL:PAGE	Sets or queries the display page for numeric display (All items display).	5-35

5.1 Command List

Command	Explanation	Page
:DISPlay:NUMeric<x>[:NORMal]:FORMat	Sets or queries the numeric display format.	5-36
:DISPlay:NUMeric<x>[:NORMal]:LIST?	Queries all numeric display (harmonic list display) settings.	5-36
:DISPlay:NUMeric<x>[:NORMal]:LIST:CURSor	Sets or queries the cursor position in numeric display (harmonic list display).	5-36
:DISPlay:NUMeric<x>[:NORMal]:LIST:HEADer	Sets or queries the cursor position in the header area of numeric display (harmonic list display).	5-36
:DISPlay:NUMeric<x>[:NORMal]:LIST:ITEM<x>	Sets or queries the display item (function and element) for numeric display (harmonic list display).	5-37
:DISPlay:NUMeric<x>[:NORMal]:LIST:ORDer	Sets or queries the harmonic order cursor position in the data area of numeric display (harmonic list display).	5-37
:DISPlay:NUMeric<x>[:NORMal]:MATRix?	Queries all numeric display (matrix display) settings.	5-37
:DISPlay:NUMeric<x>[:NORMal]:MATRix:COLumn?	Queries all settings for display columns in numeric display (matrix display).	5-37
:DISPlay:NUMeric<x>[:NORMal]:MATRix:COLumn:ITEM<x>	Sets or queries the column display item for numeric display (matrix display).	5-37
:DISPlay:NUMeric<x>[:NORMal]:MATRix:COLumn:NUMber	Sets or queries the number of columns in numeric display (matrix display).	5-38
:DISPlay:NUMeric<x>[:NORMal]:MATRix:COLumn:RESet	Resets the column display items for numeric display (matrix display) to their default values.	5-38
:DISPlay:NUMeric<x>[:NORMal]:MATRix:CURSor	Sets or queries the cursor position in numeric display (matrix display).	5-38
:DISPlay:NUMeric<x>[:NORMal]:MATRix:ITEM<x>	Sets or queries the display item (function and harmonic order) for numeric display (matrix display).	5-38
:DISPlay:NUMeric<x>[:NORMal]:MATRix:PAGE	Sets or queries the display page for numeric display (matrix display).	5-39
:DISPlay:NUMeric<x>[:NORMal]:MATRix:PRESet	Presets the display items for numeric display (matrix display) to a specified pattern.	5-39
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}?	Queries all numeric display ({4 items 8 items 16 items} display) settings.	5-39
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}:CURSor	Sets or queries the cursor position in numeric display ({4 items 8 items 16 items} display).	5-39
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}:ITEM<x>[:FUNcti on]	Sets or queries the display item (function, element, and harmonic order) for numeric display ({4 items 8 items 16 items} display).	5-40
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}:ITEM<x>:UNIT	Sets or queries the unit of the display item (function, element, and harmonic order) for numeric display ({4 items 8 items 16 items} display).	5-40
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}:PAGE	Sets or queries the display page for numeric display ({4 items 8 items 16 items} display).	5-40
:DISPlay:NUMeric<x>[:NORMal]:{VAL4 VAL8 VAL16}:PRESet	Presets the display items for numeric display ({4 items 8 items 16 items} display) to a specified pattern.	5-41
:DISPlay:VECTor<x>?	Queries all vector display settings.	5-41
:DISPlay:VECTor<x>:FORMat	Sets or queries the vector display format (number of screen divisions).	5-41
:DISPlay:VECTor<x>:ITEM<x>?	Queries all settings for each vector.	5-41
:DISPlay:VECTor<x>:ITEM<x>:OBJec t	Sets or queries the wiring unit to be displayed as the vector object.	5-41
:DISPlay:VECTor<x>:ITEM<x>:{UMAG IMAG}	Sets or queries the vector display {voltage current} zoom factor.	5-41

Command	Explanation	Page
:DISPlay:VECTor<x>:NUMeric	Sets or queries the on/off status of numeric data display for vector display.	5-42
:DISPlay:VUNit<x>	Sets or queries the view unit (View Unit).	5-42
:DISPlay:WAVE<x>?	Queries all waveform display settings.	5-42
:DISPlay:WAVE<x>:ALL	Sets on/off for all waveform displays at once.	5-42
:DISPlay:WAVE<x>:FORMat	Sets or queries the waveform display format (number of screen divisions).	5-42
:DISPlay:WAVE<x>:GRATicule	Sets or queries the graticule type.	5-42
:DISPlay:WAVE<x>:INTerpolate	Sets or queries the waveform interpolation method.	5-42
:DISPlay:WAVE<x>:MAPPing?	Queries all settings for assigning waveforms to split formats (split screens).	5-42
:DISPlay:WAVE<x>:MAPPing[:MODE]	Sets or queries the method for assigning waveforms to split formats (split screens).	5-42
:DISPlay:WAVE<x>:MAPPing:USER:{U<x> I<x> SPEed<x> TORQue<x> AUX<x>}	Sets or queries the assignment of {voltage current rotation speed torque auxiliary signal} waveforms to split formats (split screens).	5-43
:DISPlay:WAVE<x>:POSition?	Queries all waveform vertical position (center-position level) settings.	5-43
:DISPlay:WAVE<x>:POSition:{U<x> I<x>}	Sets or queries the vertical position (center-position level) of the {voltage current} waveform for each element.	5-43
:DISPlay:WAVE<x>:POSition:{UALL IALL}	Sets the vertical position (center-position level) of the {voltage current} waveforms for all elements at once.	5-43
:DISPlay:WAVE<x>:SVALue	Sets or queries the on/off status of scale value display.	5-43
:DISPlay:WAVE<x>:TDIV	Sets or queries the waveform Time/div value.	5-43
:DISPlay:WAVE<x>:TLABel	Sets or queries the on/off status of waveform label name display.	5-43
:DISPlay:WAVE<x>:{U<x> I<x> SPEed<x> TORQue<x> AUX<x>}	Sets or queries the on/off status of {voltage current rotation speed torque auxiliary signal} waveform display.	5-44
:DISPlay:WAVE<x>:VZoom?	Queries all waveform vertical zoom factor settings.	5-44
:DISPlay:WAVE<x>:VZoom:{U<x> I<x>}	Sets or queries the vertical zoom factor of the {voltage current} waveform for each element.	5-44
:DISPlay:WAVE<x>:VZoom:{UALL IALL}	Sets the vertical zoom factor of the {voltage current} waveforms for all elements at once.	5-44

FILE Group

:FILE?	Queries all file operation settings.	5-50
:FILE:CDIRectory	Changes the current directory.	5-50
:FILE:COUNt?	Queries the number of subdirectories and files in the target directory.	5-50
:FILE:DELeTe:IMAGe:{BMP PNG JPEG}	Deletes a screen image data file.	5-50
:FILE:DELeTe:NUMeric:ASCIi	Deletes a numeric data file.	5-50
:FILE:DELeTe:SETup	Deletes a setup parameter file.	5-50
:FILE:DELeTe:STORE:{DATA HEADer}	Deletes a stored numeric data file.	5-50
:FILE:DELeTe:WAVE:ASCIi	Deletes a waveform display data file.	5-50
:FILE:DRIVE	Sets the current drive.	5-50
:FILE:FILTer	Sets or queries the file list filter.	5-50
:FILE:FREE?	Queries the free space (in bytes) on the current drive.	5-50
:FILE:ITEM	Sets the type of files to include in the file list.	5-51

5.1 Command List

Command	Explanation	Page
:FILE:LIST?	Queries information about subdirectories and files in the current directory (name, size, date and time, and attributes).	5-51
:FILE:LOAD:ABORT	Aborts file loading.	5-51
:FILE:LOAD:SETup	Loads a setup parameter file.	5-51
:FILE:PATH?	Queries the absolute path of the current directory.	5-51
:FILE:SAVE?	Queries all file save settings.	5-51
:FILE:SAVE:ABORT	Aborts file saving.	5-51
:FILE:SAVE:ANAMing	Sets or queries the auto naming function for saving files.	5-51
:FILE:SAVE:COMMeNt	Sets or queries the comment added to saved files.	5-51
:FILE:SAVE:NUMeric:ESElect?	Queries all settings for numeric data file saving (element selection method).	5-51
:FILE:SAVE:NUMeric:ESElect:ALL	Sets all output elements for all functions at once when saving numeric data files.	5-51
:FILE:SAVE:NUMeric:ESElect:<Function>	Sets or queries the output elements for each function when saving numeric data files.	5-51
:FILE:SAVE:NUMeric:ESElect:PRESet<x>	Presets the output elements of functions to a specified pattern when saving numeric data files.	5-52
:FILE:SAVE:NUMeric[:EXECute]	Saves numeric data to a file.	5-52
:FILE:SAVE:NUMeric:ITEM	Sets or queries the selection method for items to save when saving numeric data to a file.	5-52
:FILE:SAVE:NUMeric:NORMal?	Queries all settings for saving numeric data files (manual selection method for items to save).	5-52
:FILE:SAVE:NUMeric:NORMal:ALL	Sets the output on/off status for all elements and functions at once when saving numeric data to a file.	5-52
:FILE:SAVE:NUMeric:NORMal:{ELEMENT<x> SIGMA SIGMB}	Sets or queries the output on/off status for each element or wiring unit ({ΣA ΣB}) when saving numeric data to a file.	5-52
:FILE:SAVE:NUMeric:NORMal:<Function>	Sets or queries the output on/off status for each function when saving numeric data to a file.	5-52
:FILE:SAVE:NUMeric:NORMal:MOTor<x>	Sets or queries the output on/off status for each motor when saving numeric data to a file.	5-52
:FILE:SAVE:NUMeric:NORMal:PRESet<x>	Presets the output on/off status of elements and functions to a specified pattern when saving numeric data to a file.	5-53
:FILE:SAVE:NUMeric:NORMal:UNIT	Sets or queries the unit for saving numeric data to a file.	5-53
:FILE:SAVE:SETup[:EXECute]	Saves setup parameters to a file.	5-53
:FILE:SAVE:WAVE[:EXECute]	Saves waveform display data to a file.	5-53

HARMonics Group

:HARMonics<x>?	Queries all harmonic measurement settings.	5-54
:HARMonics<x>:CONFigure?	Queries the harmonic measurement groups for all elements.	5-54
:HARMonics<x>:CONFigure[:ALL]	Sets the harmonic measurement group for all elements at once.	5-54
:HARMonics<x>:CONFigure:ELEMeNt<x>	Sets or queries the harmonic measurement group for each element.	5-54
:HARMonics<x>:CONFigure:{SIGMA SIGMB}	Sets the harmonic measurement group for all elements belonging to wiring unit {ΣA ΣB} at once.	5-54
:HARMonics<x>:CPLLsource:SOURce	Sets or queries the common PLL source for multi-unit synchronization.	5-54
:HARMonics<x>:CPLLsource:UNIT	Sets or queries the common PLL unit for multi-unit synchronization.	5-54
:HARMonics<x>:ORDer	Sets or queries the minimum and maximum analysis harmonic orders.	5-55

Command	Explanation	Page
:HARMonics<x>:PLLSource	Sets or queries the PLL source.	5-55
:HARMonics<x>:THD	Sets or queries the formula for calculating THD (total harmonic distortion).	5-55

HISTory Group

:HISTory?	Queries all history function settings.	5-56
:HISTory:COMMUnicate?	Queries all history communication output settings.	5-56
:HISTory:COMMUnicate:COUNt	Sets or queries the history communication output count.	5-56
:HISTory:COMMUnicate:FORMat	Sets or queries the format of history communication output data.	5-56
:HISTory:COMMUnicate:HOLD	Sets or queries whether to hold (ON) or release (OFF) history communication output data.	5-56
:HISTory:COMMUnicate:OUTPutmode	Sets or queries the history communication output mode.	5-56
:HISTory:NUMeric:LIST:VALue?	Queries history communication numeric list data.	5-57
:HISTory:NUMeric:VALue?	Queries history communication numeric data.	5-57

HOLD Group

:HOLD	Sets or queries the hold function for output data (display data, communication data, etc.).	5-58
-------	---	------

IMAGe Group

:IMAGe?	Queries all settings for outputting screen image data.	5-59
:IMAGe:ABORt	Aborts output of screen image data.	5-59
:IMAGe:COLor	Sets or queries the color tone for saving screen image data.	5-59
:IMAGe:EXECute	Executes output of screen image data.	5-59
:IMAGe:FORMat	Sets or queries the format for saving screen image data.	5-59
:IMAGe:SAVE?	Queries all settings for saving screen image data.	5-59
:IMAGe:SAVE:ANAMing	Sets or queries the auto naming function for saving files.	5-59
:IMAGe:SAVE:CDIRectory	Changes the destination directory for saving screen image data.	5-59
:IMAGe:SAVE:DRIVE	Sets the destination drive for saving screen image data.	5-59
:IMAGe:SAVE:FREE?	Queries the free space (byte) on the destination drive for saving screen image data.	5-59
:IMAGe:SAVE:NAME	Sets or queries the save file name.	5-60
:IMAGe:SAVE:PATH?	Queries the absolute path of the destination for saving screen image data.	5-60
:IMAGe:SEND?	Queries screen image data.	5-60

INPut Group

:INPut?	Queries all input element settings.	5-61
[:INPut] :CFACtor	Sets or queries the crest factor.	5-61
[:INPut] :CORRection:AMPLitude?	Queries all amplitude correction settings.	5-61
[:INPut] :CORRection:AMPLitude:CURRent?	Queries all current amplitude correction settings.	5-61
[:INPut] :CORRection:AMPLitude:CURRent:RATio?	Queries the current amplitude correction ratios for all elements.	5-61
[:INPut] :CORRection:AMPLitude:CURRent:RATio[:ALL]	Sets the current amplitude correction ratio for all elements at once.	5-61

5.1 Command List

Command	Explanation	Page
[:INPut]:CORRection:AMPLitude:CURRent:RATio:ELEMeNt<x>	Sets or queries the current amplitude correction ratio for each element.	5-61
[:INPut]:CORRection:AMPLitude:CURRent:STATe?	Queries the current amplitude correction on/off status for all elements.	5-61
[:INPut]:CORRection:AMPLitude:CURRent[:STATe][:ALL]	Sets the current amplitude correction on/off status for all elements at once.	5-61
[:INPut]:CORRection:AMPLitude:CURRent[:STATe]:ELEMeNt<x>	Sets or queries the current amplitude correction on/off status for each element.	5-62
[:INPut]:CORRection:DC?	Queries all DC offset correction settings.	5-62
[:INPut]:CORRection:DC:CURRent?	Queries all current DC offset correction settings.	5-62
[:INPut]:CORRection:DC:CURRent:OFFSet?	Queries the current DC offset correction values for all elements.	5-62
[:INPut]:CORRection:DC:CURRent:OFFSet[:ALL]	Sets the current DC offset correction value for all elements at once.	5-62
[:INPut]:CORRection:DC:CURRent:OFFSet:ELEMeNt<x>	Sets or queries the current DC offset correction value for each element.	5-62
[:INPut]:CORRection:DC:CURRent:STATe?	Queries the current DC offset correction on/off status for all elements.	5-62
[:INPut]:CORRection:DC:CURRent[:STATe][:ALL]	Sets the current DC offset correction on/off status for all elements at once.	5-62
[:INPut]:CORRection:DC:CURRent[:STATe]:ELEMeNt<x>	Sets or queries the current DC offset correction on/off status for each element.	5-62
[:INPut]:CORRection:PHASe?	Queries all phase correction settings.	5-63
[:INPut]:CORRection:PHASe:CURRent?	Queries all current phase correction settings.	5-63
[:INPut]:CORRection:PHASe:CURRent:DPHase?	Queries the input-output phase difference for calculating the current phase correction value for all elements.	5-63
[:INPut]:CORRection:PHASe:CURRent:DPHase[:ALL]	Sets the input-output phase difference for calculating the current phase correction value for all elements at once.	5-63
[:INPut]:CORRection:PHASe:CURRent:DPHase:ELEMeNt<x>	Sets or queries the input-output phase difference for calculating the current phase correction value for each element.	5-63
[:INPut]:CORRection:PHASe:CURRent:DTIME?	Queries the input-output time difference (phase correction value) for current.	5-63
[:INPut]:CORRection:PHASe:CURRent:FREQuency?	Queries the frequency for calculating the current phase correction value for all elements.	5-63
[:INPut]:CORRection:PHASe:CURRent:FREQuency[:ALL]	Sets the frequency for calculating the current phase correction value for all elements at once.	5-63
[:INPut]:CORRection:PHASe:CURRent:FREQuency:ELEMeNt<x>	Sets or queries the frequency for calculating the current phase correction value for each element.	5-64
[:INPut]:CORRection:PHASe:CURRent:STATe?	Queries the current phase correction on/off status for all elements.	5-64
[:INPut]:CORRection:PHASe:CURRent[:STATe][:ALL]	Sets the current phase correction on/off status for all elements at once.	5-64
[:INPut]:CORRection:PHASe:CURRent[:STATe]:ELEMeNt<x>	Sets or queries the current phase correction on/off status for each element.	5-64
[:INPut]:CURRent?	Queries all current measurement settings.	5-64
[:INPut]:CURRent:AUTO?	Queries the auto current range on/off status for all elements.	5-64
[:INPut]:CURRent:AUTO[:ALL]	Sets the auto current range on/off status for all elements at once.	5-64
[:INPut]:CURRent:AUTO:ELEMeNt<x>	Sets or queries the auto current range on/off status for each element.	5-64

Command	Explanation	Page
[:INPut] :CURRent:AUTO: { SIGMA SIG MB }	Sets the auto current range on/off status at once for the elements belonging to wiring unit {ΣA ΣB}.	5-64
[:INPut] :CURRent:CONFig?	Queries the enabled current ranges for all elements.	5-65
[:INPut] :CURRent:CONFig[:ALL]	Sets the enabled current ranges for all elements at once.	5-65
[:INPut] :CURRent:CONFig:ELEMe nt<x>	Sets or queries the enabled current ranges for each element.	5-65
[:INPut] :CURRent:CT?	Queries all CT settings for current sensor elements.	5-65
[:INPut] :CURRent:CT:PRESet?	Queries the CT presets for all elements.	5-65
[:INPut] :CURRent:CT:PRESet[:ALL]	Sets the CT preset for all elements at once.	5-65
[:INPut] :CURRent:CT:PRESet:ELEMe nt<x>	Sets or queries the CT preset for each element.	5-65
[:INPut] :CURRent:CT:RESistance?	Queries the input resistances for all elements.	5-65
[:INPut] :CURRent:CT:RESistance[: ALL]	Sets the input resistance for all elements at once.	5-65
[:INPut] :CURRent:CT:RESistance:E LEMe nt<x>	Sets or queries the input resistance for each element.	5-66
[:INPut] :CURRent:EXTSensor?	Queries all external current sensor range settings.	5-66
[:INPut] :CURRent:EXTSensor:CONF ig?	Queries the enabled external current sensor ranges for all elements.	5-66
[:INPut] :CURRent:EXTSensor:CONF ig[:ALL]	Sets the enabled external current sensor ranges for all elements at once.	5-66
[:INPut] :CURRent:EXTSensor:CONF ig:ELEMe nt<x>	Sets or queries the enabled external current sensor ranges for each element.	5-66
[:INPut] :CURRent:EXTSensor:POJu mp?	Queries the range to jump to when a current peak-over occurs for all elements.	5-66
[:INPut] :CURRent:EXTSensor:POJum p[:ALL]	Sets the range to jump to when a current peak-over occurs for all elements at once.	5-66
[:INPut] :CURRent:EXTSensor:POJum p:ELEMe nt<x>	Sets or queries the range to jump to when a current peak-over occurs for each element.	5-67
[:INPut] :CURRent:POJump?	Queries the range to jump to when a current peak-over occurs for all elements.	5-67
[:INPut] :CURRent:POJump[:ALL]	Sets the range to jump to when a current peak-over occurs for all elements at once.	5-67
[:INPut] :CURRent:POJump:ELEMe nt<x>	Sets or queries the range to jump to when a current peak-over occurs for each element.	5-67
[:INPut] :CURRent:PROBe?	Queries all current probe settings for current sensor elements.	5-67
[:INPut] :CURRent:PROBe:PRESet?	Queries the current probe presets for all elements.	5-67
[:INPut] :CURRent:PROBe:PRESet[:A LL]	Sets the current probe preset for all elements at once.	5-67
[:INPut] :CURRent:PROBe:PRESet:EL EMe nt<x>	Sets or queries the current probe preset for each element.	5-67
[:INPut] :CURRent:PROBe:RATE?	Queries the output voltage rate (current probe mV/A ratio) for all elements.	5-68
[:INPut] :CURRent:PROBe:RATE[:A LL]	Sets the output voltage rate (current probe mV/A ratio) for all elements at once.	5-68
[:INPut] :CURRent:PROBe:RATE:ELEM ent<x>	Sets or queries the output voltage rate (current probe mV/A ratio) for each element.	5-68
[:INPut] :CURRent:RANGe?	Queries the current ranges for all elements.	5-68
[:INPut] :CURRent:RANGe[:ALL]	Sets the current range for all elements at once.	5-68

5.1 Command List

Command	Explanation	Page
[:INPut] :CURRent:RANGe:ELEMenT<x>	Sets or queries the current range for each element.	5-68
[:INPut] :CURRent:RANGe:{SIGMA SIGMB}	Sets the current range at once for the elements belonging to wiring unit {ΣA ΣB}.	5-68
[:INPut] :CURRent:SENSor:AUTO?	Queries the CT-ID automatic sensor on/off status for all elements.	5-69
[:INPut] :CURRent:SENSor:AUTO[:ALL]	Sets the CT-ID automatic sensor on/off status for all elements at once.	5-69
[:INPut] :CURRent:SENSor:AUTO:ELEMenT<x>	Sets or queries the CT-ID automatic sensor on/off status for each element.	5-69
[:INPut] :CURRent:SENSor:MODEl?	Queries the sensor models for all elements.	5-69
[:INPut] :CURRent:SENSor:MODEl[:ALL]	Sets the sensor model for all elements at once.	5-69
[:INPut] :CURRent:SENSor:MODEl:ELEMenT<x>	Sets or queries the sensor model for each element.	5-69
[:INPut] :CURRent:SENSor:OTYPE?	Queries the sensor output types for all elements.	5-69
[:INPut] :CURRent:SENSor:OTYPE[:ALL]	Sets the sensor output type for all elements at once.	5-69
[:INPut] :CURRent:SENSor:OTYPE:ELEMenT<x>	Sets or queries the sensor output type for each element.	5-70
[:INPut] :CURRent:SENSor:OTYPE:{SIGMA SIGMB}	Sets the sensor output type at once for the elements belonging to wiring unit {ΣA ΣB}.	5-70
[:INPut] :DEMAg:CTRL:IALL	Sets whether current is included in demagnetization operation for all elements at once.	5-70
[:INPut] :DEMAg:CTRL:I<x>	Sets or queries whether current is included in demagnetization operation for each element.	5-70
[:INPut] :DEMAg:EXECute	Executes demagnetization.	5-70
[:INPut] :ESElect	Sets or queries the element for setting the measurement range.	5-70
[:INPut] :FILTer?	Queries all input filter settings.	5-71
[:INPut] :FILTer:FREQuency?	Queries all frequency filter settings.	5-71
[:INPut] :FILTer:FREQuency:ADVanced	Sets or queries the on/off status of frequency filter advanced settings.	5-71
[:INPut] :FILTer:FREQuency[:ALL]	Sets the frequency filter for all elements at once.	5-71
[:INPut] :FILTer:FREQuency:ELEMenT<x>	Sets or queries the frequency filter for each element.	5-71
[:INPut] :FILTer:FREQuency:LPFilt er?	Queries all low-pass filter settings. (Advanced settings)	5-71
[:INPut] :FILTer:FREQuency:LPFilt er[:ALL]	Sets the low-pass filter for all elements at once. (Advanced settings)	5-71
[:INPut] :FILTer:FREQuency:LPFilt er:AUTO?	Queries the auto frequency filter on/off status for all elements.	5-71
[:INPut] :FILTer:FREQuency:LPFilt er:AUTO[:ALL]	Sets the auto frequency filter on/off status for all elements at once.	5-71
[:INPut] :FILTer:FREQuency:LPFilt er:AUTO:ELEMenT<x>	Sets or queries the auto frequency filter on/off status for each element.	5-71
[:INPut] :FILTer:FREQuency:LPFilt er:ELEMenT<x>	Sets or queries the low-pass filter for each element. (Advanced settings)	5-72
[:INPut] :FILTer:LINE?	Queries all line filter settings.	5-72
[:INPut] :FILTer[:LINE]:AAFilter?	Queries all anti-aliasing filter settings. (Advanced settings)	5-72

Command	Explanation	Page
[:INPut]:FILTer[:LINE]:AAFilter[:ALL]	Sets the anti-aliasing filter for all elements at once. (Advanced settings)	5-72
[:INPut]:FILTer[:LINE]:AAFilter:ELEMeNt<x>	Sets or queries the anti-aliasing filter for each element. (Advanced settings)	5-72
[:INPut]:FILTer[:LINE]:ADVanced	Sets or queries the on/off status of line filter advanced settings.	5-72
[:INPut]:FILTer[:LINE][:ALL]	Sets the line filter for all elements at once.	5-72
[:INPut]:FILTer[:LINE]:ELEMeNt<x>	Sets or queries the line filter for each element.	5-73
[:INPut]:FILTer[:LINE]:DLFilter?	Queries all digital line filter settings. (Advanced settings)	5-73
[:INPut]:FILTer[:LINE]:DLFilter:{NORmal HARMonics}?	Queries all digital line filter settings for {normal measurement harmonic measurement}. (Advanced settings)	5-73
[:INPut]:FILTer[:LINE]:DLFilter:{NORmal HARMonics}[:ALL]	Sets the digital line filter for {normal measurement harmonic measurement} for all elements at once. (Advanced settings)	5-73
[:INPut]:FILTer[:LINE]:DLFilter:{NORmal HARMonics}:ELEMeNt<x>	Sets or queries the digital line filter for {normal measurement harmonic measurement} for each element. (Advanced settings)	5-73
[:INPut]:FILTer[:LINE]:DLFilter:HARMonics:AUTO?	Queries the auto harmonic filter on/off status for all elements.	5-73
[:INPut]:FILTer[:LINE]:DLFilter:HARMonics:AUTO[:ALL]	Sets the auto harmonic filter on/off status for all elements at once.	5-73
[:INPut]:FILTer[:LINE]:DLFilter:HARMonics:AUTO:ELEMeNt<x>	Sets or queries the auto harmonic filter on/off status for each element.	5-74
[:INPut]:MODUle?	Queries the input element type.	5-74
[:INPut]:MTR<x>[:CONFig]	Sets or queries the configuration of MTR1 and MTR2.	5-74
[:INPut]:NULL:CONDition:{SPEed<x> TORQue<x> AUX<x>}?	Queries the NULL operation status of {rotation speed torque AUX}.	5-75
[:INPut]:NULL:CONDition:{U<x> I<x>}?	Queries the NULL operation status of {voltage current} for each element.	5-75
[:INPut]:NULL:CTRL:{SPEed<x> TORQue<x> AUX<x>}	Sets or queries whether {rotation speed torque AUX} is included in NULL operation.	5-75
[:INPut]:NULL:CTRL:{U<x> I<x>}	Sets or queries whether {voltage current} for each element is included in NULL operation.	5-75
[:INPut]:NULL:NOTify	Sets or queries the on/off status of the NULL notification function after the warm-up time has elapsed.	5-76
[:INPut]:NULL[:STATe]	Sets or queries the NULL function on/off status.	5-76
[:INPut]:NULL:TARGet?	Queries all settings for NULL operation targets.	5-76
[:INPut]:NULL:TARGet[:MODE]	Sets or queries the selection mode for items to include in NULL operation.	5-76
[:INPut]:NULL:TARGet:{SPEed<x> TORQue<x> AUX<x>}	Sets or queries the NULL operation setting for {rotation speed torque AUX}.	5-76
[:INPut]:NULL:TARGet:{U<x> I<x>}	Sets or queries the NULL operation setting for {voltage current} for each element.	5-76
[:INPut]:NULL:TARGet:{UALL IALl}	Sets the NULL operation setting for {voltage current} for all elements at once.	5-77
[:INPut]:NULL:VALue:{SPEed<x> TORQue<x> AUX<x>}	Sets or queries the NULL operation status NEW/HOLD of {rotation speed torque AUX}.	5-77
[:INPut]:NULL:VALue:{U<x> I<x>}	Sets or queries the NULL operation status NEW/HOLD of {voltage current} for each element.	5-77
[:INPut]:POVer?	Queries peak-over information.	5-77
[:INPut]:RLINKage	Sets or queries the on/off status of range Σ link, which links measurement ranges within wiring unit Σ .	5-77

5.1 Command List

Command	Explanation	Page
[:INPut] :SCALing?	Queries all scaling settings.	5-78
[:INPut] :SCALing:STATe?	Queries the scaling on/off status for all elements.	5-78
[:INPut] :SCALing[:STATe] [:ALL]	Sets the scaling on/off status for all elements at once.	5-78
[:INPut] :SCALing[:STATe]:ELEMeNt <x>	Sets or queries the scaling on/off status for each element.	5-78
[:INPut] :SCALing:{VT CT SFACt or}?	Queries the {VT ratio CT ratio power coefficient} for all elements.	5-78
[:INPut] :SCALing:{VT CT SFACtor} [:ALL]	Sets the {VT ratio CT ratio power coefficient} for all elements at once.	5-78
[:INPut] :SCALing:{VT CT SFACtor} :ELEMeNt<x>	Sets or queries the {VT ratio CT ratio power coefficient} for each element.	5-78
[:INPut] :SCALing:{VT CT SFACtor} :{SIGMA SIGMB}	Sets the {VT ratio CT ratio power coefficient} at once for the elements belonging to wiring unit {ΣA ΣB}.	5-78
[:INPut] :SYNChronize?	Queries the sync sources for all elements.	5-78
[:INPut] :SYNChronize[:ALL]	Sets the sync source for all elements at once.	5-78
[:INPut] :SYNChronize:CZCRoss:SOU Rce	Sets or queries the common zero-cross source for multi-unit synchronization.	5-79
[:INPut] :SYNChronize:CZCRoss:UN IT	Sets or queries the common zero-cross unit for multi-unit synchronization.	5-79
[:INPut] :SYNChronize:ELEMeNt<x>	Sets or queries the sync source for each element.	5-79
[:INPut] :SYNChronize:{SIGMA SIG MB}	Sets the sync source at once for the elements belonging to wiring unit {ΣA ΣB}.	5-79
[:INPut] :SYNChronize:LEVel?	Queries all sync source level settings.	5-80
[:INPut] :SYNChronize:LEVel:{VOLT age CURRent} [:ALL]	Sets the {voltage current} sync source level for all elements at once.	5-80
[:INPut] :SYNChronize:LEVel:{VOLT age CURRent}:ELEMeNt<x>	Sets or queries the {voltage current} sync source level for each element.	5-80
[:INPut] :SYNChronize:RECTifier?	Queries all sync source rectifier on/off settings.	5-80
[:INPut] :SYNChronize:RECTifier:{ VOLTage CURRent} [:ALL]	Sets {voltage current} sync source rectifier on/off for all elements at once.	5-80
[:INPut] :SYNChronize:RECTifier:{ VOLTage CURRent}:ELEMeNt<x>	Sets or queries the on/off status of sync source rectification for {voltage current} for each element.	5-80
[:INPut] :VOLTage?	Queries all voltage measurement settings.	5-80
[:INPut] :VOLTage:AUTO?	Queries the auto voltage range on/off status for all elements.	5-80
[:INPut] :VOLTage:AUTO[:ALL]	Sets the auto voltage range on/off status for all elements at once.	5-80
[:INPut] :VOLTage:AUTO:ELEMeNt<x>	Sets or queries the auto voltage range on/off status for each element.	5-80
[:INPut] :VOLTage:AUTO:{SIGMA SIG MB}	Sets the auto voltage range on/off status at once for the elements belonging to wiring unit {ΣA ΣB}.	5-81
[:INPut] :VOLTage:CONFig?	Queries the enabled voltage ranges for all elements.	5-81
[:INPut] :VOLTage:CONFig[:ALL]	Sets the enabled voltage ranges for all elements at once.	5-81
[:INPut] :VOLTage:CONFig:ELEMeNt <x>	Sets or queries the enabled voltage ranges for each element.	5-81
[:INPut] :VOLTage:POJump?	Queries the range to jump to when a voltage peak-over occurs for all elements.	5-81
[:INPut] :VOLTage:POJump[:ALL]	Sets the range to jump to when a voltage peak-over occurs for all elements at once.	5-81
[:INPut] :VOLTage:POJump:ELEMeNt <x>	Sets or queries the range to jump to when a voltage peak-over occurs for each element.	5-82
[:INPut] :VOLTage:RANGe?	Queries the voltage ranges for all elements.	5-82

Command	Explanation	Page
[:INPut]:VOLTagE:RANGe[:ALL]	Sets the voltage range for all elements at once.	5-82
[:INPut]:VOLTagE:RANGe:ELEMent<x>	Sets or queries the voltage range for each element.	5-82
[:INPut]:VOLTagE:RANGe:{SIGMA SIGMA SI GMB}	Sets the voltage range at once for the elements belonging to wiring unit {ΣA ΣB}.	5-82
[:INPut]:WIRing	Sets or queries the wiring system.	5-83

INTEGrate Group

:INTEGrate?	Queries all integration settings.	5-84
:INTEGrate:ACAL	Sets or queries the integration auto calibration on/off status.	5-84
:INTEGrate:INdependent	Sets or queries the on/off status of independent integration by element.	5-84
:INTEGrate:MODE	Sets or queries the integration mode.	5-84
:INTEGrate:QMODE?	Queries the current mode for current integration for all elements.	5-84
:INTEGrate:QMODE[:ALL]	Sets the current mode for current integration for all elements at once.	5-84
:INTEGrate:QMODE:ELEMent<x>	Sets or queries the current mode for current integration for each element.	5-84
:INTEGrate:RACTION	Sets or queries the integration restart operation when power is restored during integration.	5-84
:INTEGrate:RESet	Resets integration values.	5-85
:INTEGrate:RTAll:{START END}	Sets the reserved integration {start end} time for all elements at once in real-time integration mode.	5-85
:INTEGrate:RTIME<x>?	Queries the scheduled integration start/end times in real-time integration mode.	5-85
:INTEGrate:RTIME<x>:{START END}	Sets or queries the scheduled integration {start end} time in real-time integration mode.	5-85
:INTEGrate:START	Starts integration.	5-85
:INTEGrate:STATe?	Queries the integration status.	5-86
:INTEGrate:STOP	Stops integration.	5-86
:INTEGrate:TIMer<x>	Sets or queries the integration timer duration.	5-86
:INTEGrate:TMAll	Sets the integration timer duration for all elements at once.	5-86
:INTEGrate:WPTYPE?	Queries the watt-hour computation method by polarity (WP+/WP-) for all elements.	5-86
:INTEGrate:WPTYPE[:ALL]	Sets the watt-hour computation method by polarity (WP+/WP-) for all elements at once.	5-86
:INTEGrate:WPTYPE:ELEMent<x>	Sets or queries the watt-hour computation method by polarity (WP+/WP-) for each element.	5-86

MEASure Group

:MEASure?	Queries all computation settings.	5-87
:MEASure:AVERaging?	Queries all averaging settings.	5-87
:MEASure:AVERaging:COUNT	Sets or queries the averaging coefficient.	5-87
:MEASure:AVERaging[:STATe]	Sets or queries the averaging on/off status.	5-87
:MEASure:AVERaging:TYPE	Sets or queries the averaging type.	5-87
:MEASure:EFFiciency?	Queries all efficiency computation settings.	5-87
:MEASure:EFFiciency:ETA<x>[:FUNCTION]	Sets or queries the efficiency formula.	5-87
:MEASure:EFFiciency:ETA<x>:UNIT	Sets or queries the units used in the efficiency formula.	5-88

5.1 Command List

Command	Explanation	Page
:MEASure:EFFiciency:UDEF<x>[:FUN Ction]	Sets or queries the user-defined parameters used in the efficiency formula.	5-88
:MEASure:EFFiciency:UDEF<x>:UNIT	Sets or queries the units for the user-defined parameters used in the efficiency formula.	5-89
:MEASure:EVENT<x>?	Queries all user-defined event settings.	5-89
:MEASure:EVENT<x>:EXPRession?	Queries all settings for the user-defined event conditional expression.	5-89
:MEASure:EVENT<x>:EXPRession:CON Dition	Sets or queries the user-defined event conditional expression (condition-combination type).	5-89
:MEASure:EVENT<x>:EXPRession:INV erse	Sets or queries the user-defined event conditional expression (compound condition type).	5-89
:MEASure:EVENT<x>:EXPRession:ITE M[:FUNCTION]	Sets or queries the item used in the user-defined event conditional expression (range type).	5-90
:MEASure:EVENT<x>:EXPRession:ITE M:UNIT	Sets or queries the unit for the item used in the user-defined event conditional expression (range type).	5-90
:MEASure:EVENT<x>:EXPRession:LIM it<x>	Sets or queries the range limit for the user-defined event conditional expression (range type).	5-90
:MEASure:EVENT<x>:EXPRession:STR ing?	Queries the user-defined event conditional expression in string format.	5-91
:MEASure:EVENT<x>:EXPRession:TYP E	Sets or queries the user-defined event conditional expression type.	5-91
:MEASure:EVENT<x>:FLABel	Sets or queries the display data string used when the user-defined event condition is not met (False).	5-91
:MEASure:EVENT<x>:NAME	Sets or queries the user-defined event name.	5-91
:MEASure:EVENT<x>[:STATe]	Sets or queries whether the user-defined event is enabled (ON) or disabled (OFF).	5-91
:MEASure:EVENT<x>:TLABel	Sets or queries the display data string used when the user-defined event condition is met (True).	5-91
:MEASure:FUNCTion<x>?	Queries all user-defined function settings.	5-91
:MEASure:FUNCTion<x>:EXPRession	Sets or queries the user-defined function expression.	5-91
:MEASure:FUNCTion<x>:NAME	Sets or queries the user-defined function name.	5-91
:MEASure:FUNCTion<x>:PRESet[:EXE Cute]	Executes the user-defined function preset.	5-92
:MEASure:FUNCTion<x>:PRESet:FILE :LOAD	Loads a user-defined function file.	5-92
:MEASure:FUNCTion<x>:PRESet:FILE :SAVE	Saves the user-defined function contents to a file.	5-92
:MEASure:FUNCTion<x>[:STATe]	Sets or queries whether the user-defined function is enabled (ON) or disabled (OFF).	5-92
:MEASure:FUNCTion<x>:UNIT	Sets or queries the unit appended to the user-defined function computation result.	5-92
:MEASure:MHOLd	Sets or queries whether the MAX HOLD function used in user-defined functions is enabled (ON) or disabled (OFF).	5-92
:MEASure:PC?	Queries all settings related to Pc (Corrected Power) computation.	5-93
:MEASure:PC:FORMula	Sets or queries the Pc (Corrected Power) formula.	5-93
:MEASure:PC:P<x>	Sets or queries a parameter for Pc (Corrected Power) computation.	5-93
:MEASure:PHASe	Sets or queries the phase difference display format.	5-93
:MEASure:PHASe:POLarity	Sets or queries the phase difference polarity.	5-93
:MEASure:SFORMula	Sets or queries the formula for S (apparent power).	5-93

Command	Explanation	Page
:MEASure:SQFormula	Sets or queries the formulas for S (apparent power) and Q (reactive power).	5-93
:MEASure:SYNChronize	Sets or queries the synchronous measurement mode.	5-93
:MEASure:UFFormula	Sets or queries the unbalance factor formula.	5-93

MOTor Group

:MOTor<x>?	Queries all settings for the motor evaluation function.	5-94
:MOTor<x>:EANGLE?	Queries all settings for electrical angle measurement.	5-94
:MOTor<x>:EANGLE:CORRection?	Queries all settings related to the electrical angle correction value setting.	5-94
:MOTor<x>:EANGLE:CORRection:AENTER?	Queries all settings related to automatic entry of the electrical angle correction value.	5-94
:MOTor<x>:EANGLE:CORRection:AENTER[:EXECute]	Executes automatic entry of the electrical angle correction value.	5-94
:MOTor<x>:EANGLE:CORRection:AENTER:TARGET	Sets or queries the target source for automatic entry of the electrical angle correction value.	5-94
:MOTor<x>:EANGLE:CORRection:CLEAR	Clears the electrical angle correction value.	5-94
:MOTor<x>:EANGLE:CORRection[:VALUE]	Sets or queries the electrical angle correction value.	5-94
:MOTor<x>:EANGLE:HTRigger	Sets or queries the trigger for electrical angle measurement.	5-95
:MOTor<x>:EANGLE[:STATE]	Sets or queries the electrical angle measurement on/off status.	5-95
:MOTor<x>:EANGLE:TRIGger:CZPHase:SOURce	Sets or queries the common Z Phase source for multi-unit synchronization.	5-95
:MOTor<x>:EANGLE:TRIGger:CZPHase:UNIT	Sets or queries the common Z Phase unit for multi-unit synchronization.	5-95
:MOTor<x>:FILTer?	Queries all input filter settings.	5-95
:MOTor<x>:FILTer[:LINE]	Sets or queries the line filter.	5-95
:MOTor<x>:FILTer:NOISe:SPEEd	Sets or queries the noise filter for the rotation signal (pulse input method).	5-96
:MOTor<x>:FILTer:NOISe:TORQue	Sets or queries the noise filter for the torque signal (pulse input method).	5-96
:MOTor<x>:PM?	Queries all settings related to motor output (Pm).	5-96
:MOTor<x>:PM:SCALing	Sets or queries the scaling coefficient for motor output computation.	5-96
:MOTor<x>:PM:UNIT	Sets or queries the unit appended to the motor output computation result.	5-96
:MOTor<x>:POLE	Sets or queries the number of motor poles.	5-96
:MOTor<x>:RESolver?	Queries all resolver calculation settings.	5-97
:MOTor<x>:RESolver:AUTO	Sets or queries the auto voltage range on/off status for resolver calculation.	5-97
:MOTor<x>:RESolver:EAAVeraging	Sets or queries the electrical angle averaging on/off status for resolver calculation.	5-97
:MOTor<x>:RESolver:HYSTeresis	Sets or queries the carrier zero-crossing detection hysteresis for resolver calculation.	5-97
:MOTor<x>:RESolver:LFIlter	Sets or queries the line filter for resolver calculation.	5-97
:MOTor<x>:RESolver:RPPair	Sets or queries the resolver pole pairs for resolver calculation.	5-97
:MOTor<x>:RESolver:PRANge	Sets or queries the range for resolver calculation.	5-97
:MOTor<x>:RESolver:RANGe	Queries all voltage range settings for resolver calculation.	5-97

5.1 Command List

Command	Explanation	Page
:MOTor<x>:RESolver:RANGe[:CARRie r]	Sets or queries the voltage range of the excitation signal for resolver calculation.	5-97
:MOTor<x>:RESolver:RANGe:COsine	Sets or queries the voltage range of the cosine signal for resolver calculation.	5-98
:MOTor<x>:RESolver:RANGe:LiNKage	Sets or queries the on/off status of voltage range linkage for resolver calculation.	5-98
:MOTor<x>:RESolver:RANGe:SiNE	Sets or queries the voltage range of the sine signal for resolver calculation.	5-98
:MOTor<x>:RESolver:SCALing	Sets or queries the scaling factor for resolver calculation.	5-98
:MOTor<x>:RESolver:TFiLter	Sets or queries the tracking filter for resolver calculation.	5-98
:MOTor<x>:RESolver:UNIT	Sets or queries the unit appended to resolver calculation results.	5-98
:MOTor<x>:SPeEd?	Queries all rotation speed (Speed) settings.	5-99
:MOTor<x>:SPeEd:AUTO	Sets or queries the auto voltage range on/off status for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale?	Queries all linear scaling settings for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale:AVALue	Sets or queries slope A of the linear scale for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale:BVALue	Sets or queries offset value B of the linear scale for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale:CALCula te?	Queries all linear scale parameter calculation settings for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale:CALCulate :{P1X P1Y P2X P2Y}	Sets or queries the {Point1X Point1Y Point2X Point2Y} data for linear scale parameter calculation for the rotation signal (analog input type).	5-99
:MOTor<x>:SPeEd:LSCale:CALCulate :EXECute	Executes linear scale parameter calculation for the rotation signal (analog input type).	5-100
:MOTor<x>:SPeEd:PRANGe	Sets or queries the range for the rotation signal (pulse input type).	5-100
:MOTor<x>:SPeEd:PULSe	Sets or queries the pulse count for the rotation signal (pulse input type).	5-100
:MOTor<x>:SPeEd:RANGe	Sets or queries the voltage range for the rotation signal (analog input type).	5-100
:MOTor<x>:SPeEd:SCALing	Sets or queries the scaling factor for rotation speed calculation.	5-100
:MOTor<x>:SPeEd:TYPE	Sets or queries the input type for the rotation signal.	5-100
:MOTor<x>:SPeEd:UNIT	Sets or queries the unit appended to rotation speed calculation results.	5-101
:MOTor<x>:SSPeEd	Sets or queries the frequency measurement source for sync speed (SyncSp) calculation.	5-101
:MOTor<x>:SYNChronize	Sets or queries the sync source for rotation speed (Speed) and torque (Torque) calculation.	5-101
:MOTor<x>:TORQue?	Queries all torque (Torque) settings.	5-101
:MOTor<x>:TORQue:AUTO	Sets or queries the auto voltage range on/off status for the torque signal (analog input type).	5-101
:MOTor<x>:TORQue:LSCale?	Queries all linear scaling settings for the torque signal (analog input type).	5-101
:MOTor<x>:TORQue:LSCale:AVALue	Sets or queries slope A of the linear scale for the torque signal (analog input type).	5-101
:MOTor<x>:TORQue:LSCale:BVALue	Sets or queries offset value B of the linear scale for the torque signal (analog input type).	5-102

Command	Explanation	Page
:MOTor<x>:TORQue:LSCale:CALCulate?	Queries all linear scale parameter calculation settings for the torque signal (analog input type).	5-102
:MOTor<x>:TORQue:LSCale:CALCulate:{P1X P1Y P2X P2Y}	Sets or queries the {Point1X Point1Y Point2X Point2Y} data for linear scale parameter calculation for the torque signal (analog input type).	5-102
:MOTor<x>:TORQue:LSCale:CALCulate:EXECute	Executes linear scale parameter calculation for the torque signal (analog input type).	5-102
:MOTor<x>:TORQue:PRANge	Sets or queries the range for the torque signal (pulse input type).	5-102
:MOTor<x>:TORQue:RANge	Sets or queries the voltage range for the torque signal (analog input type).	5-102
:MOTor<x>:TORQue:RATE?	Queries all rated value settings for the torque signal (pulse input type).	5-103
:MOTor<x>:TORQue:RATE:{UPPer LOWer}	Sets or queries the {upper lower} rated value for the torque signal (pulse input type).	5-103
:MOTor<x>:TORQue:SCALing	Sets or queries the scaling factor for torque calculation.	5-103
:MOTor<x>:TORQue:TYPE	Sets or queries the input type for the torque signal.	5-103
:MOTor<x>:TORQue:UNIT	Sets or queries the unit appended to torque calculation results.	5-103

NUMeric Group

:NUMeric?	Queries all numeric data output information.	5-104
:NUMeric:BYTeorder	Sets or queries the output byte order for numeric data (FLOAT format).	5-104
:NUMeric:FORMat	Sets or queries the numeric data format.	5-104
:NUMeric:HOLD	Sets or queries whether to hold (ON) or release (OFF) all numeric data.	5-104
:NUMeric:LIST?	Queries all settings related to harmonic measurement numeric list data output.	5-105
:NUMeric:LIST:CLEar	Clears (sets to NONE) the output items for harmonic measurement numeric list data.	5-105
:NUMeric:LIST:DELeTe	Deletes output items for harmonic measurement numeric list data.	5-105
:NUMeric:LIST:ITEM<x>[:FUNction]	Sets or queries the output item (function and element) for harmonic measurement numeric list data.	5-105
:NUMeric:LIST:ITEM<x>:UNIT	Sets or queries the unit for harmonic measurement numeric data output items.	5-105
:NUMeric:LIST:NUMber	Sets or queries the number of numeric list data items sent by :NUMeric:LIST:VALue?.	5-105
:NUMeric:LIST:ORDeR	Sets or queries the maximum output harmonic order for harmonic measurement numeric list data.	5-106
:NUMeric:LIST:PRESet	Presets the output items for harmonic measurement numeric list data to a fixed pattern.	5-106
:NUMeric:LIST:SELeCt	Sets or queries the output components for harmonic measurement numeric list data.	5-106
:NUMeric:LIST:VALue?	Queries harmonic measurement numeric list data.	5-106
:NUMeric:NORMal?	Queries all numeric data output settings.	5-106
:NUMeric[:NORMal]:CLEar	Clears (sets to NONE) the numeric data output items.	5-107
:NUMeric[:NORMal]:DELeTe	Deletes numeric data output items.	5-107
:NUMeric[:NORMal]:HEADer?	Queries numeric data headers.	5-107
:NUMeric[:NORMal]:ITEM<x>[:FUNction]	Sets or queries the numeric data output item (function, element, and harmonic order).	5-107

5.1 Command List

Command	Explanation	Page
:NUMeric[:NORMal]:ITEM<x>:UNIT	Sets or queries the unit for numeric data output items.	5-107
:NUMeric[:NORMal]:NUMBER	Sets or queries the number of numeric data items sent by :NUMeric[:NORMal]:VALue?.	5-108
:NUMeric[:NORMal]:PRESet	Presets the numeric data output items to a fixed pattern.	5-108
:NUMeric[:NORMal]:VALue?	Queries numeric data.	5-108

STATus Group

:STATus?	Queries all settings for the communication status function.	5-114
:STATus:CONDition?	Queries the contents of the condition register.	5-114
:STATus:EESe	Sets or queries the extended event enable register.	5-114
:STATus:EESR?	Queries the contents of the extended event register and clears the register.	5-114
:STATus:ERRor?	Queries the code and message of the error that occurred (the first entry in the error queue).	5-114
:STATus:FILTer<x>	Sets or queries the transition filter.	5-114
:STATus:QENable	Sets or queries whether messages other than errors are stored in the error queue (ON) or not stored (OFF).	5-114
:STATus:QMESsage	Sets or queries whether to include the message in the response to :STATus:ERRor? (ON) or not include it (OFF).	5-114
:STATus:SPOLl?	Executes a serial poll.	5-114

STORe Group

:STORe?	Queries all numeric data storage settings.	5-115
:STORe:CONDition?	Queries the storage state.	5-115
:STORe:COUNt	Sets or queries the storage count.	5-115
:STORe:END	Ends numeric data storage.	5-115
:STORe:FILE?	Queries all settings for saving data that has been stored.	5-115
:STORe:FILE:ANAMing	Sets or queries the auto naming function for saving stored numeric data files.	5-115
:STORe:FILE:CDIRectory	Changes the save destination directory for stored numeric data.	5-115
:STORe:FILE:CONVert?	Queries all settings for converting stored numeric data files to CSV format.	5-115
:STORe:FILE:CONVert:ABORT	Aborts conversion of stored numeric data files to CSV format.	5-115
:STORe:FILE:CONVert:AUTO	Sets or queries the automatic conversion on/off status for stored numeric data files to CSV format.	5-115
:STORe:FILE:CONVert:EXECute	Executes conversion of a stored numeric data file to CSV format.	5-116
:STORe:FILE:DRIVE	Sets the save destination drive for stored numeric data.	5-116
:STORe:FILE:FREE?	Queries the free space (byte) on the save destination drive for stored numeric data.	5-116
:STORe:FILE:NAME	Sets or queries the save file name for stored numeric data.	5-116
:STORe:FILE:PATH?	Queries the absolute path of the save destination for stored numeric data.	5-116
:STORe:INTerval	Sets or queries the storage interval.	5-116
:STORe:NUMeric?	Queries all settings for numeric data storage items.	5-116
:STORe:NUMeric:ESeLect?	Queries all settings for numeric data storage items (element-specific selection method).	5-116
:STORe:NUMeric:ESeLect:ALL	Sets the output elements for all functions at once when storing numeric data.	5-116

Command	Explanation	Page
:STORe:NUMeric:ESElect:<Function>	Sets or queries the output elements for each function when storing numeric data.	5-116
:STORe:NUMeric:ESElect:PRESet<x>	Presets the output elements of functions to a predefined pattern when storing numeric data.	5-116
:STORe:NUMeric:ITEM	Sets or queries how numeric data storage items are selected.	5-117
:STORe:NUMeric:NORMal?	Queries all settings for numeric data storage items (manual selection method).	5-117
:STORe:NUMeric[:NORMal]:ALL	Sets the output on/off status for all elements and functions at once when storing numeric data.	5-117
:STORe:NUMeric[:NORMal]:{ELEMENT<x> SIGMA SIGMB}	Sets or queries the output on/off status for each element or wiring unit {ΣA ΣB} when storing numeric data.	5-117
:STORe:NUMeric[:NORMal]:<Function>	Sets or queries the output on/off status for each function when storing numeric data.	5-117
:STORe:NUMeric[:NORMal]:MOTor<x>	Sets or queries the output on/off status for each motor when storing numeric data.	5-117
:STORe:NUMeric[:NORMal]:PRESet<x>	Presets the output on/off status for elements and functions to a predefined pattern when storing numeric data.	5-117
:STORe:NUMeric[:NORMal]:UNIT	Sets or queries the unit for storing numeric data.	5-117
:STORe:NUMeric:TSTamp	Sets or queries the timestamp acquisition timing when storing numeric data.	5-118
:STORe:PAUSE	Stops numeric data storage.	5-118
:STORe:RECORD	Starts numeric data storage.	5-118
:STORe:RTIME?	Queries the scheduled time for starting/ending storage in real-time mode.	5-118
:STORe:RTIME:{START END}	Sets or queries the scheduled time for storage {start end} in real-time mode.	5-118
:STORe:SASStart	Sets or queries whether to store numeric data at the start of storage.	5-118
:STORe:SMODE	Sets or queries the storage mode.	5-118
:STORe:TEVent	Sets or queries the event that triggers event-synchronized storage mode.	5-118

SYSTem Group

:SYSTem?	Queries all system settings.	5-119
:SYSTem:CLOCK?	Queries all date and time settings.	5-119
:SYSTem:CLOCK:DISPlay	Sets or queries the on/off status of the date and time display.	5-119
:SYSTem:CLOCK:FORMat	Sets or queries the date display order.	5-119
:SYSTem:CLOCK:SNTP?	Queries all SNTP date and time settings.	5-119
:SYSTem:CLOCK:SNTP[:EXECute]	Sets the date and time using SNTP.	5-119
:SYSTem:CLOCK:SNTP:GMTTime	Sets or queries the time difference from Greenwich Mean Time.	5-119
:SYSTem:CLOCK:TYPE	Sets or queries the method for setting the date and time.	5-119
:SYSTem:CTPower:PPOWer:ELEMent<x>:STATus?	Queries the current probe power supply status for the element.	5-119
:SYSTem:CTPower:SCONnect:ELEMent<x>:STATus?	Queries the current sensor connection status.	5-120
:SYSTem:CTPower:SPOWer:ELEMent<x>:STATus?	Queries the current sensor power supply status for the element.	5-120
:SYSTem:DATE	Sets or queries the date.	5-120
:SYSTem:DFLow:FREQuency	Sets or queries how frequency data is displayed during low-frequency input (no input).	5-120

5.1 Command List

Command	Explanation	Page
:SYSTem:DFlow:MOToR	Sets or queries how motor data is displayed when there is no pulse input.	5-120
:SYSTem:DPOint	Sets or queries the type of decimal point used when saving various types of data in ASCII (CSV) format.	5-120
:SYSTem:EClear	Clears the error message displayed on the screen.	5-120
:SYSTem:ELEMeNt<x>?	Queries all element settings.	5-120
:SYSTem:ELEMeNt<x>:CTID?	Queries the ID of the CT series installed in the element.	5-120
:SYSTem:HOSTname	Sets or queries the host name.	5-121
:SYSTem:KLOCK	Sets or queries the key lock on/off status.	5-121
:SYSTem:LANGuage?	Queries all screen display language settings.	5-121
:SYSTem:LANGuage:MENU	Sets or queries the menu language.	5-121
:SYSTem:LANGuage:MESSAge	Sets or queries the message language.	5-121
:SYSTem:LCD?	Queries all screen settings used when operating this instrument with an external monitor or the Web server function.	5-121
:SYSTem:LCD:COLor?	Queries all display color settings for the screen used when operating this instrument with an external monitor or the Web server function.	5-121
:SYSTem:LCD:COLor:GRAPh?	Queries all waveform display color settings.	5-121
:SYSTem:LCD:COLor:GRAPh:CHANnel<x>	Sets or queries the display color of each waveform.	5-121
:SYSTem:LCD:COLor:INTENsity:GRID	Sets or queries the grid intensity.	5-121
:SYSTem:MLCD?	Queries all settings for the monochrome LCD on the WT1500 instrument.	5-122
:SYSTem:MLCD:CONTRast	Sets or queries the monochrome LCD contrast.	5-122
:SYSTem:MLCD:DMODE	Sets the screen content of the monochrome LCD.	5-122
:SYSTem:MLCD:MODE	Sets or queries the monochrome LCD backlight on/off status.	5-122
:SYSTem:MODEl?	Queries the model code (model name).	5-122
:SYSTem:MSGLog:SAVE[:EXECute]	Saves the message log.	5-122
:SYSTem:MUSYnc?	Queries all multi-unit synchronization status settings.	5-122
:SYSTem:MUSYnc:ABORt	Aborts multi-unit synchronous operation.	5-122
:SYSTem:MUSYnc:CNUMber?	Queries the number of connected units for multi-unit synchronization (Communication Number).	5-122
:SYSTem:MUSYnc:CStAtE:SUBunit<x>?	Queries the connection status (Communication State) between the main unit and a connected subunit in multi-unit synchronization.	5-123
:SYSTem:MUSYnc:EXECute	Executes multi-unit synchronous operation.	5-123
:SYSTem:MUSYnc:MODE	Sets or queries the multi-unit synchronization mode.	5-123
:SYSTem:MUSYnc:SStAtE?	Queries the synchronized operation status (Synchronous State) for multi-unit synchronization.	5-123
:SYSTem:MUSYnc:SUNit	Sets or queries the unit used to configure measurement conditions for multi-unit synchronization and the unit to which queries are sent (Selected Unit).	5-123
:SYSTem:RZERo	Sets or queries the on/off status of rounding to zero (Rounding to Zero) for numeric data.	5-123
:SYSTem:SERial?	Queries the serial number.	5-124
:SYSTem:SUFFix?	Queries the suffix (suffix code).	5-124
:SYSTem:TIME	Sets or queries the time.	5-124
:SYSTem:TLOCK	Sets or queries the touch lock on/off status.	5-124
:SYSTem:TSYNchro:IEEE1588?	Queries all IEEE 1588 time synchronization settings.	5-124

Command	Explanation	Page
:SYSTem:TSYNchro:IEEE1588:DMECha nism	Sets or queries the IEEE 1588 time synchronization delay mechanism (time difference calculation method).	5-124
:SYSTem:TSYNchro:IEEE1588:DNUMbe r	Sets or queries the IEEE 1588 time synchronization domain number.	5-124
:SYSTem:TSYNchro:IEEE1588:NLAYe r	Sets or queries the IEEE 1588 time synchronization network layer.	5-125
:SYSTem:TSYNchro:IEEE1588:PSTa te?	Queries the IEEE 1588 time synchronization reception status.	5-125
:SYSTem:TSYNchro:IEEE1588[:STA Te]	Sets or queries IEEE 1588 time synchronization on/off.	5-125
:SYSTem:TZONE	Sets or queries the time zone.	5-125
:SYSTem:USBKeyboard	Sets or queries the USB keyboard type.	5-125

UDP Group

:UDP:IDeNtifier	Sets or queries the identifier of the header output through UDP.	5-126
:UDP:IPAdDress	Sets or queries the destination IP address for UDP output.	5-126
:UDP:MODE	Sets or queries whether UDP output starts automatically when the power is turned on.	5-126
:UDP:PORT	Sets or queries the destination port number for UDP output.	5-126
:UDP:START	Starts UDP output.	5-126
:UDP:STOP	Stops UDP output.	5-126

UPDate Group

:UPDate?	Queries all data update settings.	5-127
:UPDate:AUTO?	Queries all auto update mode settings.	5-127
:UPDate:AUTO:RATE	Sets or queries the data update interval in auto update mode.	5-127
:UPDate:AUTO:TIMEout	Sets or queries the timeout value in auto update mode.	5-127
:UPDate[:MODE]	Sets or queries the update mode.	5-127
:UPDate:RATE	Sets or queries the data update interval.	5-127
:UPDate:TRIGger?	Queries all trigger update mode settings.	5-127
:UPDate:TRIGger:LEVel	Sets or queries the trigger level.	5-127
:UPDate:TRIGger:MODE	Sets or queries the trigger mode.	5-127
:UPDate:TRIGger:SLOPe	Sets or queries the trigger slope.	5-127
:UPDate:TRIGger:SOURce	Sets or queries the trigger source.	5-128
:UPDate:TRIGger:UNIT	Sets or queries the trigger unit (Trigger Unit).	5-128

WAVeform Group

:WAVeform?	Queries all waveform display data output settings.	5-129
:WAVeform:BYTeorder	Sets or queries the output byte order of waveform display data (FLOAT format) sent with :WAVeform:SEND?.	5-129
:WAVeform:END	Sets or queries the output end point of waveform display data sent with :WAVeform:SEND?.	5-129
:WAVeform:FORMat	Sets or queries the format of waveform display data sent with :WAVeform:SEND?.	5-129
:WAVeform:HOLD	Sets or queries whether to hold (ON) or release (OFF) all waveform display data.	5-129
:WAVeform:LENGth?	Queries the total number of data points in the waveform specified by :WAVeform:TRACe.	5-129

5.1 Command List

Command	Explanation	Page
:WAVeform:SEND?	Queries the waveform display data specified by :WAVeform:TRACe.	5-130
:WAVeform:SRATe?	Queries the sample rate of the acquired waveform.	5-130
:WAVeform:START	Sets or queries the output start point of waveform display data sent with :WAVeform:SEND?.	5-130
:WAVeform:TRACe	Sets or queries the waveform to which :WAVeform:SEND? applies.	5-130
:WAVeform:TRIGger?	Queries the trigger position of the acquired waveform.	5-130
:WAVeform:UNIT	Sets or queries the unit for the waveform to which :WAVeform:SEND? applies.	5-130

Common Command Group

*CAL?	Executes calibration (zero-level compensation, the same operation as selecting Misc > Cal [Execute]) and queries the result.	5-131
*CLS	Clears the standard event register, extended event register, and error queue.	5-131
*ESE	Sets or queries the value of the standard event enable register.	5-131
*ESR?	Queries the value of the standard event register and clears it at the same time.	5-131
*IDN?	Queries the model.	5-131
*OPC	Sets bit 0 (OPC bit) of the standard event register to 1 when the specified overlap command is completed.	5-131
*OPC?	Returns ASCII code 1 if the specified overlap command has been completed.	5-132
*OPT?	Queries the installed options.	5-132
*RST	Initializes the settings.	5-132
*SRE	Sets or queries the value of the service request enable register.	5-132
*STB?	Queries the value of the status byte register.	5-132
*TRG	Executes single measurement (the same operation as selecting Misc > Single [Execute]).	5-132
*TST?	Executes a self-test and queries the result.	5-133
*WAI	Waits to execute instructions that follow *WAI until the specified overlap command is completed.	5-133

5.2 AUX Group

The AUX group is for auxiliary input functions.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Input (Advanced/Options)** > **Motor/Aux** and query the settings.

- The commands in this group are valid when motor evaluation function 1 for encoders (option /EM1), motor evaluation function 1 for resolvers (option /RM1), or the motor evaluation function for encoders/resolvers (option /EM1RM1) is installed.
- AUX5 to AUX8 are valid when motor evaluation function 2 for encoders (option /EM2), motor evaluation function 2 for resolvers (option /RM2), or the motor evaluation function for encoders/resolvers (option /EM1RM1) is installed.

:AUX<x>?

Queries all auxiliary input settings.

Syntax :AUX<x>?
<x> = 1 to 8 (AUX input channel)

:AUX<x>:AUTO

Sets or queries auto voltage range on/off for auxiliary signals (analog input type).

Syntax :AUX<x>:AUTO {<Boolean>}
:AUX<x>:AUTO?
<x> = 1 to 8 (AUX input channel)

Example :AUX1:AUTO ON
:AUX1:AUTO? -> :AUX1:AUTO 1

- Description
- This setting is valid when the auxiliary input type (:AUX<x>:TYPE) is ANALog (analog input).
 - For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:FILTer?

Queries all auxiliary input filter settings.

Syntax :AUX<x>:FILTer?
<x> = 1 to 8 (AUX input channel)

:AUX<x>:FILTer[:LINE]

Sets or queries the auxiliary input line filter.

Syntax :AUX<x>:FILTer[:LINE] {OFF|<Frequency>}
:AUX<x>:FILTer:LINE?
<x> = 1 to 8 (AUX input channel)
OFF = line filter off
<Frequency> = 100 Hz, 500 Hz, 1 kHz (line filter on, cutoff frequency)

Example :AUX1:FILTer:LINE OFF
:AUX1:FILTer:LINE?
-> :AUX1:FILTer:LINE OFF

- Description
- This setting is valid when the auxiliary input type (:AUX<x>:TYPE) is ANALog (analog input).
 - For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:FILTer:NOISe

Sets or queries the auxiliary input pulse noise filter.

Syntax :AUX<x>:FILTer:NOISe {OFF|<Frequency>}
:AUX<x>:FILTer:NOISe?
<x> = 1 to 8 (AUX input channel)
OFF = Noise filter off
<Frequency> = 10 kHz, 100 kHz, 1 MHz (noise filter on, cutoff frequency)

Example :AUX1:FILTer:NOISe OFF
:AUX1:FILTer:NOISe?
-> :AUX1:FILTer:NOISe OFF

- Description
- This setting is valid when the auxiliary input type (:AUX<x>:TYPE) is PULSe (pulse input).
 - For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:LSCale?

Queries all auxiliary input linear scaling settings.

Syntax :AUX<x>:LSCale?
<x> = 1 to 8 (AUX input channel)

Description Auxiliary input linear scaling is a setting common to both ANALog (analog input) and PULSe (pulse input) input types (:AUX<x>:TYPE).

:AUX<x>:LSCale:AVALue

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

Syntax :AUX<x>:LSCale:AVALue {<NRf>}
AUX<x>:LSCale:AVALue?

<x> = 1 to 8 (AUX input channel)
<NRf> = -1.000E+06 to 1.000E+06
Example :AUX1:LSCALE:AVALUE 1.000
:AUX1:LSCALE:AVALUE?
-> :AUX1:LSCALE:AVALUE 1.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.2 AUX Group

:AUX<x>:LSCale:BVALue

Sets or queries offset value B of the auxiliary input linear scale.

Syntax :AUX<x>:LSCale:BVALue {<NRf>}
:AUX<x>:LSCale:BVALue?
<x> = 1 to 8 (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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:LSCale:CALCulate?

Queries all settings for calculating the parameters of the auxiliary input linear scale.

Syntax :AUX<x>:LSCale:CALCulate?
<x> = 1 to 8 (AUX input channel)

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

Sets or queries the data {Point1X|Point1Y|Point2X|Point2Y} used to calculate the parameters of the auxiliary input linear scale.

Syntax :AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y} {<NRf>}
:AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}?
<x> = 1 to 8 (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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:LSCale:CALCulate:EXECute

Executes the calculation of the parameters of the auxiliary input linear scale.

Syntax :AUX<x>:LSCale:CALCulate:EXECute
<x> = 1 to 8 (AUX input channel)

Example :AUX1:LSCALE:CALCULATE:EXECUTE

Description • This command calculates and sets slope A and offset value B of the linear scale from the previously set data (Point1X, Point1Y, Point2X, Point2Y).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:NAME

Sets or queries the auxiliary input name.

Syntax :AUX<x>:NAME {<String>}
AUX<x>:NAME?

<x> = 1 to 8 (AUX input channel)
<String> = Up to 8 characters

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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:PRANge

Sets or queries the range for auxiliary signals (pulse input type).

Syntax :AUX<x>:PRANge {<NRf>, <NRf>}
:AUX<x>:PRANge?
<x> = 1 to 8 (AUX input channel)
<NRf> = -10.00E+06 to 10.00E+06

Example :AUX1:PRANGE 10000,0
AUX1:PRANGE?
-> :AUX1:PRANGE 10.00E+03,0.000E+00

Description • Specify the upper limit and then the lower limit.
• This setting is valid when the auxiliary input type (:AUX<x>:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:RANge

Sets or queries the voltage range for auxiliary signals (analog input type).

Syntax :AUX<x>:RANge {<Voltage>}
:AUX<x>:RANge?
<x> = 1 to 8 (AUX input channel)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :AUX1:RANGE 20V
AUX1:RANGE? -> :AUX1:RANGE 20.0E+00

Description • This setting is valid when the auxiliary input type (:AUX<x>:TYPE) is ANALog (analog input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:AUX<x>:SCALing

Sets or queries the scaling factor for auxiliary input.

Syntax : AUX<x>:SCALing {<NRf>}
 : AUX<x>:SCALing?
 <x> = 1 to 8 (AUX input channel)
 <NRf> = 0.0001 to 99999.9999

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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:AUX<x>:TYPE

Sets or queries the auxiliary input type.

Syntax : AUX<x>:TYPE {ANALog|PULSe}
 : AUX<x>:TYPE?
 <x> = 1 to 8 (AUX input channel)

Example : AUX1:TYPE ANALOG
 : AUX1:TYPE? -> :AUX1:TYPE ANALOG

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:AUX<x>:UNIT

Sets or queries the unit added to the auxiliary input.

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

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

Description

- This command does not affect the calculation result.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

5.3 CAN Group

The commands in the CAN group are valid only when the CAN/CAN FD interface (option /CN1) is installed.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output > CAN** and query the settings and measured values.

: CAN?

Queries all settings for the CAN data output function.

Syntax :CAN?

: CAN: CONDItion?

Queries the CAN bus signal port status.

Syntax :CAN:CONDItion?

Example :CAN:CONDITION? -> STOP

Description The response contents are as follows.

STOP = CAN stopped.

START = CAN transmitting.

: CAN: FD: SETUp: BRATe

Sets or queries the bit rate of the arbitration phase for CAN bus signals (CAN FD).

Syntax :CAN:FD:SETUp:BRATe {<Nrf>}
:CAN:FD:SETUp:BRATe?

<Nrf> = 500000, 1000000

Example :CAN:FD:SETUP:BRATE 500000

:CAN:FD:SETUP::BRATE?

-> :CAN:FD:SETUP:BRATE 500000

: CAN: FD: SETUp: DBRate

Sets or queries the bit rate of the data phase for CAN bus signals (CAN FD).

Syntax :CAN:FD:SETUp:DBRate {<Nrf>}
:CAN:FD:SETUp:DBRate?

<Nrf> = 500000, 1000000, 2000000, 4000000

Example :CAN:FD:SETUP:DBRATE 125000

:CAN:FD:SETUP:DBRATE?

-> :CAN:FD:SETUP:DBRATE 125000

: CAN: FD: SETUp: DSPoint

Sets or queries the sample point of the data phase for CAN bus signals (CAN FD).

Syntax :CAN:FD:SETUp:DSPoint {<Nrf>}
:CAN:FD:SETUp:DSPoint?

<Nrf> = 50.0, 58.3, 66.6, 75.0, 83.3, 91.6

Example :CAN:FD:SETUP:DSPOINT 50.0

:CAN:FD:SETUP:DSPOINT?

-> :CAN:FD:SETUP:DSPOINT 50.0

: CAN: FD: SETUp: SPOInt

Sets or queries the sample point of the arbitration phase for CAN bus signals (CAN FD).

Syntax :CAN:FD:SETUp:SPOInt {<Nrf>}
:CAN:FD:SETUp:SPOInt?

<Nrf> = 50.0, 58.3, 66.6, 75.0, 83.3, 91.6

Example :CAN:FD:SETUP:SPOINT 50.0

:CAN:FD:SETUP:SPOINT?

-> :CAN:FD:SETUP:SPOINT 50.0

: CAN: FD: SETUp: TERMinator

Sets or queries the port terminator for CAN bus signals (CAN FD).

Syntax :CAN:FD:SETUp:TERMinator {<Boolean>}
:CAN:FD:SETUp:TERMinator?

Example :CAN:FD:SETUP:TERMINATOR 1

:CAN:FD:SETUP:TERMINATOR?

-> :CAN:FD:TERMINATOR 1

: CAN: FILE?

Queries all settings for saving DBC files.

Syntax :CAN:FILE?

: CAN: FILE: ANAMing

Sets or queries the function for automatically generating DBC file names when saving.

Syntax :CAN:FILE:ANAMing {OFF|NUMBERING|
DATE}

:CAN:FILE:ANAMing?

Example :CAN:FILE:ANAMING NUMBERING

:CAN:FILE:ANAMING?

-> :CAN:FILE:ANAMING NUMBERING

: CAN: FILE: CDIRectory

Changes the save destination directory for DBC files.

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

Example :CAN:FILE:CDIRECTORY "DBC"

Description To move to the parent directory, specify "..".

:CAN:FILE:DRIVE

Sets the save destination drive for DBC files.

Syntax :CAN:FILE:Drive {USER|USB[, <Nrf>] | NETWORK}

USER = internal memory drive
 USB = USB memory drive,
 <Nrf> = 0, 1, or 2 (drive number)
 NETWORK = network drive

Example :CAN:FILE:DRIVE USER

:CAN:FILE:FREE?

Queries the free space (byte) on the save destination drive for stored numeric data.

Syntax :CAN:FILE:FREE?

Example :CAN:FILE:FREE? -> 20912128

:CAN:FILE:NAME

Sets or queries the save file name for DBC files.

Syntax :CAN:FILE:NAME {<String>}
 :CAN:FILE:NAME?

<String> = File name

Example :CAN:FILE:NAME "DBC1"
 :CAN:FILE:NAME?
 -> :CAN:FILE:NAME "DBC1"

:CAN:FILE:PATH?

Queries the DBC file save destination as an absolute path.

Syntax :CAN:FILE:PATH?

Example :CAN:FILE:PATH? -> "USB-0/DBC"

:CAN:FILE:SAVE:DBC:ABORT

Aborts saving the DBC file.

Syntax :CAN:FILE:SAVE:DBC:ABORT

Example :CAN:FILE:SAVE:DBC:ABORT

:CAN:FILE:SAVE:DBC:EXECUTE

Saves the DBC file.

Syntax :CAN:FILE:SAVE:DBC:EXECUTE

Example :CAN:FILE:SAVE:DBC:EXECUTE "DBC1"

Description Specify the file name without the extension (.dbc).
 This command is an overlap command.

:CAN:MFORMAT

Sets or queries the message format for CAN bus signals.

Syntax :CAN:MFORMAT {STANDARD|EXTENDED}
 :CAN:MFORMAT?

STANDARD = standard format
 EXTENDED = extended format

Example :CAN:MFORMAT STANDARD
 :CAN:MFORMAT?
 -> :CAN:MFORMAT STANDARD

:CAN:NUMERIC:ESELECT?

Queries all settings for transmit items of CAN bus signals for numeric data (element selection method).

Syntax :CAN:NUMERIC:ESELECT?

Description This command is valid when the stored item selection method (:CAN:NUMERIC:ITEM) is element selection method (ESELECT).

:CAN:NUMERIC:ESELECT:ALL

Sets the output elements for all functions collectively when numeric data is transmitted as CAN bus signals.

Syntax :CAN:NUMERIC:ESELECT:ALL {<Boolean>}

Example :CAN:NUMERIC:ESELECT:ALL ON

:CAN:NUMERIC:ESELECT:<Function>

Sets or queries the output elements for each function when numeric data is transmitted as CAN bus signals.

Syntax :CAN:NUMERIC:ESELECT:
 <Function> {ALL|<Element>[, <Element>]
 [, <Element>] [, <Element>] ... [, <Element>]}
 :CAN:NUMERIC:ESELECT:<Function>?
 <Function> = {URMS|IRMS|P|S|Q|...}
 <Element> = {<Nrf>|SIGMA|SIGMB}
 (<Nrf> = 1 to 4)

ALL = All elements and Σ

Example :CAN:NUMERIC:ESELECT:URMS 1,2,3,SIGMA
 :CAN:NUMERIC:ESELECT:URMS?
 -> :CAN:NUMERIC:ESELECT:
 URMS 1,2,3,SIGMA

Description • For the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPLAY group.
 • List all elements and Σ to turn output on for in the parameter. To turn output on for all elements and Σ , specify ALL as the parameter.

:CAN:NUMERIC:ESELECT:PRESET<x>

Presets the output elements for functions to a predefined pattern when numeric data is transmitted as CAN bus signals.

Syntax :CAN:NUMERIC:ESELECT:PRESET<x>
 <x> = 1 or 2 (preset pattern number)

Example :CAN:NUMERIC:ESELECT:PRESET1

Description For the stored item setting patterns that are applied when a preset is executed, see the User's Manual (IM WT1501-01EN).

5.3 CAN Group

:CAN:NUMERIC[:NORMAL]:UNIT

Sets or queries the target unit when numeric data is transmitted as CAN bus signals.

Syntax :CAN:NUMERIC:NORMAL:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}

:CAN:NUMERIC:NORMAL:UNIT?

Example :CAN:NUMERIC:NORMAL:UNIT MAIN

:CAN:NUMERIC:NORMAL:UNIT?

-> :CAN:NUMERIC:NORMAL:UNIT MAIN

Description • This command is valid only when the multi-unit synchronization interface (option /SY1) is installed.

- This is a communication command for the main unit.
- If the multi-unit synchronization mode (:SYSTEM:MUSYNC:MODE) is not MAIN, an error is returned.

:CAN:NUMERIC:MID

Sets or queries the message ID for CAN bus signals.

Syntax :CAN:NUMERIC:MID {<String>}
<String> = Hexadecimal string from 0 to 7e0

Example :CAN:NUMERIC:MID "7e0"

:CAN:NUMERIC:MID?

-> :CAN:NUMERIC:MID "7e0"

Description Default value: 7e0

:CAN:OUT:COUNT

Sets or queries the number of times CAN bus signal data frames are transmitted.

Syntax :CAN:OUT:COUNT {<NRF>|INFINITE}
:CAN:OUT:COUNT?

<NRF> = 1 to 9999999

INFINITE = No limit

Example :CAN:OUT:COUNT 100

:CAN:OUT:COUNT? -> :CAN:OUT:COUNT 100

:CAN:PROTOCOL

Sets or queries the protocol of the CAN bus signal port.

Syntax :CAN:PROTOCOL {CAN|CANFD_ISO|CANFD_NONISO}

:CAN:PROTOCOL?

Example :CAN:PROTOCOL CAN

:CAN:PROTOCOL -> :CAN:PROTOCOL CAN

:CAN:SETUP:BRATE

Sets or queries the bit rate for CAN bus signals.

Syntax :CAN:SETUP:BRATE {<NRF>}
:CAN:SETUP:BRATE?
<NRF> = 125000, 250000, 500000, 1000000

Example :CAN:SETUP:BRATE 125000

:CAN:SETUP:BRATE?

-> :CAN:SETUP:BRATE 125000

:CAN:SETUP:SPOINT

Sets or queries the sample point for CAN bus signals.

Syntax :CAN:SETUP:SPOINT {<NRF>}
:CAN:SETUP:SPOINT?
<NRF> = 50.0, 58.3, 66.6, 75.0, 83.3, 91.6

Example :CAN:SETUP:SPOINT 50.0

:CAN:SETUP:SPOINT?

-> :CAN:SETUP:SPOINT 50.0

:CAN:SETUP:TERMINATOR

Sets or queries the port terminator for CAN bus signals.

Syntax :CAN:SETUP:TERMINATOR {<Boolean>}
:CAN:SETUP:TERMINATOR?

Example :CAN:SETUP:TERMINATOR 1

:CAN:SETUP:TERMINATOR?

-> :CAN:TERMINATOR 1

:CAN:START

Starts CAN bus signal output for numeric data.

Syntax :CAN:START

Example :CAN:START

:CAN:STOP

Stops CAN bus signal output for numeric data.

Syntax :CAN:STOP

Example :CAN:STOP

5.4 COMMunicate Group

The COMMunicate group is for communication.

:COMMunicate?

Queries all communication settings.

Syntax :COMMunicate?

:COMMunicate:HEADer

Sets or queries whether responses to queries are returned with a header (example: :DISPLAY:CONFIG NUMERIC) or without a header (example: 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 commands that are subject to *OPC, *OPC?, and *WAI.

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 example above, all bits are set to 1 so that all overlap commands are included. However, bits fixed at 0 are not set to 1, so only bit 6 is 1 in the query response.

:COMMunicate:OPSR?

Queries the value of 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 device access not complete

:COMMunicate:OVERlap

Sets or queries which commands 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 example above, all bits are set to 1 so that all overlap commands are included. However, bits fixed at 0 are not set to 1, so only bit 6 is 1 in the query response.
• For how to synchronize using COMMunicate:OVERlap, see page 4-12.

:COMMunicate:REMOte

Sets remote/local. The instrument is in remote mode when this is ON.

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

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

5.4 COMMunicate Group

:COMMunicate:VERBose

Sets or queries whether responses to queries are returned in full form (example: :INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03) or abbreviated form (example: :VOLT:RANG:ELEM1 1.000E+03).

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

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

:COMMunicate:WAIT

Waits until one of the specified extended events occurs.

Syntax :COMMunicate:WAIT <Register>
 <Register> = 0 to 65535 (extended event register;
 see section 6.4)

Example :COMMUNICATE:WAIT 1

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

:COMMunicate:WAIT?

Creates a response when one of the specified extended events occurs.

Syntax :COMMunicate:WAIT? <Register>
 <Register> = 0 to 65535 (extended event register;
 see section 6.4)

Example :COMMUNICATE:WAIT? 65535 -> 1

5.5 CURSor Group

The CURSor group is for cursor measurement.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output > Display > Cursors** and query the settings and measured values.

:CURSor?

Queries all cursor measurement settings.

Syntax :CURSor?

:CURSor:BAR<x>?

Queries all settings for cursor measurement in the bar graph display.

Syntax :CURSor:BAR<x>?
<x> = 1 to 4 (bar graph display group)

:CURSor:BAR<x>:LINKage

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

Syntax :CURSor:BAR<x>:LINKage {<Boolean>}
:CURSor:BAR<x>:LINKage?
<x> = 1 to 4 (bar graph display group)

Example :CURSOR:BAR1:LINKAGE OFF
:CURSOR:BAR1:LINKAGE?
-> :CURSOR:BAR1:LINKAGE 0

:CURSor:BAR<x>:POSition<x>

Sets or queries the cursor position in the bar graph display.

Syntax :CURSor:BAR<x>:POSition<x> {<Nrf>}
:CURSor:BAR<x>:POSition<x>?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of POSition<x> = 1 or 2
(1 = C1+, 2 = C2x)
<Nrf> = 0 to 100

Example :CURSOR:BAR1:POSITION1 1
:CURSOR:BAR1:POSITION1?
-> :CURSOR:BAR1:POSITION1 1

:CURSor:BAR<x>[:STATe]

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

Syntax :CURSor:BAR<x>[:STATe] {<Boolean>}
:CURSor:BAR<x>:STATe?
<x> = 1 to 4 (bar graph display group)

Example :CURSOR:BAR1:STATE ON
:CURSOR:BAR1:STATE?
-> :CURSOR:BAR1:STATE 1

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

Queries the cursor measurement values in the bar graph display.

Syntax :CURSor:BAR<x>:{Y<x>|DY}?
<x> of BAR<x> = 1 to 4 (bar graph display group)
Y<x> = Y value of the cursor position (Y1 = Y1+, Y2+, Y3+; Y2 = Y1x, Y2x, Y3x)
DY = Y value between cursors (DY1, DY2, DY3)

Example :CURSOR:BAR1:Y1? -> 78.628E+00

Description • When multiple bar graphs are displayed, this command returns the Bar Graph Cursor Measurement values of each bar graph in order.
• If cursor display is not turned on for the bar graph, this command returns NAN(Not A Number).

:CURSor:WAVE<x>?

Queries all settings for cursor measurement in the waveform display.

Syntax :CURSor:WAVE<x>?
<x> = 1 to 4 (waveform display group)

:CURSor:WAVE<x>:LINKage

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

Syntax :CURSor:WAVE<x>:LINKage {<Boolean>}
:CURSor:WAVE<x>:LINKage?
<x> = 1 to 4 (waveform display group)

Example :CURSOR:WAVE1:LINKAGE OFF
:CURSOR:WAVE1:LINKAGE?
-> :CURSOR:WAVE1:LINKAGE 0

:CURSor:WAVE<x>:PATH

Sets or queries the cursor path in the waveform display.

Syntax :CURSor:WAVE<x>:PATH {MAX|MIN|MID}
:CURSor:WAVE<x>:PATH?
<x> = 1 to 4 (waveform display group)

Example :CURSOR:WAVE1:PATH MAX
:CURSOR:WAVE1:PATH?
-> :CURSOR:WAVE1:PATH MAX

5.5 CURSor Group

:CURSor:WAVE<x>:POSition<x>

Sets or queries the cursor position in the waveform display.

Syntax :CURSor:WAVE<x>:POSition<x> {<NRf>}
:CURSor:WAVE<x>:POSition<x>?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of POSition<x> = 1 or 2
(1 = C1+, 2 = C2x)
<NRf> = 0 to 800

Example :CURSOR:WAVE1:POSITION1 200
:CURSOR:WAVE1:POSITION1?
-> :CURSOR:WAVE1:POSITION1 200

:CURSor:WAVE<x>[:STATE]

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

Syntax :CURSor:WAVE<x>[:STATE] {<Boolean>}
:CURSor:WAVE<x>:STATE?
<x> = 1 to 4 (waveform display group)

Example :CURSOR:WAVE1:STATE ON
:CURSOR:WAVE1:STATE?
-> :CURSOR:WAVE1:STATE 1

:CURSor:WAVE<x>:TRACe<x>

Sets or queries the trace for the cursor in the waveform display.

Syntax :CURSor:WAVE<x>:TRACe<x> {U<x>|I<x>|SPEed<x>|TORQue<x>|AUX<x>}
:CURSor:WAVE<x>:TRACe<x>?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of TRACe<x> = 1 or 2 (1 = C1+, 2 = C2x)
<x> of U<x> and I<x> = 1 to 4 (element)
<x> of SPEed<x> and TORQue<x> = 1 to 4
(motor number)
<x> of AUX<x> = 1 to 8 (AUX input channel)

Example :CURSOR:WAVE1:TRACE1 U1
:CURSOR:WAVE1:TRACE1?
-> :CURSOR:WAVE1:TRACE1 U1

Description

- {SPEed<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
- The values <x> = 3 or 4 for {SPEed<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.

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

Queries cursor measurement values in the waveform display.

Syntax :CURSor:WAVE<x>:{X<x>|DX|PERDt|Y<x>|DY}?
<x> of WAVE<x> = 1 to 4 (waveform display group)
X<x> = X value of the cursor position (X1 = X+, X2 = Xx)
DX = X value between cursors (DX)
PERDt = 1/DT value between cursors (1/DX)
Y<x> = Y value of the cursor position (Y1 = Y+, Y2 = Yx)
DY = Y value between cursors (DY)

Example :CURSOR:WAVE1:Y1? -> 78.628E+00

Description If cursor display is not turned on in the waveform display, this command returns NAN (Not A Number).

5.6 DISPlay Group

The DISPlay group is for screen display.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output** > **Display** and query the settings.

:DISPlay?

Queries all display settings.

Syntax :DISPlay?

Description Returns all settings related to the current display mode (:DISPlay:CONFig).

:DISPlay:BAR<x>?

Queries all bar graph display settings.

Syntax :DISPlay:BAR<x>?

<x> = 1 to 4 (bar graph display group)

:DISPlay:BAR<x>:FORMAt

Sets or queries the bar graph display format (number of screen divisions).

Syntax :DISPlay:BAR<x>:FORMAt {SINGLe|DUAL|TRIad}
:DISPlay:BAR<x>:FORMAt?

<x> = 1 to 4 (bar graph display group)

Example :DISPLAY:BAR1:FORMAT SINGLE
:DISPLAY:BAR1:FORMAT?
-> :DISPLAY:BAR1:FORMAT SINGLE

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

Queries all settings for each bar graph.

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

<x> of BAR<x> = 1 to 4 (bar graph display group)

<x> of ITEM<x> = 1 to 3 (item number)

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

Sets or queries the bar graph display item (function and element).

Syntax :DISPlay:BAR<x>:ITEM<x>[:
FUNCTION] {<Function>,<Element>}
:DISPlay:BAR<x>:ITEM<x>:FUNCTION?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)
<Function> = {U|I|P|S|Q|LAMBda|PHI|PHIU|PHII|
Z|RS|XS|RP|XP}
<Element> = 1 to 4

Example :DISPLAY:BAR1:ITEM1 U,1
:DISPLAY:BAR1:ITEM1?
-> :DISPLAY:BAR1:ITEM1 U,1

Description For details on the <Function> options, see "Numeric list data functions" at the end of this group, on page 5-49.

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

Queries all bar graph scaling settings.

Syntax :DISPlay:BAR<x>:ITEM<x>:SCALIng?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)

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

Sets or queries the bar graph scaling mode.

Syntax :DISPlay:BAR<x>:ITEM<x>:SCALIng:
MODE {FIXEd|MANual}
:DISPlay:BAR<x>:ITEM<x>:SCALIng:MODE?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)

Example :DISPLAY:BAR1:ITEM1:SCALING:
MODE FIXED
:DISPLAY:BAR1:ITEM1:SCALING:MODE?
-> :DISPLAY:BAR1:ITEM1:SCALING:
MODE FIXED

5.6 DISPlay Group

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

Sets or queries the bar graph manual scaling upper limit.

Syntax :DISPlay:BAR<x>:ITEM<x>:SCALing:VAlue {<NRf>}
:DISPlay:BAR<x>:ITEM<x>:SCALing:VAlue?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)
<NRf> = 0 to 9.999E+12

Example :DISPLAY:BAR1:ITEM1:SCALING:VALUE 100
:DISPLAY:BAR1:ITEM1:SCALING:VALUE?
-> :DISPLAY:BAR1:ITEM1:SCALING:VALUE 100.0E+00

Description • This setting is valid when the bar graph scaling mode (:DISPlay:BAR<x>:ITEM<x>:SCALing:MODE) is set to MANual.
• Only the upper limit is set. The lower limit is determined automatically as follows according to the vertical scale type (:DISPlay:BAR<x>:ITEM<x>:SCALing:VERTical).
(1) When set to LInear
The lower limit is "0" when the X-axis position (:DISPlay:BAR<x>:ITEM<x>:SCALing:XAXis) is set to BOTTom, and "– upper limit" when it is set to CENTer.
(2) When set to LOG
The lower limit is "upper limit/10000."

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

Sets or queries the bar graph vertical scale type.

Syntax :DISPlay:BAR<x>:ITEM<x>:SCALing:VERTical {LInear|LOG}
:DISPlay:BAR<x>:ITEM<x>:SCALing:VERTical?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)

Example :DISPLAY:BAR1:ITEM1:SCALING:VERTICAL LOG
:DISPLAY:BAR1:ITEM1:SCALING:VERTICAL?
-> :DISPLAY:BAR1:ITEM1:SCALING:VERTICAL LOG

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

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

Sets or queries the bar graph X-axis position.

Syntax :DISPlay:BAR<x>:ITEM<x>:SCALing:XAXis {BOTTom|CENTer}
:DISPlay:BAR<x>:ITEM<x>:SCALing:XAXis?
<x> of BAR<x> = 1 to 4 (bar graph display group)
<x> of ITEM<x> = 1 to 3 (item number)

Example :DISPLAY:BAR1:ITEM1:SCALING:XAXIS BOTTOM
:DISPLAY:BAR1:ITEM1:SCALING:XAXIS?
-> :DISPLAY:BAR1:ITEM1:SCALING:XAXIS BOTTOM

Description This setting is valid when the bar graph scaling mode (:DISPlay:BAR<x>:ITEM<x>:SCALing:MODE) is set to MANual and the vertical scale type (:DISPlay:BAR<x>:ITEM<x>:SCALing:VERTical) is set to LInear.

:DISPlay:BAR<x>:ORDER

Sets or queries the starting and ending harmonic orders displayed in the bar graph.

Syntax :DISPlay:BAR<x>:ORDER {<NRf>,<NRf>}
:DISPlay:BAR<x>:ORDER?
<x> = 1 to 4 (bar graph display group)
First <NRf> = 0 to 90 (starting harmonic order)
Second <NRf> = 10 to 100 (ending harmonic order)

Example :DISPLAY:BAR1:ORDER 1,100
:DISPLAY:BAR1:ORDER?
-> :DISPLAY:BAR1:ORDER 1,100

Description • Specify the starting harmonic order and then the ending harmonic order.
• Set the ending harmonic order to a value greater than or equal to the starting harmonic order + 10.

:DISPlay:CONFig

Sets or queries the display mode.

Syntax :DISPlay:CONFig { (NUMeric|WAVE<x>|
BAR<x>|VECTor<x>) [, (NUMeric|WAVE<x>|
BAR<x>|VECTor<x>)] }
:DISPlay:CONFig?
NUMeric = numeric display
WAVE<x> = waveform display
BAR<x> = bar graph
VECTor<x> = vector
<x> = 1 to 4 (display group)

Example :DISPLAY:CONFIG NUMERIC,WAVE
:DISPLAY:CONFIG?
-> :DISPLAY:CONFIG NUMERIC,WAVE

Description • For full screen display, specify one parameter.
• For split screen display, specify two parameters in the order of the top screen and bottom screen.

:DISPlay:NUMeric<x>?

Queries all numeric display settings.

Syntax :DISPlay:NUMeric<x>?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen

:DISPlay:NUMeric<x>:NORMal?

Queries all numeric display settings.

Syntax :DISPlay:NUMeric<x>:NORMal?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen

Description Returns all settings for the current numeric display format (:DISPlay:NUMeric<x>[:NORMal]:FORMat).

:DISPlay:NUMeric<x>[:NORMal]:ALL?

Queries all numeric display (All items display) settings.

Syntax :DISPlay:NUMeric<x>[:NORMal]:ALL?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen

:DISPlay:NUMeric<x>[:NORMal]:ALL:ORDER

Sets or queries the harmonic order displayed on the harmonic measurement function display page of numeric display (All items display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:ALL:
ORDER {<Order>}
:DISPlay:NUMeric<x>[:NORMal]:ALL:
ORDER?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen
<Order> = {TOTal|DC|<NRF>}
(<NRF> = 1 to 100)

Example :DISPLAY:NUMERIC1:NORMAL:ALL:ORDER 1
:DISPLAY:NUMERIC1:NORMAL:ALL:ORDER?
-> :DISPLAY:NUMERIC1:NORMAL:ALL:
ORDER 1

Description This setting is valid when the display page number for numeric display (All items display) (:DISPlay:NUMeric<x>[:NORMal]:ALL:PAGE) is 9 or 10.

:DISPlay:NUMeric<x>[:NORMal]:ALL:PAGE

Sets or queries the display page for numeric display (All items display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:ALL:
PAGE {<NRF>}
:DISPlay:NUMeric<x>[:NORMal]:ALL:PAGE?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen
<NRF> = 1 to 11 (page number)

Example :DISPLAY:NUMERIC1:NORMAL:ALL:PAGE 1
:DISPLAY:NUMERIC1:NORMAL:ALL:PAGE?
-> :DISPLAY:NUMERIC1:NORMAL:ALL:
PAGE 1

5.6 DISPlay Group

:DISPlay:NUMeric<x>[:NORMal]:FORMat

Sets or queries the numeric display format.

Syntax :DISPlay:NUMeric<x>[:NORMal]:
 FORMat {VAL4|VAL8|VAL16|MATRix|ALL|
 SINGle|DUAL}
 :DISPlay:NUMeric<x>[:NORMal]:FORMat?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen

Example :DISPLAY:NUMERIC1:NORMAL:FORMAT VAL4
 :DISPLAY:NUMERIC1:NORMAL:FORMAT?
 -> :DISPLAY:NUMERIC1:NORMAL:
 FORMAT VAL4

Description The displayed numeric data is as follows.
 {VAL4|VAL8|VAL16} = Displays numeric display
 items in item-number order. (The number
 indicates the number of items displayed on one
 screen (page).)
 MATRix = Displays selected functions arranged
 by element. (matrix display)
 ALL = Displays all functions arranged by element.
 (All items display)
 SINGle = Displays one list display item in a list,
 separated into EVEN and ODD. (Hrm List
 Single (harmonic list single display))
 DUAL = Displays two list display items in a list,
 arranged in harmonic order. (Hrm List Dual
 (harmonic list dual display))

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

Queries all numeric display (harmonic list display) settings.

Syntax :DISPlay:NUMeric<x>[:NORMal]:LIST?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen

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

Sets or queries the cursor position in numeric display (harmonic list display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:LIST:
 CURSor {HEADer|LIST}
 :DISPlay:NUMeric<x>[:NORMal]:LIST:
 CURSor?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen

HEADer = Scrolls pages in the header area (data related to all harmonics, left side of the screen).

LIST = Scrolls pages in the data area (numeric data for each harmonic, right side of the screen).

Example :DISPLAY:NUMERIC1:NORMAL:LIST:
 CURSOR LIST
 :DISPLAY:NUMERIC1:NORMAL:LIST:CURSOR?
 -> :DISPLAY:NUMERIC1:NORMAL:LIST:
 CURSOR LIST

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

Sets or queries the cursor position in the header area of numeric display (harmonic list display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:LIST:
 HEADer {<NRf>}
 :DISPlay:NUMeric<x>[:NORMal]:LIST:
 HEADer?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 to 110 (header row)

Example :DISPLAY:NUMERIC1:NORMAL:LIST:
 HEADER 1
 :DISPLAY:NUMERIC1:NORMAL:LIST:HEADER?
 -> :DISPLAY:NUMERIC1:NORMAL:LIST:
 HEADER 1

Description This setting is valid when the cursor position in numeric display (harmonic list display) (:DISPlay:NUMeric<x>[:NORMal]:LIST:CURSor) is set to HEADer.

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

Sets or queries the display item (function and element) for numeric display (harmonic list display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:LIST:ITEM<x> {<Function>,<Element>}
:DISPlay:NUMeric<x>[:NORMal]:LIST:ITEM<x>?
<x> of NUMeric<x> = 1 or 2
1: full screen or top screen
2: bottom screen
<x> of ITEM<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}
(<NRf> = 1 to 4)

Example :DISPLAY:NUMERIC1:NORMAL:LIST:ITEM1 U,1
:DISPLAY:NUMERIC1:NORMAL:LIST:ITEM1?
-> :DISPLAY:NUMERIC1:NORMAL:LIST:ITEM1 U,1

Description • The display items are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting or querying.
• For details on the <Function> options, see “Numeric list data functions” at the end of this group, on page 5-49.

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

Sets or queries the harmonic order cursor position in the data area of numeric display (harmonic list display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:LIST:ORDER {<NRf>}
:DISPlay:NUMeric<x>[:NORMal]:LIST:ORDER?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen
<NRf> = 1 to 100 (harmonic order)

Example :DISPLAY:NUMERIC1:NORMAL:LIST:ORDER 1
:DISPLAY:NUMERIC1:NORMAL:LIST:ORDER?
-> :DISPLAY:NUMERIC1:NORMAL:LIST:ORDER 1

Description This setting is valid when the cursor position in numeric display (harmonic list display) (:DISPlay:NUMeric<x>[:NORMal]:LIST:CURSOr) is set to LIST.

:DISPlay:NUMeric<x>[:NORMal]:MATRIX?

Queries all numeric display (matrix display) settings.

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRIX?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen

:DISPlay:NUMeric<x>[:NORMal]:MATRIX:COLumn?

Queries all settings for display columns in numeric display (matrix display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRIX:COLumn?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen

Description Column display settings are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting or querying.

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

Sets or queries the column display item for numeric display (matrix display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRIX:COLumn:ITEM<x> {NONE|<Element>}
:DISPlay:NUMeric<x>[:NORMal]:MATRIX:COLumn:ITEM<x>?
<x> of NUMeric<x> = 1 or 2
1: full screen or top screen
2: bottom screen
<x> of ITEM<x> = 1 to 6 (display item (column) number)
<Element> = {<NRf>|SIGMa|SIGMB}
(<NRf> = 1 to 4)

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:COLumn:ITEM1 1
:DISPLAY:NUMERIC1:NORMAL:MATRIX:COLumn:ITEM1?
-> :DISPLAY:NUMERIC1:NORMAL:MATRIX:COLumn:ITEM1 1

5.6 DISPlay Group

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

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

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRix:
 COLumn:NUMBER {<NRf>}
 :DISPlay:NUMeric<x>[:NORMal]:MATRix:
 COLumn:NUMBER?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 4, 6

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 COLUMN:NUMBER 4
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 COLUMN:NUMBER?
 -> :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 COLUMN:NUMBER 4

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

Resets the column display items for numeric display (matrix display) to their default values.

Syntax :DISPlay:NUMeric[:NORMal]:MATRix:
 COLumn:RESet
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 COLUMN:RESET

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

Sets or queries the cursor position in numeric display (matrix display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:
 MATRix:CURSor {<NRf>}
 :DISPlay:NUMeric<x>[:NORMal]:MATRix:
 CURSor?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 to 81 (item number)

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 CURSOR 1
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 CURSOR?
 -> :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 CURSOR 1

Description Specify the cursor position by item number.

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

Sets or queries the display item (function and harmonic order) for numeric display (matrix display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRix:
 ITEM<x> {NONE|<Function>[,<Element>]
 [,<Order>]}
 :DISPlay:NUMeric<x>[:NORMal]:MATRix:
 ITEM<x>?
 <x> of NUMeric<x> = 1 or 2
 1: full screen or top screen
 2: bottom screen
 <x> of ITEM<x> = 1 to 81 (item number)
 NONE = no display item
 <Function> = {URMS|IRMS|P|S|Q|...}
 <Element> = {<NRf>|SIGMa|SIGMB}
 (<NRf> = 1 to 4)
 <Order> = {TOTal|DC|<NRf>}
 (<NRf> = 1 to 100)

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1 URMS
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1?
 -> :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1 URMS,1
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1 UK,1,1
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1?
 -> :DISPLAY:NUMERIC1:NORMAL:MATRIX:
 ITEM1 UK,1,1

- Description**
- The display items are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting or querying.
 - For details on the <Function> options, see "Numeric data functions" at the end of this group, on page 5-45.
 - The <Element> setting does not affect the display. If omitted, element 1 is set.
 - If <Order> is omitted, TOTal is set.
 - In responses for functions that do not require <Element> or <Order>, <Element> or <Order> is omitted.

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

Sets or queries the display page for numeric display (matrix display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRix:PAGE {<NRf>}
 :DISPlay:NUMeric<x>[:NORMal]:MATRix:PAGE?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 to 9 (page number)

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:PAGE 1
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:PAGE?
 -> :DISPLAY:NUMERIC1:NORMAL:MATRIX:PAGE 1

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

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

Presets the display items for numeric display (matrix display) to a specified pattern.

Syntax :DISPlay:NUMeric<x>[:NORMal]:MATRix:PRESet {<NRf>|EORigin|FORigin|CLRPage|CLRAll}
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 or EORigin = element-based reset pattern (Element Origin)
 <NRf> = 2 or FORigin = function-based reset pattern (Function Origin)
 <NRf> = 3 or CLRPage = clears the display items on the current page (Clear Current Page)
 <NRf> = 4 or CLRAll = clears the display items on all pages (Clear All Pages)

Example :DISPLAY:NUMERIC1:NORMAL:MATRIX:PRESET 1
 :DISPLAY:NUMERIC1:NORMAL:MATRIX:PRESET EORIGIN

Description

- The display items are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting.
- The numeric display item display pattern (order) is the same as the display order used when you select Items [display item switching] > Reset Items [display item reset] > Execute [execute reset] from the Display menu. For details on the display order when a reset is executed, see the User's Manual (IM WT1501-01EN).

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

Queries all numeric display ({4 items|8 items|16 items} display) settings.

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen

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

Sets or queries the cursor position in numeric display ({4 items|8 items|16 items} display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:CURSor {<NRf>}
 :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:CURSor?
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 to 48 (item number, when VAL4 is selected)
 <NRf> = 1 to 96 (item number, when VAL8 is selected)
 <NRf> = 1 to 192 (item number, when VAL16 is selected)

Example :DISPLAY:NUMERIC1:NORMAL:VAL4:CURSOR 1
 DISPLAY:NUMERIC1:NORMAL:VAL4:CURSOR?
 -> :DISPLAY:NUMERIC1:NORMAL:VAL4:CURSOR 1

Description Specify the cursor position by item number.

5.6 DISPlay Group

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

Sets or queries the display item (function, element, and harmonic order) for numeric display ({4 items|8 items|16 items} display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>:FUNction {NONE|<Function>[,<Element>][,<Order>]}

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

 <x> of NUMeric<x> = 1 or 2

 1: full screen or top screen

 2: bottom screen

 <x> of ITEM<x> = 1 to 48 (item number, when VAL4 is selected)

 <x> of ITEM<x> = 1 to 96 (item number, when VAL8 is selected)

 <x> of ITEM<x> = 1 to 192 (item number, when VAL16 is selected)

 NONE = no display item

 <Function> = {URMS|IRMS|P|S|Q|...}

 <Element> = {<NRf>|SIGMa|SIGMB}

 (<NRf> = 1 to 4)

 <Order> = {TOTa|DC|<NRf>}

 (<NRf> = 1 to 100)

Example :DISPLAY:NUMERIC1:NORMAL:VAL4:

 ITEM1:FUNCTION URMS,1

 :DISPLAY:NUMERIC1:NORMAL:VAL4:ITEM1:

 FUNCTION?

 -> :DISPLAY:NUMERIC1:NORMAL:VAL4:

 ITEM1:FUNCTION URMS,1

 :DISPLAY:NUMERIC1:NORMAL:VAL4:

 ITEM1:FUNCTION UK,1,1

 :DISPLAY:NUMERIC1:NORMAL:VAL4:ITEM1:

 FUNCTION?

 -> :DISPLAY:NUMERIC1:NORMAL:VAL4:

 ITEM1:FUNCTION UK,1,1

Description • The display items are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting or querying.

- For details on the <Function> options, see “Numeric data functions” at the end of this group, on page 5-45.
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTa is set.
- In responses for functions that do not require <Element> or <Order>, <Element> or <Order> is omitted.
- For motor functions (<Function> = {SPEed|TORQue|SYNCsp|SLIP|PM}), the <Element> part is <Motor>. <Order> is not required.

 <Motor> = 1 to 4 (motor number; motor 1 if omitted)

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

Sets or queries the unit of the display item (function, element, and harmonic order) for numeric display ({4 items|8 items|16 items} display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}

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

 <x> of NUMeric<x> = 1 or 2

 1: full screen or top screen

 2: bottom screen

 <x> of ITEM<x> = 1 to 48 (item number, when VAL4 is selected)

 <x> of ITEM<x> = 1 to 96 (item number, when VAL8 is selected)

 <x> of ITEM<x> = 1 to 192 (item number, when VAL16 is selected)

Example :DISPLAY:NUMERIC1:NORMAL:VAL4:ITEM1:

 UNIT MAIN

 :DISPLAY:NUMERIC1:NORMAL:VAL4:ITEM1:

 UNIT?

 -> :DISPLAY:NUMERIC1:NORMAL:VAL4:ITEM1:

 :UNIT MAIN

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

Sets or queries the display page for numeric display ({4 items|8 items|16 items} display).

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:PAGE {<NRf>}

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

 <x> = 1 to 2

 1: full screen or top screen

 2: bottom screen

 <NRf> = 1 to 12 (page number)

Example :DISPLAY:NUMERIC1:NORMAL:VAL4:PAGE 1

 :DISPLAY:NUMERIC1:NORMAL:VAL4:PAGE?

 -> :DISPLAY:NUMERIC1:NORMAL:VAL4:

 PAGE 1

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

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

Presets the display items for numeric display ({4 items|8 items|16 items} display) to a specified pattern.

Syntax :DISPlay:NUMeric<x>[:NORMal]:{VAL4|VAL8|VAL16}:PRESet {<NRf>|EORigin|FORigin|CLRPage|CLRAll}
 <x> = 1 to 2
 1: full screen or top screen
 2: bottom screen
 <NRf> = 1 or EORigin = element-based reset pattern (Element Origin)
 <NRf> = 2 or FORigin = function-based reset pattern (Function Origin)
 <NRf> = 3 or CLRPage = clears the display items on the current page (Clear Current Page)
 <NRf> = 4 or CLRAll = clears the display items on all pages (Clear All Pages)

Example :DISPLAY:NUMERIC1:NORMAL:VAL4:
 PRESET 1
 :DISPLAY:NUMERIC1:NORMAL:VAL4:
 PRESET EORIGIN

Description

- The display items are common to full screen display and split screen display. The <x> of NUMeric<x> is irrelevant to setting.
- The numeric display item display pattern (order) is the same as the display order used when you select Items [display item switching] > Reset Items [display item reset] > Execute [execute reset] from the Display menu. For details on the display order when a reset is executed, see the User's Manual (IM WT1501-01EN).

:DISPlay:VECTor<x>?

Queries all vector display settings.

Syntax :DISPlay:VECTor<x>?
 <x> = 1 to 4 (vector display group)

:DISPlay:VECTor<x>:FORMat

Sets or queries the vector display format (number of screen divisions).

Syntax :DISPlay:VECTor<x>:FORMat {SINGLE|DUAL}
 :DISPlay:VECTor<x>:FORMat?
 <x> = 1 to 4 (vector display group)

Example :DISPLAY:VECTOR1:FORMAT SINGLE
 :DISPLAY:VECTOR1:FORMAT?
 -> :DISPLAY:VECTOR1:FORMAT SINGLE

:DISPlay:VECTor<x>:ITEM<x>?

Queries all settings for each vector.

Syntax :DISPlay:VECTor<x>:ITEM<x>?
 <x> of VECTor<x> = 1 to 4 (vector display group)
 <x> of ITEM<x> = 1 to 2 (item number)

:DISPlay:VECTor<x>:ITEM<x>:OBJect

Sets or queries the wiring unit to be displayed as the vector object.

Syntax :DISPlay:VECTor<x>:ITEM<x>:OBJect {<Element>}
 :DISPlay:VECTor<x>:ITEM<x>:OBJect?
 <x> of VECTor<x> = 1 to 4 (vector display group)
 <x> of ITEM<x> = 1 to 2 (item number)
 <Element> = {<NRf>|SIGMa|SIGMB}
 (<NRf> = 1 to 4)

Example :DISPLAY:VECTOR1:ITEM1:OBJECT SIGMA
 :DISPLAY:VECTOR1:ITEM1:OBJECT?
 -> :DISPLAY:VECTOR1:ITEM1:
 OBJECT SIGMA

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

Sets or queries the vector display {voltage|current} zoom factor.

Syntax :DISPlay:VECTor<x>:ITEM<x>:{UMAG|IMAG} {<NRf>}
 :DISPlay:VECTor<x>:ITEM<x>:{UMAG|IMAG}?
 <x> of VECTor<x> = 1 to 4 (vector display group)
 <x> of ITEM<x> = 1 to 2 (item number)
 <NRf> = 0.100 to 100.000

Example :DISPLAY:VECTOR1:ITEM1:UMAG 1
 :DISPLAY:VECTOR1:ITEM1:UMAG?
 -> :DISPLAY:VECTOR1:ITEM1:UMAG 1.000

5.6 DISPlay Group

:DISPlay:VECTor<x>:NUMeric

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

Syntax :DISPlay:VECTor<x>:
NUMeric {<Boolean>}
:DISPlay:VECTor<x>:NUMeric?
<x> = 1 to 4 (vector display group)
Example :DISPLAY:VECTOR1:NUMERIC ON
:DISPLAY:VECTOR1:NUMERIC?
-> :DISPLAY:VECTOR1:NUMERIC 1

:DISPlay:VUNit<x>

Sets or queries the view unit (View Unit).

Syntax :DISPlay:VUNit<x> {MAIN|SUB1|SUB2|
SUB3|SUB4}
:DISPlay:VUNit<x>?
<x> = 1 to 2
1: full screen or top screen
2: bottom screen
Example :DISPLAY:VUNIT1 MAIN
:DISPLAY:VUNIT1?
-> :DISPLAY:VUNIT1 MAIN
Description This setting is valid only when the multi-unit
synchronization interface option (/SY1) is
installed.

:DISPlay:WAVE<x>?

Queries all waveform display settings.

Syntax :DISPlay:WAVE<x>?
<x> = 1 to 4 (waveform display group)

:DISPlay:WAVE<x>:ALL

Sets on/off for all waveform displays at once.

Syntax :DISPlay:WAVE<x>:ALL {<Boolean>}
<x> = 1 to 4 (waveform display group)
Example :DISPLAY:WAVE1:ALL ON

:DISPlay:WAVE<x>:FORMat

Sets or queries the waveform display format (number of screen divisions).

Syntax :DISPlay:WAVE<x>:FORMat {SINGLE|DUAL|
TRIad|QUAD|HEXa}
:DISPlay:WAVE<x>:FORMat?
<x> = 1 to 4 (waveform display group)
Example :DISPLAY:WAVE1:FORMAT SINGLE
:DISPLAY:WAVE1:FORMAT?
-> :DISPLAY:WAVE1:FORMAT SINGLE

:DISPlay:WAVE<x>:GRATicule

Sets or queries the graticule type.

Syntax :DISPlay:WAVE<x>:GRATicule {GRID|
FRAME|CROSShair}
:DISPlay:WAVE<x>:GRATicule?
<x> = 1 to 4 (waveform display group)
Example :DISPLAY:WAVE1:GRATICULE GRID
:DISPLAY:WAVE1:GRATICULE?
-> :DISPLAY:WAVE1:GRATICULE GRID
Description This setting is common to waveform display
groups. The <x> of WAVE<x> is irrelevant to
setting or querying.

:DISPlay:WAVE<x>:INTERpolate

Sets or queries the waveform interpolation method.

Syntax :DISPlay:WAVE<x>:INTERpolate {OFF|
LINE}
:DISPlay:WAVE<x>:INTERpolate?
<x> = 1 to 4 (waveform display group)
Example :DISPLAY:WAVE1:INTERPOLATE LINE
:DISPLAY:WAVE1:INTERPOLATE?
-> :DISPLAY:WAVE1:INTERPOLATE LINE
Description This setting is common to waveform display
groups. The <x> of WAVE<x> is irrelevant to
setting or querying.

:DISPlay:WAVE<x>:MAPPING?

Queries all settings for assigning waveforms to split formats (split screens).

Syntax :DISPlay:WAVE<x>:MAPPING?
<x> = 1 to 4 (waveform display group)

:DISPlay:WAVE<x>:MAPPING[:MODE]

Sets or queries the method for assigning waveforms to split formats (split screens).

Syntax :DISPlay:WAVE<x>:MAPPING[:
MODE] {AUTO|FIXed|USER}
:DISPlay:WAVE<x>:MAPPING:MODE?
<x> = 1 to 4 (waveform display group)
Example :DISPLAY:WAVE1:MAPPING:MODE AUTO
:DISPLAY:WAVE1:MAPPING:MODE?
-> :DISPLAY:WAVE1:MAPPING:MODE AUTO

:DISPlay:WAVE<x>:MAPPING:USER: {U<x>| I<x>| SPEEd<x>| TORQue<x>| AUX<x>}

Sets or queries the assignment of {voltage|current|rotation speed|torque|auxiliary signal} waveforms to split formats (split screens).

Syntax :DISPlay:WAVE<x>:MAPPING:USER: {U<x>| I<x>| SPEEd<x>| TORQue<x>| AUX<x>} {<Nrf>}
:DISPlay:WAVE<x>:MAPPING:USER: {U<x>| I<x>| SPEEd<x>| TORQue<x>| AUX<x>}?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of U<x> and I<x> = 1 to 4 (element)
<x> of SPEEd<x> and TORQue<x> = 1 to 4 (MOTOR input channel)
<x> of AUX<x> = 1 to 8 (AUX input channel)
<Nrf> = 1 to 6

Example :DISPLAY:WAVE1:MAPPING:USER:U1 1
:DISPLAY:WAVE1:MAPPING:USER:U1?
-> :DISPLAY:WAVE:MAPPING:USER:U1 1

Description

- This setting is valid when the waveform assignment method (:DISPlay:WAVE<x>:MAP Ping[:MODE]) is set to USER.
- {SPEEd<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
- The values <x> = 3 or 4 for {SPEEd<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.

:DISPlay:WAVE<x>:POSition?

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

Syntax :DISPlay:WAVE<x>:POSition?
<x> = 1 to 4 (waveform display group)

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

Sets or queries the vertical position (center-position level) of the {voltage|current} waveform for each element.

Syntax :DISPlay:WAVE<x>:POSition: {U<x>| I<x>} {<Nrf>}
:DISPlay:WAVE<x>:POSition: {U<x>| I<x>}?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of U<x> and I<x> = 1 to 4 (element)
<Nrf> = -130.000 to 130.000 (%)

Example :DISPLAY:WAVE1:POSITION:U1 0
:DISPLAY:WAVE1:POSITION:U1?
-> :DISPLAY:WAVE1:POSITION:U1 0.000

:DISPlay:WAVE<x>:POSition: {UALL| IALL}

Sets the vertical position (center-position level) of the {voltage|current} waveforms for all elements at once.

Syntax :DISPlay:WAVE<x>:POSition: {UALL| IALL} {<Nrf>}
<x> = 1 to 4 (waveform display group)
<Nrf> = -130.000 to 130.000 (%)

Example :DISPLAY:WAVE1:POSITION:UALL 0

:DISPlay:WAVE<x>:SVALue

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

Syntax :DISPlay:WAVE<x>:SVALue {<Boolean>}
:DISPlay:WAVE<x>:SVALue?
<x> = 1 to 4 (waveform display group)

Example :DISPLAY:WAVE1:SVALUE ON
:DISPLAY:WAVE1:SVALUE?
-> :DISPLAY:WAVE1:SVALUE 1

Description This setting is common to waveform display groups. The <x> of WAVE<x> is irrelevant to setting or querying.

:DISPlay:WAVE<x>:TDIV

Sets or queries the waveform Time/div value.

Syntax :DISPlay:WAVE<x>:TDIV {<Time>}
:DISPlay:WAVE<x>:TDIV?
<x> = 1 to 4 (waveform display group)
<Time> = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500 (ms), 1, 2 (s)

Example :DISPLAY:WAVE1:TDIV 5MS
:DISPLAY:WAVE1:TDIV?
-> :DISPLAY:WAVE1:TDIV 5.00E-03

Description This setting is common to waveform display groups. The <x> of WAVE<x> is irrelevant to setting or querying.

:DISPlay:WAVE<x>:TLABel

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

Syntax :DISPlay:WAVE<x>:TLABel {<Boolean>}
:DISPlay:WAVE<x>:TLABel?
<x> = 1 to 4 (waveform display group)

Example :DISPLAY:WAVE1:TLABEL OFF
:DISPLAY:WAVE1:TLABEL?
-> :DISPLAY:WAVE1:TLABEL 0

Description This setting is common to waveform display groups. The <x> of WAVE<x> is irrelevant to setting or querying.

5.6 DISPlay Group

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

Sets or queries the on/off status of {voltage|current|rotation speed|torque|auxiliary signal} waveform display.

Syntax :DISPlay:WAVE<x>: {U<x> | I<x> | SPEEd<x> | TORQue<x> | AUX<x>} {<Boolean>}
:DISPlay:WAVE<x>: {U<x> | I<x> | SPEEd<x> | TORQue<x> | AUX<x>} ?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of U<x> and I<x> = 1 to 4 (element)
<x> of SPEEd<x> and TORQue<x> = 1 to 4 (MOTOR number)
<x> of AUX<x> = 1 to 8 (AUX input channel)

Example :DISPLAY:WAVE1:U1 ON
:DISPLAY:WAVE1:U1?
-> :DISPLAY:WAVE1:U1 1

Description • {SPEEd<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
• The values <x> = 3 or 4 for {SPEEd<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.

:DISPlay:WAVE<x>:VZoom?

Queries all waveform vertical zoom factor settings.

Syntax :DISPlay:WAVE<x>:VZoom?
<x> = 1 to 4 (waveform display group)

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

Sets or queries the vertical zoom factor of the {voltage|current} waveform for each element.

Syntax :DISPlay:WAVE<x>:VZoom: {U<x> | I<x>} {<NRf>}
:DISPlay:WAVE<x>:VZoom: {U<x> | I<x>} ?
<x> of WAVE<x> = 1 to 4 (waveform display group)
<x> of U<x> and I<x> = 1 to 4 (element)
<NRf> = 0.1 to 100

Example :DISPLAY:WAVE1:VZOOM:U1 1
:DISPLAY:WAVE1:VZOOM:U1?
-> :DISPLAY:WAVE1:VZOOM:U1 1.00

Description For details on the zoom factor options, see the User's Manual (IM WT1501-01EN).

:DISPlay:WAVE<x>:VZoom: {UALL | IALL}

Sets the vertical zoom factor of the {voltage|current} waveforms for all elements at once.

Syntax :DISPlay:WAVE<x>:VZoom: {UALL | IALL} {<NRf>}
<x> = 1 to 4 (waveform display group)
<NRf> = 0.1 to 100

Example :DISPLAY:WAVE1:VZOOM:UALL 1

Description For details on the zoom factor options, see the User's Manual (IM WT1501-01EN).

Function options (<Function>) list

- Numeric data functions
- Numeric list data functions

Numeric data functions

Applicable commands

```
:CAN:NUMERIC:ESELECT:<Function> {ALL|<Element>[,Element>][,Element>]
[,Element>]...[,Element>]}
:DISPlay:NUMERIC<x>[:NORMAL]:{VAL4|VAL8|VAL16}:ITEM<x>[:FUNCTION
on] {<Function>[,<Element>][,<Order>]}
:DISPlay:NUMERIC<x>[:NORMAL]:MATRIX:ITEM<x> {<Function>[,<Element>]
[,<Order>]}
:FILE:SAVE:NUMERIC:ESELECT:<Function> {ALL|<Element>[,Element>]
[,Element>][,Element>]...[,Element>]}
:FILE:SAVE:NUMERIC:NORMAL:<Function> {<Boolean>}
:MEASURE:EVENT<x>:EXPRESSION:ITEM[:FUNCTION] {<Function>[,<Element>]
[,<Order>]}
:NUMERIC[:NORMAL]:ITEM<x>[:FUNCTION] {<Function>[,<Element>]
[,<Order>]}
:STORE:NUMERIC:ESELECT:<Function> {ALL|<Element>[,Element>]
[,Element>][,Element>]...[,Element>]}
:STORE:NUMERIC[:NORMAL]:<Function> {<Boolean>}
```

Yes: Required. No: Not required.

<Function>	Function notation on the menu (numeric display header name)	<Element> (element)	<Order> (harmonic order)
URMS	Urms	Yes	No
UMN	Umn	Yes	No
UDC	Udc	Yes	No
URMN	Urmn	Yes	No
UAC	Uac	Yes	No
IRMS	Irms	Yes	No
IMN	Imn	Yes	No
IDC	Idc	Yes	No
IRMN	Irmn	Yes	No
IAC	Iac	Yes	No
P	P	Yes	No
S	S	Yes	No
Q	Q	Yes	No
LAMBda	λ	Yes	No
PHI	ϕ	Yes	No
FU	FreqU(fU)	Yes	No
FI	FreqI(fI)	Yes	No
UPPeak	U+peak(U+pk)	Yes	No
UMPeak	U-peak(U-pk)	Yes	No
IPPeak	I+peak(I+pk)	Yes	No
IMPeak	I-peak(I-pk)	Yes	No
CFU	CfU	Yes	No
CFI	CfI	Yes	No
PC	Pc	Yes	No

5.6 DISPlay Group

Yes: Required. No: Not required.

<Function>	Function notation on the menu (numeric display header name)	<Element> (element)	<Order> (harmonic order)
PPPeak	P+peak(P+pk)	Yes	No
PMPeak	P-peak(P-pk)	Yes	No
UFND	Ufnd	Yes	No
IFND	Ifnd	Yes	No
PFND	Pfnd	Yes	No
SFND	Sfnd	Yes	No
QFND	Qfnd	Yes	No
LAMBDAFND	λ fnd	Yes	No
PHIFND	ϕ fnd	Yes	No
URIP	Urip	Yes	No
IRIP	Irip	Yes	No
UUBF	Uubf	Yes (Σ only)	No
IUBF	Iubf	Yes (Σ only)	No
ITIME	ITime	Yes	No
WH	WP	Yes	No
WHP	WP+	Yes	No
WHM	WP-	Yes	No
AH	q	Yes	No
AHP	q+	Yes	No
AHM	q-	Yes	No
WS	WS	Yes	No
WQ	WQ	Yes	No
ETA1 to ETA4	η 1 to η 4	No	No
F1 to F20	F1 to F20	No	No
EV1 to EV8	Event1 to Event8	No	No
Harmonic measurement function			
UK	U(k)	Yes	Yes
IK	I(k)	Yes	Yes
PK	P(k)	Yes	Yes
SK	S(k)	Yes	Yes
QK	Q(k)	Yes	Yes
LAMBDK	λ (k)	Yes	Yes
PHIK	ϕ (k)	Yes	Yes
PHIUk	ϕ U(k)	Yes	Yes
PHIik	ϕ I(k)	Yes	Yes
Zk	Z(k)	Yes	Yes
RSk	Rs(k)	Yes	Yes
XSk	Xs(k)	Yes	Yes
RPk	Rp(k)	Yes	Yes
XPk	Xp(k)	Yes	Yes
UHDFk	Uhdf(k)	Yes	Yes
IHDFk	Ihdf(k)	Yes	Yes
PHDFk	Phdf(k)	Yes	Yes
UTHD	Uthd	Yes	No
ITHD	Ithd	Yes	No
PTHD	Pthd	Yes	No
UTHF	Uthf	Yes	No
ITHF	Ithf	Yes	No
UTIF	Utif	Yes	No
ITIF	Itif	Yes	No
HVF	hvf	Yes	No
HCF	hcf	Yes	No

Yes: Required. No: Not required.

<Function>	Function notation on the menu (numeric display header name)	<Element> (element)	<Order> (harmonic order)
KFACtor	K-factor	Yes	No
PHI_U1U2	ϕ_{Ui-Uj}	Yes (Σ only)	No
PHI_U1U3	ϕ_{Ui-Uk}	Yes (Σ only)	No
PHI_U1I1	ϕ_{Ui-Ii}	Yes	No
PHI_U2I2	ϕ_{Uj-Ij}	Yes (Σ only)	No
PHI_U3I3	ϕ_{Uk-Ik}	Yes (Σ only)	No
FPLL1	FreqPLL1 (fPLL1)	No	No
FPLL2	FreqPLL2 (fPLL2)	No	No
Functions that require motor evaluation function 1 (option /EM1, /RM1, or /EM1RM1)			
SPEed	Speed	Yes (motor number)	No
TORQue	Torque	Yes (motor number)	No
SYNCsp	SyncSp	Yes (motor number)	No
SLIP	Slip	Yes (motor number)	No
PM	Pm	Yes (motor number)	No
EAU	EaU	Yes	No
EAI	EaI	Yes	No
AUX1	Aux1	No	No
AUX2	Aux2	No	No
AUX3	Aux3	No	No
AUX4	Aux4	No	No
Functions that require motor evaluation function 2 (option /EM2, /RM2, or /EM1RM1)			
EAU	EaU	Yes	No
EAI	EaI	Yes	No
AUX5	Aux5	No	No
AUX6	Aux6	No	No
AUX7	Aux7	No	No
AUX8	Aux8	No	No

Note

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

5.6 DISPlay Group

Applicable commands

```
:FILE:SAVE:NUMERIC:NORMAL:<Function> {<Boolean>}
:NUMERIC[:NORMAL]:ITEM<x>[:FUNCTION] {<Function>[,<Element>]
[,<Order>]}
:STORE:NUMERIC[:NORMAL]:<Function> {<Boolean>}
```

Yes: Required. No: Not required.

<Function>	Function notation on the menu (numeric display header name)	<Element> (element)	<Order> (harmonic order)
Measurement range			
RNGU	RngU	Yes	No
RNGI	RngI	Yes	No
RNGSPD	RngSpd	Yes (motor number)	No
RNGTRQ	RngTrq	Yes (motor number)	No
RNGAUX	RngAux	Yes (Aux input channel)	No
Data update status (available when the update mode is Auto)			
UNPower	UpdateStsPwr	No	No
UNMotor	UpdateStsMtr	No	No
UNAux	UpdateStsAux	No	No
UHPower	UpdateStsHrm	No	No

For details on data update status, see appendix 10 of the User's Manual (IM WT1501-01EN).

Applicable commands

```
:NUMERIC[:NORMAL]:ITEM<x>[:FUNCTION] {<Function>[,<Element>]
[,<Order>]}
```

Yes: Required. No: Not required.

<Function>	Function notation on the menu (numeric display header name)	<Element> (element)	<Order> (harmonic order)
Timestamp			
TSDate	TS Date	No	No
TSTime	TS Time	No	No
TSSubsec	TS Subsec	No	No
The following are available when the update mode is Auto.			
NPDate	TS Date NrmPwr	Yes	No
NPTIME	TS Time NrmPwr	Yes	No
NPSubsec	TS Subsec NrmPwr	Yes	No
NMDate	TS Date NrmMtr	Yes (motor number)	No
NMTIME	TS Time NrmMtr	Yes (motor number)	No
NMSubsec	TS Subsec NrmMtr	Yes (motor number)	No
NADate	TS Date NrmAux	Yes (Aux input channel)	No
NATIME	TS Time NrmAux	Yes (Aux input channel)	No
NASubsec	TS Subsec NrmAux	Yes (Aux input channel)	No
HPDate	TS Date HrmPwr	Yes	No
HPTIME	TS Time HrmPwr	Yes	No
HPSubsec	TS Subsec HrmPwr	Yes	No

Numeric list data functions

Applicable commands

```
:DISPlay:BAR<x>:ITEM<x>[:FUNCTION] {<Function>,<Element>}
:DISPlay:NUMeric<x>[:NORMAL]:LIST:ITEM<x> {<Function>,<Element>}
:NUMeric:LIST:ITEM<x>[:FUNCTION] {NONE|<Function>,<Element>}
```

<Function>	Function notation 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 following are available only with :NUMeric:LIST:ITEM<x>.	
UHDF	Uhdf
IHDF	Ihdf
PHDF	Phdf

5.7 FILE Group

The FILE group is for file operations.

You can configure, execute, and query the settings of the same functions as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output** > **Data Save** > **File List**, and the **File List**, **Save Setup**, and **Load Setup** screens that appear from the setup menu.

: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 To move to the parent directory, specify "..".

:FILE:COUNT?

Queries the number of subdirectories and files in the target directory.

Syntax :FILE:COUNT?

Example :FILE:COUNT? -> 3

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

Deletes a 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 the extension.

:FILE:DELeTe:NUMeric:ASCii

Deletes a numeric data file.

Syntax :FILE:DELeTe:NUMeric:ASCii {<String>}
<String> = File name

Example :FILE:DELETE:NUMERIC:ASCII "NUM1"

Description Specify the file name without the extension.

:FILE:DELeTe:SETup

Deletes a setup parameter file.

Syntax :FILE:DELeTe:SETup {<String>}
<String> = File name

Example :FILE:DELETE:SETUP "SETUP1"

Description Specify the file name without the extension.

:FILE:DELeTe:STORE:{DATA|HEADer}

Deletes a 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 the extension.

:FILE:DELeTe:WAVE:ASCii

Deletes a 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 the extension.

:FILE:DRIVE

Sets the current drive.

Syntax :FILE:DRIVE {USER|USB[,<NRF>]|NETWork}
USER = internal memory drive
USB = USB memory drive, <NRF> = 0, 1, 2 (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

Description When ITEM is selected, the list is filtered according to the item set with :FILE:ITEM.

:FILE:FREE?

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

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

:FILE:ITEM

Sets the type of files to include in the file list.

Syntax :FILE:ITEM {SETup|WAVE|NUMeric}

Example :FILE:ITEM SETUP

Description Specify this before retrieving file list information (:FILE:COUNT? or :FILE:LIST?).

:FILE:LIST?

Queries information about subdirectories and files in the current directory (name, size, date and time, and attributes).

Syntax :FILE:LIST? {<NRf>}

<NRf> = list number (0 or later)

Example :FILE:LIST? 1

```
-> "SETUP1.SET,98560,2018/01/01,
    12:34:56,R/W"
```

Description Returns the file name, size (bytes), date and time, and attributes as a string.

:FILE:LOAD:ABORT

Aborts file loading.

Syntax :FILE:LOAD:ABORT

Example :FILE:LOAD:ABORT

:FILE:LOAD:SETup

Loads a setup parameter file.

Syntax :FILE:LOAD:SETup {<String>}

<String> = File name

Example :FILE:LOAD:SETUP "SETUP1"

Description

- Specify the file name without the 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 file saving.

Syntax :FILE:SAVE:ABORT

Example :FILE:SAVE:ABORT

:FILE:SAVE:ANAMing

Sets or queries the auto naming function 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 added to saved files.

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:ESElect?

Queries all settings for numeric data file saving (element selection method).

Syntax :FILE:SAVE:NUMeric:ESElect?

Description This setting is valid when the save item selection method (:FILE:SAVE:NUMeric:ITEM) is set to element selection (ESElect).

:FILE:SAVE:NUMeric:ESElect:ALL

Sets all output elements for all functions at once when saving numeric data files.

Syntax :FILE:SAVE:NUMeric:ESElect:

ALL {<Boolean>}

Example :FILE:SAVE:NUMERIC:ESELECT:ALL ON

:FILE:SAVE:NUMeric:ESElect:<Function>

Sets or queries the output elements for each function when saving numeric data files.

Syntax :FILE:SAVE:NUMeric:ESElect:
<Function> {ALL|<Element>[,Element>]
[,Element>][,Element>]...[,Element>]]
:FILE:SAVE:NUMeric:ESElect:
<Function>?

<Function> = {URMS|IRMS|P|S|Q|...}

<Element> = {<NRf>|SIGMa|SIGMB}

(<NRf> = 1 to 4)

ALL = All elements and Σ

Example :FILE:SAVE:NUMERIC:ESELECT:

URMS 1,2,3,SIGMA

:FILE:SAVE:NUMERIC:ESELECT:URMS?

```
-> :FILE:SAVE:NUMERIC:ESELECT:
```

URMS 1,2,3,SIGMA

Description

- For the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.

- List all elements and Σ to turn output on for in the parameter. To turn output on for all elements and Σ , specify ALL as the parameter.

5.7 FILE Group

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

Presets the output elements of functions to a specified pattern when saving numeric data files.

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

Example :FILE:SAVE:NUMERIC:ESELECT:PRESET1

Description For details on the output setting pattern used when a preset is executed, see the User's Manual (IM WT1501-01EN).

: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 the extension.
• This command is an overlap command.

:FILE:SAVE:NUMeric:ITEM

Sets or queries the selection method for items to save when saving numeric data to a file.

Syntax :FILE:SAVE:NUMeric:ITEM {DISPlayed|SELEcted|ESElect}
:FILE:SAVE:NUMeric:ITEM?
DISPlayed = automatic selection method for screen display items
SELEcted = manual selection method
ESElect = manual selection method (element selection method)

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

Description The options are as follows.
DISPlayed = saves the numeric items displayed on the screen to a file
SELEcted = saves the numeric items individually set with commands under " :FILE:SAVE:NUMeric:NORMal:..." to a file
ESElect = saves the numeric items individually set with commands under " :FILE:SAVE:NUMeric:ESElect:..." to a file

:FILE:SAVE:NUMeric:NORMal?

Queries all settings for saving numeric data files (manual selection method for items to save).

Syntax :FILE:SAVE:NUMeric:NORMal?

Description This setting is valid when the selection method for items to save (:FILE:SAVE:NUMeric:ITEM) is set to manual selection method (SELEcted).

:FILE:SAVE:NUMeric:NORMal:ALL

Sets the output on/off status for all elements and functions at once when saving numeric data 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}

Sets or queries the output on/off status for each element or wiring unit ({ΣA|ΣB}) when saving numeric data to a file.

Syntax :FILE:SAVE:NUMeric:NORMal:
{ELEMENT<x>|SIGMA|SIGMB} {<Boolean>}
:FILE:SAVE:NUMeric:NORMal:
{ELEMENT<x>|SIGMA|SIGMB}?
<x>=1 to 4

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

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

Sets or queries the output on/off status for each function when saving numeric data 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 the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.

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

Sets or queries the output on/off status for each motor when saving numeric data to a file.

Syntax :FILE:SAVE:NUMeric:NORMal:
MOTor<x> {<Boolean>}
:FILE:SAVE:NUMeric:NORMal:MOTor<x>?
<x> = 1 to 4

Example :FILE:SAVE:NUMERIC:NORMAL:MOTOR1 ON
:FILE:SAVE:NUMERIC:NORMAL:MOTOR1?
-> :FILE:SAVE:NUMERIC:NORMAL:MOTOR1 1

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

Presets the output on/off status of elements and functions to a specified pattern when saving numeric data to a file.

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

Example :FILE:SAVE:NUMERIC:NORMAL:PRESET1

Description For details on the output setting pattern used when a preset is executed, see the User's Manual (IM WT1501-01EN).

:FILE:SAVE:NUMeric:NORMal:UNIT

Sets or queries the unit for saving numeric data to a file.

Syntax :FILE:SAVE:NUMeric:NORMal:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}
:FILE:SAVE:NUMeric:NORMal:UNIT?

Example :FILE:SAVE:NUMERIC:NORMAL:UNIT MAIN
:FILE:SAVE:NUMERIC:NORMAL:UNIT?
-> :FILE:SAVE:NUMERIC:NORMAL:
UNIT MAIN

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

: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 the 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 the extension.
• This command is an overlap command.

5.8 HARMonics Group

The HARMonics group is for harmonic measurement.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output > Harmonics** and query the settings.

:HARMonics<x>?

Queries all harmonic measurement settings.

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

:HARMonics<x>:CONFigure?

Queries the harmonic measurement groups for all elements.

Syntax :HARMonics<x>:CONFigure?
Description The <x> of HARMonics<x> is irrelevant to querying.

:HARMonics<x>:CONFigure[:ALL]

Sets the harmonic measurement group for all elements at once.

Syntax :HARMonics<x>:CONFigure[:ALL] {<Nrf>}
<Nrf>=1 (Hrm1), 2 (Hrm2)
Example :HARMONICS1:CONFIGURE:ALL 1
Description The <x> of HARMonics<x> is irrelevant to setting or querying.

:HARMonics<x>:CONFigure:ELEment<x>

Sets or queries the harmonic measurement group for each element.

Syntax :HARMonics<x>:CONFigure:
ELEment<x> {<Nrf>}
:HARMonics<x>:CONFigure:ELEment<x>?
<x> of ELEment<x> = 1 to 4 (element)
<Nrf>=1 (Hrm1), 2 (Hrm2)
Example :HARMONICS1:CONFIGURE:ELEMENT1 1
:HARMONICS1:CONFIGURE:ELEMENT1?
-> :HARMONICS1:CONFIGURE:ELEMENT1 1
Description • The <x> of HARMonics<x> is irrelevant to setting or querying.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

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

Sets the harmonic measurement group for all elements belonging to wiring unit {ΣA|ΣB} at once.

Syntax :HARMonics<x>:CONFigure:
{SIGMA|SIGMB} {<Nrf>}
<Nrf>=1 (Hrm1), 2 (Hrm2)
Example :HARMONICS1:CONFIGURE:SIGMA 1
Description • The <x> of HARMonics<x> is irrelevant to setting.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:HARMonics<x>:CPLLsource:SOURce

Sets or queries the common PLL source for multi-unit synchronization.

Syntax :HARMonics<x>:CPLLsource:
SOURce {U<x>|I<x>|EXTernal}
:HARMonics<x>:CPLLsource:SOURce?
<x> = 1 to 4 (element)
EXTernal = external clock input (Ext Clk)
Example :HARMONICS1:CPLLSOURCE:SOURCE U1
:HARMONICS1:CPLLSOURCE:SOURCE?
-> :HARMONICS1:CPLLSOURCE:SOURCE U1
Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• In multi-unit synchronous operation, specify the unit with :SYSTem:MUSYnc:SUNit.
• The <x> of HARMonics<x> is irrelevant to setting or querying.

:HARMonics<x>:CPLLsource:UNIT

Sets or queries the common PLL unit for multi-unit synchronization.

Syntax :HARMonics<x>:CPLLsource:
UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}
:HARMonics<x>:CPLLsource:UNIT?
<x> = 1 to 4 (element)
Example :HARMONICS1:CPLLSOURCE:UNIT MAIN
:HARMONICS1:CPLLSOURCE:UNIT?
-> :HARMONICS1:CPLLSOURCE:UNIT MAIN
Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• In multi-unit synchronous operation, specify the unit with :SYSTem:MUSYnc:SUNit.
• The <x> of HARMonics<x> is irrelevant to setting or querying.

:HARMonics<x>:ORDER

Sets or queries the minimum and maximum analysis harmonic orders.

Syntax :HARMonics<x>:ORDER {<NRf>,<NRf>}
 :HARMonics<x>:ORDER?
 <x> = 1 or 2 (harmonic measurement group)
 First <NRf> = 0 or 1 (minimum measured harmonic order)
 Second <NRf> = 1 to 100 (maximum measured harmonic order)

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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:HARMonics<x>:PLLSource

Sets or queries the PLL source.

Syntax :HARMonics<x>:PLLSource {U<x>|I<x>|EXTErnal|CPLL1}
 :HARMonics<x>:PLLSource?
 <x> of HARMonics<x> = 1 or 2 (harmonic measurement group)
 <x> of U<x> and I<x> = 1 to 4 (element)
 EXTErnal = external clock input (Ext Clk)
 CPLL1 = common PLL (available only during multi-unit synchronous operation (option /SY1))

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

Description In multi-unit synchronous operation, specify the unit with :SYSTem:MUSYnc:SUNit.

:HARMonics<x>:THD

Sets or queries the formula for calculating THD (total harmonic distortion).

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

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

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

5.9 HISTory Group

The HISTory group is for the history function.
For details on the history function, see Appendix 2.

:HISTory?

Queries all history function 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 20

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

Description Manually selects the number of data updates to output together at one time during history communication output.

:HISTory:COMMunicate:FORMat

Sets or queries the format of history communication output data.

Syntax :HISTory:COMMunicate:
FORMat {AScii|FLOat}
:HISTory:COMMunicate:FORMat?

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

Description • History communication output data is currently fixed to FLOat.
• For the format of the numeric data that is output, see the description of “:NUMeric:FORMat.”

:HISTory:COMMunicate:HOLD

Sets or queries whether to hold (ON) or release (OFF) history communication output data.

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

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

Description Measurement data in the history is held from the start position corresponding to the history communication output mode at the time hold is turned on. The number of updates held depends on the specified output count.

: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 how to control the start position and output count for history communication output, and the HST bit of the Extended Event Status Register (EESR).
• N update mode (NUPDate) = Notifies “update complete” for history communication output every N measurement data updates and automatically sets the history output start position. Use this mode to continuously acquire all measurement data with a data update interval of 10 ms without omissions.

:HISTory:NUMeric:LIST:VALue?

Queries history communication numeric list data.

Syntax :HISTory:NUMeric:LIST:VALue?

Example :HISTORY:NUMERIC:LIST:VALUE?

-> #N(number of bytes in N digits)(data
byte sequence)

- Description
- This command outputs the same items as the numeric list data query for harmonic measurement (:NUMeric:LIST:VALue?) together for multiple updates.
 - The number of updates that are output is determined according to the history communication output mode.
 - The output items, number of output items, and output format for one update are the same as those for :NUMeric:LIST:VALue?.
 - This command does not allow you to specify a parameter (optional item number) like :NUMeric:LIST:VALue?.

:HISTory:NUMeric:VALue?

Queries history communication numeric data.

Syntax :HISTory:NUMeric:VALue?

Example :HISTORY:NUMERIC:VALUE?

-> #N(number of bytes in N digits)(data
byte sequence)

- Description
- This command outputs the same items as the numeric data query (:NUMeric[:NORMal]:VALue?) together for multiple updates.
 - The number of updates that are output is determined according to the history communication output mode.
 - The output items, number of output items, and output format for one update are the same as for :NUMeric[:NORMal]:VALue?. For details, see Appendix 2.
 - Unlike :NUMeric[:NORMal]:VALue?, this command does not allow you to specify a parameter (optional item number).

5.10 HOLD Group

The HOLD group is for the output data hold function.

You can configure the same setting as the HOLD on/off setting under the **Misc** menu icon  > **HOLD** and query the setting.

:HOLD

Sets or queries the hold function for output data (display data, communication data, etc.).

Syntax :HOLD {<Boolean>}
 :HOLD?

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

5.11 IMAGE Group

The IMAGE group is for saving screen image data.

You can configure and execute the same functions as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output** > **Data Save** and query changes to the settings.

:IMAGE?

Queries all settings for outputting screen image data.

Syntax :IMAGE?

:IMAGE:ABORT

Aborts output of screen image data.

Syntax :IMAGE:ABORT

Example :IMAGE:ABORT

:IMAGE:COLor

Sets or queries the color tone for saving screen image data.

Syntax :IMAGE:COLor {OFF|COLor|REVerse|GRAY}

:IMAGE:COLor?

Example :IMAGE:COLOR OFF

:IMAGE:COLOR?

-> :IMAGE:COLOR OFF

:IMAGE:EXECute

Executes output of screen image data.

Syntax :IMAGE:EXECute

Example :IMAGE:EXECUTE

:IMAGE:FORMat

Sets or queries the format for saving screen image data.

Syntax :IMAGE:FORMat {BMP|PNG|JPEG}

:IMAGE:FORMat?

Example :IMAGE:FORMAT BMP

:IMAGE:FORMAT?

-> :IMAGE:FORMAT BMP

:IMAGE:SAVE?

Queries all settings for saving screen image data.

Syntax :IMAGE:SAVE?

:IMAGE:SAVE:ANAMing

Sets or queries the auto naming function 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 destination directory for saving screen image data.

Syntax :IMAGE:SAVE:CDIRECTory {<String>}

<String> = Directory name

Example :IMAGE:SAVE:CDIRECTORY "IMAGE"

Description To move to the parent directory, specify "..".

:IMAGE:SAVE:DRIVE

Sets the destination drive for saving screen image data.

Syntax :IMAGE:SAVE:DRIVE {USER|USB[,<Nrf>]|NETWork}

USER = internal memory drive

USB = USB memory drive, <Nrf> = 0, 1, 2 (drive number))

NETWork = Network drive

Example :IMAGE:SAVE:DRIVE USER

:IMAGE:SAVE:FREE?

Queries the free space (byte) on the destination drive for saving screen image data.

Syntax :IMAGE:SAVE:FREE?

Example :IMAGE:SAVE:FREE? -> 20912128

5.11 IMAGe Group

:IMAGe:SAVE:NAME

Sets or queries the save file name.

Syntax :IMAGe:SAVE:NAME {<String>}
 :IMAGe:SAVE:NAME?
 <String> = File name

Example :IMAGe:SAVE:NAME "IMAGE1"
 :IMAGe:SAVE:NAME?
 -> :IMAGe:SAVE:NAME "IMAGE1"

Description • The destination drive for saving screen image data is set with the :IMAGe:SAVE:DRIVE command, and the directory with the :IMAGe:SAVE:CDIRECTORY command.
• You can query the destination path with :IMAGe:SAVE:PATH?.
• Specify the file name without the extension.

:IMAGe:SAVE:PATH?

Queries the absolute path of the destination for saving screen image data.

Syntax :IMAGe:SAVE:PATH?
Example :IMAGe:SAVE:PATH? -> "USB-0/IMAGE"

:IMAGe:SEND?

Queries screen image data.

Syntax :IMAGe:SEND?
Example :IMAGe:SEND?
 -> #N(N-digit byte count)(data byte sequence)

Description The number of digits N in the byte count varies depending on the size of the output data.

5.12 INPut Group

The INPut group is for input element measurement conditions.

You can configure the same settings as on the screens that appear when you select the **Setup** menu icon  > **Input (Basic)** and **Input (Advanced/Options)** and query the settings.

: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 = extended display range mode for crest
 factor 6 (6A)
Example **:INPUT:CFACtor 3**
 :INPUT:CFACtor?
 -> :INPUT:CFACtor 3

[[:INPut]:CORRection:AMPLitude?

Queries all amplitude correction settings.

Syntax **[[:INPut]:CORRection:AMPLitude?**

[[:INPut]:CORRection:AMPLitude:CURRent?

Queries all current amplitude correction settings.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent?

[[:INPut]:CORRection:AMPLitude:CURRent:RATio?

Queries the current amplitude correction ratios for all elements.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent:RATio?

[[:INPut]:CORRection:AMPLitude:CURRent:RATio[:ALL]

Sets the current amplitude correction ratio for all elements at once.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent:RATio[:ALL] {<NRf>}
 <NRf> = 0.800000 to 1.200000
Example **:INPUT:CORRECTION:AMPLITUDE:CURRENT:**
 RATIO:ALL 1

[[:INPut]:CORRection:AMPLitude:CURRent:RATio:ELEMeNt<x>

Sets or queries the current amplitude correction ratio for each element.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent:RATio:ELEMeNt<x> {<NRf>}
 [[:INPut]:CORRection:AMPLitude:
 CURRent:RATio:ELEMeNt<x>?
 <x> = 1 to 4 (element)
 <NRf> = 0.800000 to 1.200000

Example **:INPUT:CORRECTION:AMPLITUDE:CURRENT:**
 RATIO:ELEMENT1 1
 :INPUT:CORRECTION:AMPLITUDE:CURRENT:
 RATIO:ELEMENT1?
 -> :INPUT:CORRECTION:AMPLITUDE:
 CURRENT:RATIO:ELEMENT1 1.000000

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CORRection:AMPLitude:CURRent:STATe?

Queries the current amplitude correction on/off status for all elements.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent:STATe?

[[:INPut]:CORRection:AMPLitude:CURRent:STATe[:ALL]

Sets the current amplitude correction on/off status for all elements at once.

Syntax **[[:INPut]:CORRection:AMPLitude:**
 CURRent[:STATe][:ALL] {<Boolean>}
Example **:INPUT:CORRECTION:AMPLITUDE:CURRENT:**
 STATE:ALL OFF

5.12 INPut Group

[:INPut]:CORRection:AMPLitude:CURRent[:STATe]:ELEMeNt<x>

Sets or queries the current amplitude correction on/off status for each element.

Syntax [:INPut]:CORRection:AMPLitude:
CURRent[:STATe]:
ELEMeNt<x> {<Boolean>}
[:INPut]:CORRection:AMPLitude:
CURRent[:STATe]:ELEMeNt<x>?
<x> = 1 to 4 (element)

Example :INPUT:CORRECTION:AMPLITUDE:CURRENT:
STATE:ELEMENT1 OFF
:INPUT:CORRECTION:AMPLITUDE:CURRENT:
STATE:ELEMENT1?
-> :INPUT:CORRECTION:AMPLITUDE:
CURRENT:STATE:ELEMENT1 0

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut]:CORRection:DC?

Queries all DC offset correction settings.

Syntax [:INPut]:CORRection:DC?

[:INPut]:CORRection:DC:CURRent?

Queries all current DC offset correction settings.

Syntax [:INPut]:CORRection:DC:CURRent?

[:INPut]:CORRection:DC:CURRent:OFFSet?

Queries the current DC offset correction values for all elements.

Syntax [:INPut]:CORRection:DC:CURRent:OFFSet?

[:INPut]:CORRection:DC:CURRent:OFFSet[:ALL]

Sets the current DC offset correction value for all elements at once.

Syntax [:INPut]:CORRection:DC:CURRent:
OFFSet[:ALL] {<Nrf>}
<Nrf> = -9.99999E-03 to 9.99999E+03
:INPUT:CORRECTION:DC:CURRENT:OFFSET:
ALL 1

[:INPut]:CORRection:DC:CURRent:OFFSet:ELEMeNt<x>

Sets or queries the current DC offset correction value for each element.

Syntax [:INPut]:CORRection:DC:CURRent:OFFSet:EL
EMeNt<x> {<Nrf>}
[:INPut]:CORRection:DC:CURRent:OFFSet:EL
EMeNt<x>?
<x> = 1 to 4 (element)
<Nrf> = -9.99999E-03 to 9.99999E+03

Example :INPUT:CORRECTION:DC:CURRENT:OFFSET:
ELEMENT1 1
:INPUT:CORRECTION:DC:CURRENT:OFFSET:
ELEMENT1?
-> :INPUT:CORRECTION:DC:CURRENT:
OFFSET:ELEMENT1 1.000000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut]:CORRection:DC:CURRent:STATe?

Queries the current DC offset correction on/off status for all elements.

Syntax [:INPut]:CORRection:DC:CURRent:STATe?

[:INPut]:CORRection:DC:CURRent[:STATe][:ALL]

Sets the current DC offset correction on/off status for all elements at once.

Syntax [:INPut]:CORRection:DC:CURRent[:STATe]
[:ALL] {<Boolean>}
:INPUT:CORRECTION:DC:CURRENT:STATE:
ALL OFF

[:INPut]:CORRection:DC:CURRent[:STATe]:ELEMeNt<x>

Sets or queries the current DC offset correction on/off status for each element.

Syntax [:INPut]:CORRection:DC:CURRent[:STATe]:E
LEMeNt<x> {<Boolean>}
[:INPut]:CORRection:DC:CURRent[:STATe]:E
LEMeNt<x>?
<x> = 1 to 4 (element)

Example :INPUT:CORRECTION:DC:CURRENT:STATE:
ELEMENT1 OFF
:INPUT:CORRECTION:DC:CURRENT:STATE:
ELEMENT1?
-> :INPUT:CORRECTION:DC:CURRENT:STATE:E
LEMENT1 0

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CORRection:PHASe?

Queries all phase correction settings.

Syntax [:INPut]:CORRection:PHASe?

[[:INPut]:CORRection:PHASe:CURRENT?

Queries all current phase correction settings.

Syntax [:INPut]:CORRection:PHASe:CURRENT?

[[:INPut]:CORRection:PHASe:CURRENT:D PHase?

Queries the input-output phase difference for calculating the current phase correction value for all elements.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
DPHase?

[[:INPut]:CORRection:PHASe:CURRENT:D PHase[:ALL]

Sets the input-output phase difference for calculating the current phase correction value for all elements at once.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
DPHase[:ALL] {<NRf>}
<NRf> = -180.000 to 180.000

Example :INPUT:CORRECTION:PHASE:CURRENT:
DPHASE:ALL 0.1

[[:INPut]:CORRection:PHASe:CURRENT:D PHase:ELEMEnt<x>

Sets or queries the input-output phase difference for calculating the current phase correction value for each element.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
DPHase:ELEMEnt<x> {<NRf>}
[:INPut]:CORRection:PHASe:CURRENT:
DPHase:ELEMEnt<x>?
<x> = 1 to 4 (element)
<NRf> = -180.000 to 180.000

Example :INPUT:CORRECTION:PHASE:CURRENT:
DPHASE:ELEMENT1 0.1
:INPUT:CORRECTION:PHASE:CURRENT:
DPHASE:ELEMENT1?
-> :INPUT:CORRECTION:PHASE:CURRENT:
DPHASE:ELEMENT1 0.100

Description For multi-unit synchronization (option /
SY1), specify the target unit with
:SYSTem:MUSYnc:SUNit.

[[:INPut]:CORRection:PHASe:CURRENT:D TIME?

Queries the input-output time difference (phase correction value) for current.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
DTIME? {<NRf>}
[:INPut]:CORRection:PHASe:CURRENT:
DTIME?
<NRf> = 1 to 4 (element)

Example :INPUT:CORRECTION:PHASE:CURRENT:
DTIME? 1
-> 4.62960E-06
:INPUT:CORRECTION:PHASE:CURRENT:
DTIME?
-> 4.62960E-06,0.00000E+00,
0.00000E+00,0.00000E+00,0.00000E+00,0.000
00E+00,0.00000E+00

Description • The response is returned in <NR3> format (unit:
s).
• If the parameter is omitted, this command
outputs the phase correction values for all
elements in order from element 1.
• For multi-unit synchronization (option /
SY1), specify the target unit with
:SYSTem:MUSYnc:SUNit.

[[:INPut]:CORRection:PHASe:CURRENT:F REQuency?

Queries the frequency for calculating the current phase correction value for all elements.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
FREQuency?

[[:INPut]:CORRection:PHASe:CURRENT:F REQuency[:ALL]

Sets the frequency for calculating the current phase correction value for all elements at once.

Syntax [:INPut]:CORRection:PHASe:CURRENT:
FREQuency[:ALL] {<Frequency>}

Example :INPUT:CORRECTION:PHASE:CURRENT:
FREQUENCY:ALL 60

5.12 INPut Group

[[:INPut]:CORRection:PHASe:CURRent:FREQuency:ELEMeNt<x>

Sets or queries the frequency for calculating the current phase correction value for each element.

Syntax [:INPut]:CORRection:PHASe:CURRent:
FREQuency:ELEMeNt<x> {<Frequency>}
[:INPut]:CORRection:PHASe:CURRent:
FREQuency:ELEMeNt<x>?
<x> = 1 to 4 (element)
<Frequency> = 50 or 60 (Hz), 0.1 to 1000.0 (kHz)

Example :INPUT:CORRECTION:PHASE:CURRENT:
FREQUENCY:ELEMENT1 60
:INPUT:CORRECTION:PHASE:CURRENT:
FREQUENCY:ELEMENT1?
-> :INPUT:CORRECTION:PHASE:CURRENT:
FREQUENCY:ELEMENT1 60.0E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CORRection:PHASe:CURRent:STATE?

Queries the current phase correction on/off status for all elements.

Syntax [:INPut]:CORRection:PHASe:CURRent:
STATE?

[[:INPut]:CORRection:PHASe:CURRent[:STATE]][:ALL]

Sets the current phase correction on/off status for all elements at once.

Syntax [:INPut]:CORRection:PHASe:CURRent[:
STATE]][:ALL] {<Boolean>}

Example :INPUT:CORRECTION:PHASE:CURRENT:
STATE:ALL OFF

[[:INPut]:CORRection:PHASe:CURRent[:STATE]:ELEMeNt<x>

Sets or queries the current phase correction on/off status for each element.

Syntax [:INPut]:CORRection:PHASe:CURRent[:
STATE]:ELEMeNt<x> {<Boolean>}
[:INPut]:CORRection:PHASe:CURRent[:
STATE]:ELEMeNt<x>?
<x> = 1 to 4 (element)

Example :INPUT:CORRECTION:PHASE:CURRENT:
STATE:ELEMENT1 OFF
:INPUT:CORRECTION:PHASE:CURRENT:
STATE:ELEMENT1?
-> :INPUT:CORRECTION:PHASE:CURRENT:
STATE:ELEMENT1 0

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRent?

Queries all current measurement settings.

Syntax [:INPut]:CURRent?

[[:INPut]:CURRent:AUTO?

Queries the auto current range on/off status for all elements.

Syntax [:INPut]:CURRent:AUTO?

[[:INPut]:CURRent:AUTO[:ALL]

Sets the auto current range on/off status for all elements at once.

Syntax [:INPut]:CURRent:AUTO[:
ALL] {<Boolean>}

Example :INPUT:CURRENT:AUTO:ALL ON

[[:INPut]:CURRent:AUTO:ELEMeNt<x>

Sets or queries the auto current range on/off status for each element.

Syntax [:INPut]:CURRent:AUTO:
ELEMeNt<x> {<Boolean>}
[:INPut]:CURRent:AUTO:ELEMeNt<x>?
<x> = 1 to 4 (element)

Example :INPUT:CURRENT:AUTO:ELEMENT1 ON
:INPUT:CURRENT:AUTO:ELEMENT1?
-> :INPUT:CURRENT:AUTO:ELEMENT1 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRent:AUTO:{SIGMA|SIGMB}

Sets the auto current range on/off status at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax [:INPut]:CURRent:AUTO:
{SIGMA|SIGMB} {<Boolean>}

Example :INPUT:CURRENT:AUTO:SIGMA ON

Description • When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([[:INPut]:WIRing]), {SIGMA|SIGMB} is invalid.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRENT:CONFIg?

Queries the enabled current ranges for all elements.

Syntax [:INPut]:CURRENT:CONFIg?

[[:INPut]:CURRENT:CONFIg[:ALL]

Sets the enabled current ranges for all elements at once.

Syntax [:INPut]:CURRENT:CONFIg[:ALL] {ALL|
<Current>[,<Current>][,<Current>]...}
ALL = all ranges enabled

- When the crest factor setting is 3
<Current> = 25, 50, 100, 250, or 500 (mA), or 1 (A)
- When the crest factor setting is 6 or 6A
<Current> = 12.5, 25, 50, 125, 250, or 500 (mA)

Example :INPUT:CURRENT:CONFIg:ALL ALL
:INPUT:CURRENT:CONFIg:ALL 1,0.5,0.1,0.25

Description List all the current ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.

[[:INPut]:CURRENT:CONFIg:ELEMent<x>

Sets or queries the enabled current ranges for each element.

Syntax [:INPut]:CURRENT:CONFIg:ELEMent<x> {ALL|
<Current>[,<Current>][,<Current>]...}
[:INPut]:CURRENT:CONFIg:ELEMent<x>?
<x> = 1 to 4 (element)
ALL = all ranges enabled
<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 1,0.5,0.1,0.25
:INPUT:CURRENT:CONFIg:ELEMent1?
-> :INPUT:CURRENT:CONFIg:
ELEMent1 1.0000E+00,500.0E-03,100.0E-03,
25.0E-03

Description • List all the current ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

[[:INPut]:CURRENT:CT?

Queries all CT settings for current sensor elements.

Syntax [:INPut]:CURRENT:CT?

Description This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to CURRENT.

[[:INPut]:CURRENT:CT:PRESet?

Queries the CT presets for all elements.

Syntax [:INPut]:CURRENT:CT:PRESet?

[[:INPut]:CURRENT:CT:PRESet[:ALL]

Sets the CT preset for all elements at once.

Syntax [:INPut]:CURRENT:CT:
PRESet[:ALL] {CT2000A|CT1000A|
CT1000|CT200|CT60|CUSTom}
CUSTom = custom setting

Description This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to CURRENT.

[[:INPut]:CURRENT:CT:PRESet:ELEMent<x>

Sets or queries the CT preset for each element.

Syntax [:INPut]:CURRENT:CT:PRESet:
ELEMent<x> {CT2000A|CT1000A|CT1000|
CT200|CT60|CUSTom}
[:INPut]:CURRENT:CT:PRESet:ELEMent<x>?
<x> = 1 to 4 (element)
CUSTom = custom setting

Example :INPUT:CURRENT:CT:PRESet:
ELEMent1 CT2000A
:INPUT:CURRENT:CT:PRESet:ELEMent1?
-> :INPUT:CURRENT:CT:PRESet:
ELEMent1 CT2000A

Description • This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to CURRENT.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

[[:INPut]:CURRENT:CT:RESistance?

Queries the input resistances for all elements.

Syntax [:INPut]:CURRENT:CT:RESistance?

[[:INPut]:CURRENT:CT:RESistance[:ALL]

Sets the input resistance for all elements at once.

Syntax [:INPut]:CURRENT:CT:
RESistance[:ALL] {<Resistance>}
<Resistance> = 0.5 (Ω)

Description • This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to CURRENT.

- When this command changes the input resistance, the CT preset ([[:INPut]:CURRENT:CT:PRESet] is cleared (set to CUSTom).

5.12 INPut Group

[[:INPut]:CURRENT:CT:RESistance:ELEMENT<x>

Sets or queries the input resistance for each element.

Syntax [:INPut]:CURRENT:CT:RESistance:
ELEMENT<x> {<Resistance>}
[:INPut]:CURRENT:CT:RESistance:
ELEMENT<x>?
<x> = 1 to 4 (element)
<Resistance> = 0.5 (Ω)

Example :INPUT:CURRENT:CT:RESISTANCE:
ELEMENT1 0.5
:INPUT:CURRENT:CT:RESISTANCE:
ELEMENT1?
-> :INPUT:CURRENT:CT:RESISTANCE:
ELEMENT1 500.0E-03

Description • This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE]) is set to CURRENT.
• When this command changes the input resistance, the CT preset ([[:INPut]:CURRENT:CT:PRESet]) is cleared (set to CUSTom).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRENT:EXTSensor?

Queries all external current sensor range settings.

Syntax [:INPut]:CURRENT:EXTSensor?

[[:INPut]:CURRENT:EXTSensor:CONFig?

Queries the enabled external current sensor ranges for all elements.

Syntax [:INPut]:CURRENT:EXTSensor:CONFig?

[[:INPut]:CURRENT:EXTSensor:CONFig[:ALL]

Sets the enabled external current sensor ranges for all elements at once.

Syntax [:INPut]:CURRENT:EXTSensor:
CONFig[:ALL] {ALL|<Voltage>[,<Voltage>]
[,<Voltage>]...}
ALL = all ranges enabled
• When the crest factor setting is 3
 <Voltage> = 50, 100, 200, or 500 (mV), or 1, 2,
 or 5 (V)
• When the crest factor setting is 6 or 6A
 <Voltage> = 25, 50, 100, 250, or 500 (mV), or
 1 or 2.5 (V)

Example :INPUT:CURRENT:EXTSENSOR:CONFIG:
ALL ALL
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ALL 5,2,1

Description List all the external current sensor ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.

[[:INPut]:CURRENT:EXTSensor:CONFig:ELEMENT<x>

Sets or queries the enabled external current sensor ranges for each element.

Syntax [:INPut]:CURRENT:EXTSensor:CONFig:
ELEMENT<x> {ALL|<Voltage>[,<Voltage>]
[,<Voltage>]...}
[:INPut]:CURRENT:EXTSensor:CONFig:
ELEMENT<x>?
<x> = 1 to 4 (element)
ALL = all ranges enabled
<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 5,2,1
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1?
-> :INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1 5.00E+00,2.00E+00,1.00E+00

Description • List all the external current sensor ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRENT:EXTSensor:POJump?

Queries the range to jump to when a current peak-over occurs for all elements.

Syntax [:INPut]:CURRENT:EXTSensor:POJump?

[[:INPut]:CURRENT:EXTSensor:POJump[:ALL]

Sets the range to jump to when a current peak-over occurs for all elements at once.

Syntax [:INPut]:CURRENT:EXTSensor:
POJump[:ALL] {OFF|<Voltage>}
OFF = no current range to jump to
• When the crest factor setting is 3
 <Voltage> = 50, 100, 200, or 500 (mV), or 1, 2,
 or 5 (V)
• When the crest factor setting is 6 or 6A
 <Voltage> = 25, 50, 100, 250, or 500 (mV), or
 1 or 2.5 (V)

Example :INPUT:CURRENT:EXTSENSOR:POJUMP:
ALL OFF

[[:INPut]:CURRENT:EXTSensor:POJump:ELEMeNt<x>

Sets or queries the range to jump to when a current peak-over occurs for each element.

Syntax `[[:INPut]:CURRENT:EXTSensor:POJump:
ELEMeNt<x> {OFF|<Voltage>}
[:INPut]:CURRENT:EXTSensor:POJump:
ELEMeNt<x>?
<x> = 1 to 4 (element)
OFF = no current range to jump to
<Voltage> = see "[[:INPut]:CURRENT:EXTSensor:
POJump[:ALL]]."`

Example `:INPUT:CURRENT:EXTSENSOR:POJUMP:
ELEMENT1 5V
:INPUT:CURRENT:EXTSENSOR:POJUMP:
ELEMENT1?
-> :INPUT:CURRENT:EXTSENSOR:POJUMP:
ELEMENT1 5.00E+00`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRENT:POJump?

Queries the range to jump to when a current peak-over occurs for all elements.

Syntax `[[:INPut]:CURRENT:POJump?`

[[:INPut]:CURRENT:POJump[:ALL]

Sets the range to jump to when a current peak-over occurs for all elements at once.

Syntax `[[:INPut]:CURRENT:POJump[:ALL] {OFF|
<Current>}
OFF = no current range to jump to`

- When the crest factor setting is 3
<Current> = 25, 50, 100, 250, or 500 (mA), or 1 (A)
- When the crest factor setting is 6 or 6A
<Current> = 12.5, 25, 50, 125, 250, or 500 (mA)

Example `:INPUT:CURRENT:POJUMP:ALL OFF`

[[:INPut]:CURRENT:POJump:ELEMeNt<x>

Sets or queries the range to jump to when a current peak-over occurs for each element.

Syntax `[[:INPut]:CURRENT:POJump:
ELEMeNt<x> {OFF|<Current>}
[:INPut]:CURRENT:POJump:ELEMeNt<x>?
<x> = 1 to 4 (element)
OFF = no current range to jump to
<Current> = see
"[[:INPut]:CURRENT:POJump[:ALL]]"`

Example `:INPUT:CURRENT:POJUMP:ELEMENT1 1A
:INPUT:CURRENT:POJUMP:ELEMENT1?
-> :INPUT:CURRENT:POJUMP:
ELEMENT1 1.0000E+00`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRENT:PROBe?

Queries all current probe settings for current sensor elements.

Syntax `[[:INPut]:CURRENT:PROBe?`

Description This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to VOLTage.

[[:INPut]:CURRENT:PROBe:PRESet?

Queries the current probe presets for all elements.

Syntax `[[:INPut]:CURRENT:PROBe:PRESet?`

[[:INPut]:CURRENT:PROBe:PRESet[:ALL]

Sets the current probe preset for all elements at once.

Syntax `[[:INPut]:CURRENT:PROBe:
PRESet[:ALL] {CT1000S|720930|720931|
CUSTom}
CUSTom = custom setting`

Example `:INPUT:CURRENT:PROBE:PRESET:
ALL CUSTOM`

Description This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to VOLTage.

[[:INPut]:CURRENT:PROBe:PRESet:ELEMeNt<x>

Sets or queries the current probe preset for each element.

Syntax `[[:INPut]:CURRENT:PROBe:PRESet:
ELEMeNt<x> {CT1000S|720930|720931|
CUSTom}
[:INPut]:CURRENT:PROBe:PRESet:
ELEMeNt<x>?
<x> = 1 to 4 (element)
CUSTom = custom setting`

Example `:INPUT:CURRENT:PROBE:PRESET:
ELEMENT1 CUSTOM
:INPUT:CURRENT:PROBE:PRESET:ELEMENT1?
-> :INPUT:CURRENT:PROBE:PRESET:
ELEMENT1 CUSTOM`

Description

- This setting is valid when the current measurement terminal ([[:INPut]:CURRENT:SENSOR:OTYPE] is set to VOLTage.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.12 INPut Group

[[:INPut]:CURRent:PROBe:RATE?

Queries the output voltage rate (current probe mV/A ratio) for all elements.

Syntax `[[:INPut]:CURRent:PROBe:RATE?`

[[:INPut]:CURRent:PROBe:RATE[:ALL]

Sets the output voltage rate (current probe mV/A ratio) for all elements at once.

Syntax `[[:INPut]:CURRent:PROBe:RATE[:ALL] {<NRf>}`

`<NRf> = 0.0001 to 99999.9999`

Example `:INPUT:CURRENT:PROBE:RATE:ALL 1`

Description • This setting is valid when the current measurement terminal ([[:INPut]:CURRent:SENsOr:OTYPe]) is set to VOLTage.

- When this command changes the conversion ratio, the current probe preset ([[:INPut]:CURRent:PROBe:PRESet]) is cleared (set to CUSTom).

[[:INPut]:CURRent:PROBe:RATE:ELEMEnt<x>

Sets or queries the output voltage rate (current probe mV/A ratio) for each element.

Syntax `[[:INPut]:CURRent:PROBe:RATE:ELEMEnt<x> {<NRf>}`
`[[:INPut]:CURRent:PROBe:RATE:ELEMEnt<x>?`

`<x> = 1 to 4 (element)`

`<NRf> = 0.0001 to 99999.9999`

Example `:INPUT:CURRENT:PROBE:RATE:ELEMENT1 1`
`:INPUT:CURRENT:PROBE:RATE:ELEMENT1?`
`-> :INPUT:CURRENT:PROBE:RATE:ELEMENT1 1.0000`

Description • This setting is valid when the current measurement terminal ([[:INPut]:CURRent:SENsOr:OTYPe]) is set to VOLTage.

- When this command changes the conversion ratio, the current probe preset ([[:INPut]:CURRent:PROBe:PRESet]) is cleared (set to CUSTom).
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRent:RANGe?

Queries the current ranges for all elements.

Syntax `[[:INPut]:CURRent:RANGe?`

[[:INPut]:CURRent:RANGe[:ALL]

Sets the current range for all elements at once.

Syntax `[[:INPut]:CURRent:RANGe[:ALL] {<Voltage>| (PROBe,<Voltage>)}`

- When the crest factor setting is 3
`<Current> = 25, 50, 100, 250, or 500 (mA), or 1 (A) (when OTYPe = CURRent)`
`<Voltage> = 50, 100, 200, or 500 (mV), or 1, 2, or 5 (V) (when OTYPe = VOLTage)`
- When the crest factor setting is 6 or 6A
`<Current> = 12.5, 25, 50, 125, 250, or 500 (mA) (when OTYPe = CURRent)`
`<Voltage> = 25, 50, 100, 250, or 500 (mV), or 1 or 2.5 (V) (when OTYPe = VOLTage)`

Example `:INPUT:CURRENT:RANGE:ALL 25mA`
`:INPUT:CURRENT:RANGE:ALL PROBE,50mV`

[[:INPut]:CURRent:RANGe:ELEMEnt<x>

Sets or queries the current range for each element.

Syntax `[[:INPut]:CURRent:RANGe:ELEMEnt<x> {<Current>| (PROBe,<Voltage>)}`
`[[:INPut]:CURRent:RANGe:ELEMEnt<x>?`
`<x> = 1 to 4 (element)`
`<Current>, <Voltage> = see`

Example `:INPUT:CURRENT:RANGE:ELEMENT1 25mA`
`:INPUT:CURRENT:RANGE:ELEMENT1?`
`-> :INPUT:CURRENT:RANGE:ELEMENT1 25.0E-03`
`:INPUT:CURRENT:RANGE:ELEMENT1 PROBE,50mV`
`:INPUT:CURRENT:RANGE:ELEMENT1?`
`-> :INPUT:CURRENT:RANGE:ELEMENT1 PROBE,50.0E-03`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRent:RANGe:{SIGMA|SIGMB}

Sets the current range at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[[:INPut]:CURRent:RANGe:{SIGMA|SIGMB} {<Current>|(PROBe,<Voltage>)}`
`<Current>, <Voltage> = see`
`[[:INPut]:CURRent:RANGe[:ALL]`

Example `:INPUT:CURRENT:RANGE:SIGMA 25mA`
`:INPUT:CURRENT:RANGE:SIGMA PROBE,50mV`

Description • When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([[:INPut]:WIRing]), {SIGMA|SIGMB} is invalid.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:CURRent:SENSor:AUTO?

Queries the CT-ID automatic sensor on/off status for all elements.

Syntax `[[:INPut]:CURRent:SENSor:AUTO?`

[[:INPut]:CURRent:SENSor:AUTO[:ALL]

Sets the CT-ID automatic sensor on/off status for all elements at once.

Syntax `[[:INPut]:CURRent:SENSor:
AUTO[:ALL] {<Boolean>}`

Example `:INPUT:CURRENT:SENSOR:AUTO:AL ON`

[[:INPut]:CURRent:SENSor:AUTO:ELEMent<x>

Sets or queries the CT-ID automatic sensor on/off status for each element.

Syntax `[[:INPut]:CURRent:SENSor:AUTO:
ELEMent<x> {<Boolean>}
[[:INPut]:CURRent:SENSor:AUTO:
ELEMent<x>?`

`<x> = 1 to 4 (element)`

Example `:INPUT:CURRENT:SENSOR:AUTO:
ELEMENT1 ON
:INPUT:CURRENT:SENSOR:AUTO:ELEMENT1?
-> :INPUT:CURRENT:SENSOR:AUTO:
ELEMENT1 1`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRent:SENSor:MODEL?

Queries the sensor models for all elements.

Syntax `[[:INPut]:CURRent:SENSor:MODEL?`

[[:INPut]:CURRent:SENSor:MODEL[:ALL]

Sets the sensor model for all elements at once.

Syntax `[[:INPut]:CURRent:SENSor:
MODEL[:ALL] {CT2000A|CT1000S|CT1000A|
CT1000|CT200|CT60|720930|720931|CUSTom}
CUSTom = custom setting`

Example `:INPUT:CURRENT:SENSOR:MODEL:
ALL CT2000A`

[[:INPut]:CURRent:SENSor:MODEL:ELEMent<x>

Sets or queries the sensor model for each element.

Syntax `[[:INPut]:CURRent:SENSor:MODEL:
ELEMent<x> {CT2000A|CT1000S|CT1000A|
CT1000|CT200|CT60|720930|720931|CUSTom}
[[:INPut]:CURRent:SENSor:MODEL:
ELEMent<x>?
<x> = 1 to 4 (element)
CUSTom = custom setting`

Example `:INPUT:CURRENT:SENSOR:MODEL:
ELEMENT1 CT2000A
:INPUT:CURRENT:SENSOR:MODEL:ELEMENT1?
-> :INPUT:CURRENT:SENSOR:MODEL:
ELEMENT1 CT2000A`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRent:SENSor:OTYPE?

Queries the sensor output types for all elements.

Syntax `[[:INPut]:CURRent:SENSor:OTYPE?`

[[:INPut]:CURRent:SENSor:OTYPE[:ALL]

Sets the sensor output type for all elements at once.

Syntax `[[:INPut]:CURRent:SENSor:
OTYPE[:ALL] {<OType>}
<OType> = {CURRENT|VOLTage}`

Example `:INPUT:CURRENT:SENSOR:OTYPE:ALL CURRENT`

Description The sensor output type can be set at the same time as the current range for all elements by using the current range setting command ([[:INPut]:CURRent:RANGe[:ALL]]).

5.12 INPut Group

[[:INPut]:CURRent:SENSor:OTYPe:ELEMe nt<x>

Sets or queries the sensor output type for each element.

Syntax [:INPut]:CURRent:SENSor:OTYPe:
ELEMENT<x> {<OTYPe>}
[:INPut]:CURRent:SENSor:OTYPe:
ELEMENT<x>?
<x> = 1 to 4 (element)
<OTYPe> = see "[[:INPut]:CURRent:SENSor:OTY
Pe:ALL]"

Example :INPUT:CURRENT:SENSOR:OTYPE:
ELEMENT1 CURRENT
:INPUT:CURRENT:SENSOR:OTYPE:ELEMENT1?
-> :INPUT:CURRENT:SENSOR:OTYPE:
ELEMENT1 CURRENT

Description • The sensor output type can be set at the same time as the current range for each element by using the current range setting command ([[:INPut]:CURRent:RANGe:ELEMe<x>]).

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:CURRent:SENSor:OTYPe:{SIGMA |SIGMB}

Sets the sensor output type at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax [:INPut]:CURRent:SENSor:
OTYPe:{SIGMA|SIGMB} {CURRent|VOLTage}
<OTYPe> = see "[[:INPut]:CURRent:SENSor:OTY
Pe:ALL]"

Example :INPUT:CURRENT:SENSOR:OTYPE:
SIGMA CURRENT

Description • When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([[:INPut]:WIRing]), {SIGMA|SIGMB} is invalid.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:DEMAG:CTRL:IALL

Sets whether current is included in demagnetization operation for all elements at once.

Syntax [:INPut]:DEMAG:CTRL:IALL {<Boolean>}
Example :INPUT:DEMAG:CTRL:IALL ON

[[:INPut]:DEMAG:CTRL:I<x>

Sets or queries whether current is included in demagnetization operation for each element.

Syntax [:INPut]:DEMAG:CTRL:I<x> {<Boolean>}
[:INPut]:DEMAG:CTRL:I<x>?
<x> = 1 to 4 (element)

Example :INPUT:DEMAG:CTRL:I1 ON
:INPUT:DEMAG:CTRL:I1?
-> :INPUT:DEMAG:CTRL:I1 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:DEMAG:EXECute

Executes demagnetization.

Syntax [:INPut]:DEMAG:EXECute
Example :INPUT:DEMAG:EXECUTE

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:ESElect

Sets or queries the element for setting the measurement range.

Syntax [:INPut]:ESElect {<NRf>|OPTions}
[:INPut]:ESElect?
<NRf> = 1 to 4 (element)

Example :INPUT:ESELECT 1
:INPUT:ESELECT? -> :INPUT:ESELECT 1

Description • OPTions is for setting only. You can switch the input information area on the right side of the screen to Options.

- For queries, this command returns the element for setting the measurement range.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:FILTer?]

Queries all input filter settings.

Syntax `[[:INPut]:FILTer?]`

[[:INPut]:FILTer:FREQuency?]

Queries all frequency filter settings.

Syntax `[[:INPut]:FILTer:FREQuency?]`

[[:INPut]:FILTer:FREQuency:ADVanced]

Sets or queries the on/off status of frequency filter advanced settings.

Syntax `[[:INPut]:FILTer:FREQuency:
ADVanced {<Boolean>}]`

Example `[[:INPut]:FILTer:FREQuency:ADVanced?
:INPUT:FILTER:FREQUENCY:ADVANCED OFF
:INPUT:FILTER:FREQUENCY:ADVANCED?
-> :INPUT:FILTER:FREQUENCY:ADVANCED 0`

[[:INPut]:FILTer:FREQuency[:ALL]]

Sets the frequency filter for all elements at once.

Syntax `[[:INPut]:FILTer:
FREQuency[:ALL] {OFF|<Frequency>}]`
OFF = frequency filter off
<Frequency> = 125, 250, or 500 (Hz), or 1, 2, 4,
8, 16, or 32 (kHz) (frequency filter on, cutoff
frequency)

Example `:INPUT:FILTER:FREQUENCY:ALL OFF`

Description This setting is valid when advanced settings ([[:INPut]:FILTer:FREQuency:ADVanced]) are OFF (0).

[[:INPut]:FILTer:FREQuency:ELEMeNt<x>]

Sets or queries the frequency filter for each element.

Syntax `[[:INPut]:FILTer:FREQuency:
ELEMeNt<x> {OFF|<Frequency>}]`
[[:INPut]:FILTer:FREQuency:
ELEMeNt<x>?]
<x> = 1 to 4 (element)
OFF = frequency filter off
<Frequency> = 125, 250, or 500 (Hz), or 1, 2, 4,
8, 16, or 32 (kHz) (frequency filter on, cutoff
frequency)

Example `:INPUT:FILTER:FREQUENCY:
ELEMENT1 125HZ
:INPUT:FILTER:FREQUENCY:ELEMENT1?
-> :INPUT:FILTER:FREQUENCY:
ELEMENT1 125.0E+00`

Description • This setting is valid when advanced settings ([[:INPut]:FILTer:FREQuency:ADVanced]) are OFF (0).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYNc:SUNit.

[[:INPut]:FILTer:FREQuency:LPFilter?]

Queries all low-pass filter settings. (Advanced settings)

Syntax `[[:INPut]:FILTer:FREQuency:LPFilter?]`

Description This setting is valid when advanced settings ([[:INPut]:FILTer:FREQuency:ADVanced]) are ON (1).

[[:INPut]:FILTer:FREQuency:LPFilter[:ALL]]

Sets the low-pass filter for all elements at once. (Advanced settings)

Syntax `[[:INPut]:FILTer:FREQuency:
LPFilter[:ALL] {OFF|<Frequency>}]`
OFF = low-pass filter off
<Frequency> = 125, 250, or 500 (Hz), or 1, 2,
4, 8, 16, or 32 (kHz) (low-pass filter on, cutoff
frequency)

Example `:INPUT:FILTER:FREQUENCY:LPFILTER:
ALL OFF`

Description This setting is valid when advanced settings ([[:INPut]:FILTer:FREQuency:ADVanced]) are ON (1).

[[:INPut]:FILTer:FREQuency:LPFilter:AUTO?]

Queries the auto frequency filter on/off status for all elements.

Syntax `[[:INPut]:FILTer:FREQuency:LPFilter:
AUTO?]`

[[:INPut]:FILTer:FREQuency:LPFilter:AUTO[:ALL]]

Sets the auto frequency filter on/off status for all elements at once.

Syntax `[[:INPut]:FILTer:FREQuency:LPFilter:
AUTO[:ALL] {<Boolean>}]`

Example `:INPUT:FILTER:FREQUENCY:LPFILTER:
AUTO:ALL ON`

[[:INPut]:FILTer:FREQuency:LPFilter:AUTO:ELEMeNt<x>]

Sets or queries the auto frequency filter on/off status for each element.

Syntax `[[:INPut]:FILTer:FREQuency:LPFilter:
AUTO:ELEMeNt<x> {<Boolean>}]`
[[:INPut]:FILTer:FREQuency:LPFilter:
AUTO:ELEMeNt<x>?]
<x> = 1 to 4 (element)

Example `:INPUT:FILTER:FREQUENCY:LPFILTER:
AUTO:ELEMENT1 ON
:INPUT:FILTER:FREQUENCY:LPFILTER:
AUTO:ELEMENT1?
-> :INPUT:FILTER:FREQUENCY:LPFILTER:
AUTO:ELEMENT1 1`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYNc:SUNit.

5.12 INPut Group

[[:INPut]:FILTer:FREQuency:LPFilter:ELEMent<x>

Sets or queries the low-pass filter for each element. (Advanced settings)

Syntax `[[:INPut]:FILTer:FREQuency:LPFilter:ELEMent<x> {OFF|<Frequency>}
[[:INPut]:FILTer:FREQuency:LPFilter:ELEMent<x>?
<x> = 1 to 4 (element)
OFF = low-pass filter off
<Frequency> = 125, 250, or 500 (Hz), or 1, 2, 4, 8, 16, or 32 (kHz) (low-pass filter on, cutoff frequency)`

Example `:INPUT:FILTER:FREQUENCY:LPFILTER:
ELEMENT1 125HZ
:INPUT:FILTER:FREQUENCY:LPFILTER:
ELEMENT1?
-> :INPUT:FILTER:FREQUENCY:LPFILTER:
ELEMENT1 125.0E+00`

Description • This setting is valid when advanced settings ([[:INPut]:FILTer:FREQuency:ADVanced]) are ON (1).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:FILTer:LINE?

Queries all line filter settings.

Syntax `[[:INPut]:FILTer:LINE?`

[[:INPut]:FILTer[:LINE]:AAFilter?

Queries all anti-aliasing filter settings. (Advanced settings)

Syntax `[[:INPut]:FILTer[:LINE]:AAFilter?`

Description This setting is valid when advanced settings ([[:INPut]:FILTer[:LINE]:ADVanced]) are ON (1).

[[:INPut]:FILTer[:LINE]:AAFilter[:ALL]

Sets the anti-aliasing filter for all elements at once. (Advanced settings)

Syntax `[[:INPut]:FILTer[:LINE]:
AAFilter[:ALL] {<Boolean>}`

Example `:INPUT:FILTER:LINE:AAFILTER:ALL OFF`

Description This setting is valid when advanced settings ([[:INPut]:FILTer[:LINE]:ADVanced]) are ON (1).

[[:INPut]:FILTer[:LINE]:AAFilter:ELEMent<x>

Sets or queries the anti-aliasing filter for each element. (Advanced settings)

Syntax `[[:INPut]:FILTer[:LINE]:AAFilter:
ELEMent<x> {<Boolean>}
[[:INPut]:FILTer[:LINE]:AAFilter:
ELEMent<x>?
<x> = 1 to 4 (element)`

Example `:INPUT:FILTER:LINE:AAFILTER:
ELEMENT1 ON
:INPUT:FILTER:LINE:AAFILTER:
ELEMENT1?
-> :INPUT:FILTER:LINE:AAFILTER:
ELEMENT1 1`

Description • This setting is valid when advanced settings ([[:INPut]:FILTer[:LINE]:ADVanced]) are ON (1).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:FILTer[:LINE]:ADVanced

Sets or queries the on/off status of line filter advanced settings.

Syntax `[[:INPut]:FILTer[:LINE]:
ADVanced {<Boolean>}
[[:INPut]:FILTer[:LINE]:ADVanced?`

Example `:INPUT:FILTER:LINE:ADVANCED OFF
:INPUT:FILTER:LINE:ADVANCED?
-> :INPUT:FILTER:LINE:ADVANCED 0`

[[:INPut]:FILTer[:LINE]:[ALL]

Sets the line filter for all elements at once.

Syntax `[[:INPut]:FILTer[:LINE]:[ALL] {OFF|
<Frequency>}
OFF = line filter off
<Frequency> = 0.1 to 200.0 or 300 (kHz) (line filter on, cutoff frequency)`

Example `:INPUT:FILTER:LINE:ALL OFF`

Description • This setting is valid when advanced settings ([[:INPut]:FILTer[:LINE]:ADVanced]) are OFF (0).
• Values from 0.1 to 200.0 (kHz) can be set with a resolution of 0.1 (kHz).

[[:INPut]:FILTer[:LINE]:ELEMeNt<x>

Sets or queries the line filter for each element.

Syntax [:INPut]:FILTer[:LINE]:
ELEMeNt<x> {OFF|<Frequency>}
[:INPut]:FILTer[:LINE]:ELEMeNt<x>?
<x> = 1 to 4 (element)
OFF = line filter off
<Frequency> = 0.1 to 200.0 or 300 (kHz) (line
filter on, cutoff frequency)

Example :INPUT:FILTER:LINE:ELEMENT1 0.5KHZ
:INPUT:FILTER:LINE:ELEMENT1?
-> :INPUT:FILTER:LINE:
ELEMENT1 500.0E+00

Description • This setting is valid when advanced settings
([[:INPut]:FILTer[:LINE]:ADVanced]) are OFF (0).
• Values from 0.1 to 200.0 (kHz) can be set with
a resolution of 0.1 (kHz).
• For multi-unit synchronization (option
/SY1), specify the target unit with
:SYSTem:MUSYnc:SUNit.

[[:INPut]:FILTer[:LINE]:DLFilter?

Queries all digital line filter settings. (Advanced settings)

Syntax [:INPut]:FILTer[:LINE]:DLFilter?

Description This setting is valid when advanced settings ([[:IN
Put]:FILTer[:LINE]:ADVanced]) are ON (1).

**[[:INPut]:FILTer[:LINE]:DLFilter:{NOR
mal|HARMonics}?**

Queries all digital line filter settings for {normal
measurement|harmonic measurement}. (Advanced settings)

Syntax [:INPut]:FILTer[:LINE]:DLFilter:
{NORmal|HARMonics}?

Description This setting is valid when advanced settings ([[:IN
Put]:FILTer[:LINE]:ADVanced]) are ON (1).

**[[:INPut]:FILTer[:LINE]:DLFilter:{NOR
mal|HARMonics}[:ALL]**

Sets the digital line filter for {normal measurement|harmonic
measurement} for all elements at once. (Advanced settings)

Syntax [:INPut]:FILTer[:LINE]:DLFilter:
{NORmal|HARMonics}[:ALL] {OFF|
<Frequency>}
OFF = digital line filter off
<Frequency> = 0.1 to 200.0 (kHz) (digital line
filter on, cutoff frequency)

Example :INPUT:FILTER:LINE:DLFILTER:NORMAL:
ALL OFF

Description This setting is valid when advanced settings ([[:IN
Put]:FILTer[:LINE]:ADVanced]) are ON (1).

**[[:INPut]:FILTer[:LINE]:DLFilter:{NOR
mal|HARMonics}:ELEMeNt<x>**

Sets or queries the digital line filter for {normal
measurement|harmonic measurement} for each element.
(Advanced settings)

Syntax [:INPut]:FILTer[:LINE]:DLFilter:
{NORmal|HARMonics}:ELEMeNt<x> {OFF|
<Frequency>}
[:INPut]:FILTer[:LINE]:DLFilter:
{NORmal|HARMonics}:ELEMeNt<x>?
<x> = 1 to 4 (element)
OFF = digital line filter off
<Frequency> = 0.1 to 200.0 (kHz) (digital line
filter on, cutoff frequency)

Example :INPUT:FILTER:LINE:DLFILTER:NORMAL:
ELEMENT1 0.5KHZ
:INPUT:FILTER:LINE:DLFILTER:NORMAL:
ELEMENT1?
-> :INPUT:FILTER:LINE:DLFILTER:
NORMAL:ELEMENT1 500.0E+00

Description • This setting is valid when advanced settings
([[:INPut]:FILTer[:LINE]:ADVanced]) are ON (1).
• For multi-unit synchronization (option
/SY1), specify the target unit with
:SYSTem:MUSYnc:SUNit.

**[[:INPut]:FILTer[:LINE]:DLFilter:HARM
onics:AUTO?**

Queries the auto harmonic filter on/off status for all elements.

Syntax [:INPut]:FILTer[:LINE]:DLFilter:
HARMonics:AUTO?

Description This setting is valid when advanced settings ([[:IN
Put]:FILTer[:LINE]:ADVanced]) are ON (1).

**[[:INPut]:FILTer[:LINE]:DLFilter:HARM
onics:AUTO[:ALL]**

Sets the auto harmonic filter on/off status for all elements at
once.

Syntax [:INPut]:FILTer[:LINE]:DLFilter:
HARMonics:AUTO[:ALL] {<Boolean>}

Example :INPUT:FILTER:LINE:DLFILTER:HARMONICS:A
UTO:ALL ON

Description This setting is valid when advanced settings ([[:IN
Put]:FILTer[:LINE]:ADVanced]) are ON (1).

5.12 INPut Group

[:INPut] :FILTer [:LINE] :DLFilter :HARMonics :AUTO :ELEMeNt < x >

Sets or queries the auto harmonic filter on/off status for each element.

Syntax [:INPut] :FILTer [:LINE] :DLFilter :
 HARMonics :AUTO :ELEMeNt < x > { < Boolean > }
 [:INPut] :FILTer [:LINE] :DLFilter :
 HARMonics :AUTO :ELEMeNt < x > ?
 < x > = 1 to 4 (element)

Example :INPUT:FILTER:LINE:DLFILTER:
 HARMONICS:AUTO:ELEMENT1 ON
 :INPUT:FILTER:LINE:DLFILTER:
 HARMONICS:AUTO:ELEMENT1?
 -> :INPUT:FILTER:LINE:DLFILTER:
 HARMONICS:AUTO:ELEMENT1 1

Description • This setting is valid when advanced settings
 ([:INPut] :FILTer [:LINE] :ADVanced) are ON (1).
 • For multi-unit synchronization (option
 /SY1), specify the target unit with
 :SYSTem:MUSYnc:SUNit.

[:INPut] :MODUle ?

Queries the input element type.

Syntax [:INPut] :MODUle ? { < NRf > }
 [:INPut] :MODUle ?
 < NRf > = 1 to 4 (element)

Example :INPUT:MODULE? 1 -> HV
 :INPUT:MODULE? -> HV,HV,HV,0

Description • The response contents are as follows.
 HV = HV element
 WB = WB element
 0 = no input element
 • If the parameter is omitted, this command
 outputs the input element types for all elements
 in order from element 1.
 • For multi-unit synchronization (option
 /SY1), specify the target unit with
 :SYSTem:MUSYnc:SUNit.

[:INPut] :MTR < x > [:CONFig]

Sets or queries the configuration of MTR1 and MTR2.

Syntax [:INPut] :MTR < x > [:CONFig] { SINGLE,
 < Type > | DOUBle | AUXiliary }
 [:INPut] :MTR < x > :CONFig?
 < x > = 1 to 2 (MTR option number)
 < Type > = { ANALog | PULSe | RESolver } (rotation
 signal input type)

Example :INPUT:MTR1:CONFIG SINGLE,ANALOG
 :INPUT:MTR1:CONFIG DOUBLE
 :INPUT:MTR1:CONFIG AUXILIARY
 :INPUT:MTR1:CONFIG?
 -> :INPUT:MTR1:CONFIG SINGLE,ANALOG

Description • This command is valid on models with the
 motor evaluation function 1 for encoders (option
 /EM1), motor evaluation function 1 for resolvers
 (option /RM1), motor evaluation function 2
 for encoders (option /EM2), motor evaluation
 function 2 for resolvers (option /RM2), or motor
 evaluation function for encoders/resolvers
 (option /EM1RM1).
 • For the motor evaluation function for encoders,
 < Type > = { ANALog | PULSe }
 • For the motor evaluation function for resolvers,
 only < Type > = { RESolver } is valid, and
 { DOUBle } is invalid.
 • For multi-unit synchronization (option
 /SY1), specify the target unit with
 :SYSTem:MUSYnc:SUNit.

[[:INPut]:NULL:CONDition:{SPEEd<x>|TORQue<x>|AUX<x>}?]

Queries the NULL operation status of {rotation speed|torque|AUX}.

Syntax [[:INPut]:NULL:CONDition:{SPEEd<x>|TORQue<x>|AUX<x>}?
 <x> of SPEEd<x> = 1 to 4 (Speed input channel)
 <x> of TORQue<x> = 1 to 4 (Torque input channel)
 <x> of AUX<x> = 1 to 8 (AUX input channel)

Example :INPUT:NULL:CONDITION:SPEED? -> 1

Description

- The response contents are as follows.
 0 = NULL function off
 1 = NULL function in operation
- {SPEEd<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
- The values <x> = 3 or 4 for {SPEEd<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:NULL:CONDition:{U<x>|I<x>}?]

Queries the NULL operation status of {voltage|current} for each element.

Syntax [[:INPut]:NULL:CONDition:{U<x>|I<x>}?
 <x> = 1 to 4 (element)

Example :INPUT:NULL:CONDITION:U1? -> 1

Description The response contents are as follows.
 0 = NULL function off
 1 = NULL function in operation

[[:INPut]:NULL:CTRL:{SPEEd<x>|TORQue<x>|AUX<x>}]

Sets or queries whether {rotation speed|torque|AUX} is included in NULL operation.

Syntax [[:INPut]:NULL:CTRL:{SPEEd<x>|TORQue<x>|AUX<x>} {ON|OFF}
 [[:INPut]:NULL:CTRL:{SPEEd<x>|TORQue<x>|AUX<x>}?
 <x> of SPEEd<x> = 1 to 4 (Speed input channel)
 <x> of TORQue<x> = 1 to 4 (Torque input channel)
 <x> of AUX<x> = 1 to 8 (AUX input channel)
 ON = NULL function enabled
 OFF = NULL function disabled

Example :INPUT:NULL:CTRL:SPEED ON
 :INPUT:NULL:CTRL:SPEED?
 -> :INPUT:NULL:CTRL:SPEED ON

Description

- {SPEEd<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
- The values <x> = 3 or 4 for {SPEEd<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:NULL:CTRL:{U<x>|I<x>}]

Sets or queries whether {voltage|current} for each element is included in NULL operation.

Syntax [[:INPut]:NULL:CTRL:{U<x>|I<x>} {ON|OFF}
 [[:INPut]:NULL:CTRL:{U<x>|I<x>}?
 <x> = 1 to 4 (element)
 ON = NULL function enabled
 OFF = NULL function disabled

Example :INPUT:NULL:CTRL:U1 ON
 :INPUT:NULL:CTRL:U1?
 -> :INPUT:NULL:CTRL:U1 ON

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.12 INPut Group

[:INPut] :NULL:NOTify

Sets or queries the on/off status of the NULL notification function after the warm-up time has elapsed.

Syntax [:INPut] :NULL:NOTify { <Boolean> }
 [:INPut] :NULL:NOTify?

Example :INPUT:NULL:NOTIFY ON
 :INPUT:NULL:NOTIFY?
 -> :INPUT:NULL:NOTIFY 1

[:INPut] :NULL[:STATE]

Sets or queries the NULL function on/off status.

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 NULL operation targets.

Syntax [:INPut] :NULL:TARGET?

[:INPut] :NULL:TARGET[:MODE]

Sets or queries the selection mode for items to include in NULL operation.

Syntax [:INPut] :NULL:TARGET[:MODE] { ALL |
 SElect }

Example :INPUT:NULL:TARGET:MODE ALL
 :INPUT:NULL:TARGET:MODE?
 -> :INPUT:NULL:TARGET:MODE ALL

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :NULL:TARGET: { SPEed<x> | TORQue<x> | AUX<x> }

Sets or queries the NULL operation setting for {rotation speed|torque|AUX}.

Syntax [:INPut] :NULL:TARGET: { SPEed<x> |
 TORQue<x> | AUX<x> } { ON | HOLD | OFF }
 [:INPut] :NULL:TARGET: { SPEed<x> |
 TORQue<x> | AUX<x> } ?
 <x> of SPEed<x> = 1 to 4 (Speed input channel)
 <x> of TORQue<x> = 1 to 4 (Torque input channel)
 <x> of AUX<x> = 1 to 8 (AUX input channel)
 ON = NULL function enabled (acquires a new NULL value when NULL is turned on)
 HOLD = NULL function enabled (continues using the previous NULL value when NULL is turned on)
 OFF = NULL function disabled (excluded from NULL operation)

Example :INPUT:NULL:TARGET:SPEED ON
 :INPUT:NULL:TARGET:SPEED?
 -> :INPUT:NULL:TARGET:SPEED ON

Description • { SPEed<x> | TORQue<x> | AUX<x> } is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.
• The values <x> = 3 or 4 for { SPEed<x> | TORQue<x> } and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :NULL:TARGET: { U<x> | I<x> }

Sets or queries the NULL operation setting for {voltage|current} for each element.

Syntax [:INPut] :NULL:TARGET: { U<x> | I<x> } { ON |
 HOLD | OFF }
 [:INPut] :NULL:TARGET: { U<x> | I<x> } ?
 <x> = 1 to 4 (element)
 ON = NULL function enabled (acquires a new NULL value when NULL is turned on)
 HOLD = NULL function enabled (continues using the previous NULL value when NULL is turned on)
 OFF = NULL function disabled (excluded from NULL operation)

Example :INPUT:NULL:TARGET:U1 ON
 :INPUT:NULL:TARGET:U1?
 -> :INPUT:NULL:TARGET:U1 ON

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :NULL:TARGET: {UALL|IALL}

Sets the NULL operation setting for {voltage|current} for all elements at once.

Syntax [:INPut] :NULL:TARGET: {UALL|IALL} {ON|HOLD|OFF}

ON = NULL function enabled (acquires a new NULL value when NULL is turned on)

HOLD = NULL function enabled (continues using the previous NULL value when NULL is turned on)

OFF = NULL function disabled (excluded from NULL operation)

Example :INPUT:NULL:TARGET:UALL ON

[:INPut] :NULL:VALue: {SPEed<x>|TORQue<x>|AUX<x>}

Sets or queries the NULL operation status NEW/HOLD of {rotation speed|torque|AUX}.

Syntax [:INPut] :NULL:VALue: {SPEed<x>|TORQue<x>|AUX<x>} {NEW|HOLD}

[:INPut] :NULL:VALue: {SPEed<x>|TORQue<x>|AUX<x>}?

<x> of SPEed<x> = 1 to 4 (Speed input channel)

<x> of TORQue<x> = 1 to 4 (Torque input channel)

<x> of AUX<x> = 1 to 8 (AUX input channel)

NEW = acquires a new NULL value when NULL is turned on

HOLD = continues using the previous NULL value when NULL is turned on

Example :INPUT:NULL:VALue:SPEED NEW
:INPUT:NULL:VALue:SPEED?
-> :INPUT:NULL:VALue:SPEED NEW

Description • {SPEed<x>|TORQue<x>|AUX<x>} is valid when the encoder motor evaluation 1 option (/EM1), resolver motor evaluation 1 option (/RM1), or encoder/resolver motor evaluation option (/EM1RM1) is installed.

- The values <x> = 3 or 4 for {SPEed<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :NULL:VALue: {U<x>|I<x>}

Sets or queries the NULL operation status NEW/HOLD of {voltage|current} for each element.

Syntax [:INPut] :NULL:VALue: {U<x>|I<x>} {NEW|HOLD}

[:INPut] :NULL:VALue: {U<x>|I<x>}?

<x> = 1 to 4 (element)

NEW = acquires a new NULL value when NULL is turned on

HOLD = continues using the previous NULL value when NULL is turned on

Example :INPUT:NULL:VALue:U1 NEW
:INPUT:NULL:VALue:U1?
-> :INPUT:NULL:VALue:U1 NEW

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :POVer?

Queries peak-over information.

Syntax [:INPut] :POVer?

Example :INPUT:POVER? -> 0

Description • Peak-over information for each element is assigned as follows.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
0	0	0	0	0	0	0	0	ch.H	ch.G	ch.F	ch.E	ch.D	ch.C	ch.B	ch.A	
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	I4	U4	I3	U3	I2	U2	I1	U1

ch.A to ch.H: MTR1/MTR2 input

- The response returns the sum of the decimal values of each bit.

Example If the response is 16, a peak-over has occurred in U3.

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :RLINKage

Sets or queries the on/off status of range Σ link, which links measurement ranges within wiring unit Σ .

Syntax [:INPut] :RLINKage {<Boolean>}

[:INPut] :RLINKage?

Example :INPUT:RLINKAGE ON
:INPUT:RLINKAGE?
-> :INPUT:RLINKAGE 1

Description This setting is valid only when two or more input elements are installed.

5.12 INPut Group

[:INPut] :SCALing?

Queries all scaling settings.

Syntax `[ :INPut ] :SCALing?`

[:INPut] :SCALing:STATE?

Queries the scaling on/off status for all elements.

Syntax `[ :INPut ] :SCALing:STATE?`

[:INPut] :SCALing [:STATE] [:ALL]

Sets the scaling on/off status for all elements at once.

Syntax `[ :INPut ] :SCALing [ :STATE ] [ :ALL ] { <Boolean> }`

Example `:INPUT:SCALING:STATE:ALL OFF`

[:INPut] :SCALing [:STATE] :ELEMENT<x>

Sets or queries the scaling on/off status for each element.

Syntax `[ :INPut ] :SCALing [ :STATE ] :ELEMENT<x> { <Boolean> }`
`[ :INPut ] :SCALing [ :STATE ] :ELEMENT<x>?`
`<x> = 1 to 4 (element)`

Example `:INPUT:SCALING:STATE:ELEMENT1 OFF`
`:INPUT:SCALING:STATE:ELEMENT1?`
`-> :INPUT:SCALING:STATE:ELEMENT1 0`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :SCALing: {VT|CT|SFACtor}?

Queries the {VT ratio|CT ratio|power coefficient} for all elements.

Syntax `[ :INPut ] :SCALing: {VT|CT|SFACtor}?`

[:INPut] :SCALing: {VT|CT|SFACtor} [:ALL]

Sets the {VT ratio|CT ratio|power coefficient} for all elements at once.

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|power coefficient} for each element.

Syntax `[ :INPut ] :SCALing: {VT|CT|SFACtor} :ELEMENT<x> { <NRf> }`
`[ :INPut ] :SCALing: {VT|CT|SFACtor} :ELEMENT<x>?`
`<x> = 1 to 4 (element)`
`<NRf> = 0.0001 to 99999.9999`

Example `:INPUT:SCALING:VT:ELEMENT1 1`
`:INPUT:SCALING:VT:ELEMENT1?`
`-> :INPUT:SCALING:VT:ELEMENT1 1.0000`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :SCALing: {VT|CT|SFACtor} : {SIGMA|SIGMB}

Sets the {VT ratio|CT ratio|power coefficient} at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[ :INPut ] :SCALing: {VT|CT|SFACtor} : {SIGMA|SIGMB} { <NRf> }`
`<NRf> = 0.0001 to 99999.9999`

Example `:INPUT:SCALING:VT:SIGMA 1`

Description • When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([:INPut]:WIRing), {SIGMA|SIGMB} is invalid.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :SYNChronize?

Queries the sync sources for all elements.

Syntax `[ :INPut ] :SYNChronize?`

[:INPut] :SYNChronize [:ALL]

Sets the sync source for all elements at once.

Syntax `[ :INPut ] :SYNChronize [ :ALL ] { U<x> | I<x> | EXTErnal | Z1 | Z3 | NONE | CZCRoss }`
`<x> = 1 to 7 (element)`
EXTernal = external clock input (Ext Clk)
Z1 = Z Phase1 (ChD)
(/EM1, /RM1, or /EM1RM1 option required)
Z3 = Z Phase3 (ChH)
(/EM2, /RM2, or /EM1RM1 option required)
NONE = no sync source
CZCRoss = common zero crossing (valid only during multi-unit synchronous operation (option /SY1))

Example `:INPUT:SYNCHRONIZE:ALL I1`

[[:INPut]:SYNChronize:CZCRoss:SOURce

Sets or queries the common zero-cross source for multi-unit synchronization.

Syntax [:INPut]:SYNChronize:CZCRoss:
 SOURce {U<x>|I<x>|EXTernal|Z1|Z3}
 [:INPut]:SYNChronize:CZCRoss:SOURce?
 <x> = 1 to 4 (element)
 EXTernal = external clock input (Ext Clk)
 Z1 = Z Phase1 (ChD)
 (/EM1, /RM1, or /EM1RM1 option required)
 Z3 = Z Phase3 (ChH)
 (/EM2, /RM2, or /EM1RM1 option required)

Example :INPUT:SYNCHRONIZE:CZCROSS:
 SOURCE U1
 :INPUT:SYNCHRONIZE:CZCROSS:SOURCE?
 -> :INPUT:SYNCHRONIZE:CZCROSS:
 SOURCE U1

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:SYNChronize:CZCRoss:UNIT

Sets or queries the common zero-cross unit for multi-unit synchronization.

Syntax [:INPut]:SYNChronize:CZCRoss:
 UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}
 [:INPut]:SYNChronize:CZCRoss:UNIT?
Example :INPUT:SYNCHRONIZE:CZCROSS:UNIT MAIN
 :INPUT:SYNCHRONIZE:CZCROSS:UNIT?
 -> :INPUT:SYNCHRONIZE:CZCROSS:
 UNIT MAIN

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:SYNChronize:ELEMeNt<x>

Sets or queries the sync source for each element.

Syntax [:INPut]:SYNChronize:
 ELEMeNt<x> {U<x>|I<x>|EXTernal|Z1|
 Z3|NONE|CZCRoss}
 [:INPut]:SYNChronize:ELEMeNt<x>?
 <x> = 1 to 4 (element)
 EXTernal = external clock input (Ext Clk)
 Z1 = Z Phase1 (ChD)
 (/EM1, /RM1, or /EM1RM1 option required)
 Z3 = Z Phase3 (ChH)
 (/EM2, /RM2, or /EM1RM1 option required)
 NONE = no sync source
 CZCRoss = common zero crossing (valid only
 during multi-unit synchronous operation (option
 /SY1))

Example :INPUT:SYNCHRONIZE:ELEMENT1 I1
 :INPUT:SYNCHRONIZE:ELEMENT1?
 -> :INPUT:SYNCHRONIZE:ELEMENT1 I1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:SYNChronize:{SIGMA|SIGMB}

Sets the sync source at once for the elements belonging to wiring unit {ZA|ZB}.

Syntax [:INPut]:SYNChronize:{SIGMA|
 SIGMB} {U<x>|I<x>|EXTernal|Z1|Z3|
 NONE|CZCRoss}
 <x> = 1 to 4 (element)
 EXTernal = external clock input (Ext Clk)
 Z1 = Z Phase1 (ChD)
 (/EM1, /RM1, or /EM1RM1 option required)
 Z3 = Z Phase3 (ChH)
 (/EM2, /RM2, or /EM1RM1 option required)
 NONE = no sync source
 CZCRoss = common zero crossing (valid only
 during multi-unit synchronous operation (option
 /SY1))

Example :INPUT:SYNCHRONIZE:SIGMA I1

Description • When wiring unit {ZA|ZB } does not exist because of the wiring system setting ([[:INPut]:WIRing]), {SIGMA|SIGMB} is invalid.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.12 INPut Group

[[:INPut]:SYNChronize:LEVel?

Queries all sync source level settings.

Syntax `[[:INPut]:SYNChronize:LEVel?`

[[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent}[:ALL]

Sets the {voltage|current} sync source level for all elements at once.

Syntax `[[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent}[:ALL] {<Nrf>}`
`<Nrf> = -100.0 to 100.0 (%)`

Example `:INPUT:SYNCHRONIZE:LEVEL:VOLTAGE:`
`ALL 10`

[[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent}:ELEment<x>

Sets or queries the {voltage|current} sync source level for each element.

Syntax `[[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent}:ELEment<x> {<Nrf>}`
`[[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent}:ELEment<x>?`
`<x> = 1 to 4 (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`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:SYNChronize:RECTifier?

Queries all sync source rectifier on/off settings.

Syntax `[[:INPut]:SYNChronize:RECTifier?`

[[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent}[:ALL]

Sets {voltage|current} sync source rectifier on/off for all elements at once.

Syntax `[[:INPut]:SYNChronize:RECTifier:`
`{VOLTage|CURRent}[:ALL] {<Boolean>}`

Example `:INPUT:SYNCHRONIZE:RECTIFIER:`
`VOLTAGE:ALL OFF`

[[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent}:ELEment<x>

Sets or queries the on/off status of sync source rectification for {voltage|current} for each element.

Syntax `[[:INPut]:SYNChronize:RECTifier:`
`{VOLTage|CURRent}:`
`ELEment<x> {<Boolean>}`
`[[:INPut]:SYNChronize:RECTifier:`
`{VOLTage|CURRent}:ELEment<x>?`
`<x> = 1 to 4 (element)`

Example `:INPUT:SYNCHRONIZE:RECTIFIER:`
`VOLTAGE:ELEMNT1 OFF`
`:INPUT:SYNCHRONIZE:RECTIFIER:`
`VOLTAGE:ELEMNE1?`
`-> :INPUT:SYNCHRONIZE:RECTIFIER:`
`VOLTAGE:ELEMNT1 0`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:VOLTage?

Queries all voltage measurement settings.

Syntax `[[:INPut]:VOLTage?`

[[:INPut]:VOLTage:AUTO?

Queries the auto voltage range on/off status for all elements.

Syntax `[[:INPut]:VOLTage:AUTO?`

[[:INPut]:VOLTage:AUTO[:ALL]

Sets the auto voltage range on/off status for all elements at once.

Syntax `[[:INPut]:VOLTage:`
`AUTO[:ALL] {<Boolean>}`

Example `:INPUT:VOLTAGE:AUTO:ALL ON`

[[:INPut]:VOLTage:AUTO:ELEment<x>

Sets or queries the auto voltage range on/off status for each element.

Syntax `[[:INPut]:VOLTage:AUTO:`
`ELEment<x> {<Boolean>}`
`[[:INPut]:VOLTage:AUTO:ELEment<x>?`
`<x> = 1 to 4 (element)`

Example `:INPUT:VOLTAGE:AUTO:ELEMENT1 ON`
`:INPUT:VOLTAGE:AUTO:ELEMENT1?`
`-> :INPUT:VOLTAGE:AUTO:ELEMENT1 1`

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB}]

Sets the auto voltage range on/off status at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax [:INPut]:VOLTage:AUTO:{SIGMA|SIGMB} {<Boolean>}

Example :INPUT:VOLTAGE:AUTO:SIGMA ON

Description

- When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([[:INPut]:WIRing]), {SIGMA|SIGMB} is invalid.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:VOLTage:CONFig?]

Queries the enabled voltage ranges for all elements.

Syntax [:INPut]:VOLTage:CONFig?

[[:INPut]:VOLTage:CONFig[:ALL]]

Sets the enabled voltage ranges for all elements at once.

Syntax [:INPut]:VOLTage:CONFig[:ALL] {ALL|<Voltage>[,<Voltage>][,<Voltage>]...}

ALL = all ranges enabled

- When the crest factor setting is 3
HV: <Voltage> = 6, 15, 30, 60, 150, 300, 600, 1000, or 2000 (V)
WB: <Voltage> = 6, 15, 30, 60, 150, 300, 600, or 1000 (V)
- When the crest factor setting is 6 or 6A
HV: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300, 500, or 1000 (V)
WB: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300, or 500 (V)

Example :INPUT:VOLTAGE:CONFIG:ALL ALL
:INPUT:VOLTAGE:CONFIG:ALL 1000,300,150,30,15

Description List all the voltage ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.

[[:INPut]:VOLTage:CONFig:ELEMEnt<x>]

Sets or queries the enabled voltage ranges for each element.

Syntax [:INPut]:VOLTage:CONFig:
ELEMEnt<x> {ALL|<Voltage>[,<Voltage>]
[,<Voltage>]...}

[:INPut]:VOLTage:CONFig:ELEMEnt<x>?

<x> = 1 to 4 (element)

ALL = all ranges enabled

<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,
300,150,30,15
:INPUT:VOLTAGE:CONFIG:ELEMENT1?
-> :INPUT:VOLTAGE:CONFIG:
ELEMENT1 1.0000E+03,300.0E+00,
150.0E+00,30.0E+00,15.0E+00
```

Description

- List all the voltage ranges to enable as parameters. To enable all ranges, specify ALL as the parameter.
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

[[:INPut]:VOLTage:POJump?]

Queries the range to jump to when a voltage peak-over occurs for all elements.

Syntax [:INPut]:VOLTage:POJump?

[[:INPut]:VOLTage:POJump[:ALL]]

Sets the range to jump to when a voltage peak-over occurs for all elements at once.

Syntax [:INPut]:VOLTage:POJump[:ALL] {OFF|<Voltage>}

OFF = no voltage range to jump to

- When the crest factor setting is 3
HV: <Voltage> = 6, 15, 30, 60, 150, 300, 600, 1000, or 2000 (V)
WB: <Voltage> = 6, 15, 30, 60, 150, 300, 600, or 1000 (V)
- When the crest factor setting is 6 or 6A
HV: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300, 500, or 1000 (V)
WB: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300, or 500 (V)

Example :INPUT:VOLTAGE:POJUMP:ALL OFF

5.12 INPut Group

[:INPut] :VOLTagE:POJump:ELEMeNt<x>

Sets or queries the range to jump to when a voltage peak-over occurs for each element.

Syntax [:INPut] :VOLTagE:POJump:
 ELEMeNt<x> {OFF|<Voltage>}
 [:INPut] :VOLTagE:POJump:ELEMeNt<x>?
 <x> = 1 to 4 (element)
 OFF = no voltage range to jump to
 <Voltage> = see "[:INPut] :VOLTagE:POJump[:ALL]."

Example :INPUT:VOLTAGE:POJUMP:ELEMENT1 1000V
 :INPUT:VOLTAGE:POJUMP:ELEMENT1?
 -> :INPUT:VOLTAGE:POJUMP:
 ELEMENT1 1.0000E+03

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :VOLTagE:RANGe?

Queries the voltage ranges for all elements.

Syntax [:INPut] :VOLTagE:RANGe?

[:INPut] :VOLTagE:RANGe[:ALL]

Sets the voltage range for all elements at once.

Syntax [:INPut] :VOLTagE:RANGe[:ALL] {<Voltage>}
 • When the crest factor setting is 3
 HV: <Voltage> = 6, 15, 30, 60, 150, 300, 600,
 1000, or 2000 (V)
 WB: <Voltage> = 6, 15, 30, 60, 150, 300, 600,
 or 1000 (V)
 • When the crest factor setting is 6 or 6A
 HV: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300,
 500, or 1000 (V)
 WB: <Voltage> = 3, 7.5, 15, 30, 75, 150, 300,
 or 500 (V)

Example :INPUT:VOLTAGE:RANGE:ALL 1000V

[:INPut] :VOLTagE:RANGe:ELEMeNt<x>

Sets or queries the voltage range for each element.

Syntax [:INPut] :VOLTagE:RANGe:
 ELEMeNt<x> {<Voltage>}
 [:INPut] :VOLTagE:RANGe:ELEMeNt<x>?
 <x> = 1 to 4 (element)
 <Voltage> = see
 "[:INPut] :VOLTagE:RANGe[:ALL]."

Example :INPUT:VOLTAGE:RANGE:ELEMENT1 1000V
 :INPUT:VOLTAGE:RANGE:ELEMENT1?
 -> :INPUT:VOLTAGE:RANGE:
 ELEMENT1 1.000E+03

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[:INPut] :VOLTagE:RANGe:{SIGMA|SIGMB}

Sets the voltage range at once for the elements belonging to wiring unit {ΣA|ΣB}.

Syntax [:INPut] :VOLTagE:RANGe:{SIGMA|
 SIGMB} {<Vol>}
 <Voltage> = see
 "[:INPut] :VOLTagE:RANGe[:ALL]."

Example :INPUT:VOLTAGE:RANGE:SIGMA 1000V

Description • When wiring unit {ΣA|ΣB} does not exist because of the wiring system setting ([:INPut]:WIRing), {SIGMA|SIGMB} is invalid.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

[[:INPut]:WIRing

Sets or queries the wiring system.

Syntax [:INPut]:WIRing {(P1W2|P1W3|P3W3|P3W4|
V3A3|V3A3R|V3A3 _ P3W4|V3A3R _ P3W4|
P3W4 _ V3A3S)[,(P1W2|P1W3|P3W3|P3W4|
V3A3|V3A3R|V3A3 _ P3W4|V3A3R _ P3W4|
P3W4 _ V3A3S)][(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 [3P4W]
V3A3 = three-phase three-wire system (three-
voltage three-current method) [3P3W(3V3A)]
V3A3R = three-phase three-wire system (three-
voltage three-current method) [3P3W(3V3AR)]
V3A3_P3W4 = conversion from three-phase
three-wire system (three-voltage three-current
method) to three-phase four-wire system
[3P3W(3V3A)→3P4W]
V3A3R_P3W4 = conversion from three-phase
three-wire system (three-voltage three-current
method) to three-phase four-wire system
[3P3W(3V3AR)→3P4W]
P3W4_V3A3S = conversion from three-phase
four-wire system to three-phase three-wire
system (three-voltage, three-current method)
[3P4W→3P3W(3V3AS)]

Example • Examples for a 4-element model
:INPUT:WIRING P1W2,P1W2,P1W2,P1W2
:INPUT:WIRING?
-> :INPUT:WIRING P1W2,P1W2,P1W2,P1W2
:INPUT:WIRING P1W3,P1W3
:INPUT:WIRING?
-> :INPUT:WIRING P1W3,P1W3
:INPUT:WIRING P1W2,V3A3
:INPUT:WIRING?
-> :INPUT:WIRING P1W2,V3A3
• Examples 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 wiring system patterns in order from the
lowest-numbered element.
• Some wiring system patterns cannot be
selected depending on the installed input
element configuration. For wiring system
patterns, see the User's Manual (IM
WT1501-01EN).
• For multi-unit synchronization (option
/SY1), specify the target unit with
:SYSTem:MUSYnc:SUNit.

5.13 INTEGrate Group

The INTEGrate group is for integration.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output > Integration** and query the settings.

:INTEGrate?

Queries all integration settings.

Syntax :INTEGrate?

:INTEGrate:ACAL

Sets or queries the integration auto calibration on/off status.

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 integration by element.

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|PULSe}
:INTEGrate:MODE?
NORMAL = normal integration mode
CONTInuous = continuous integration mode
RNORMAL = real-time normal integration mode
RCONTInuous = real-time continuous integration
mode
PULSe = pulse integration mode

Example :INTEGRATE:MODE NORMAL
:INTEGRATE:MODE?
-> :INTEGRATE:MODE NORMAL

:INTEGrate:QMODE?

Queries the current mode for current integration for all elements.

Syntax :INTEGrate:QMODE?

:INTEGrate:QMODE[:ALL]

Sets the current mode for current integration for all elements at once.

Syntax :INTEGrate:QMODE[:ALL] {RMS|MEAN|DC|
RMEAN|AC}

Example :INTEGRATE:QMODE:ALL DC

:INTEGrate:QMODE:ELEMent<x>

Sets or queries the current mode for current integration for each element.

Syntax :INTEGrate:QMODE:ELEMent<x> {RMS|
MEAN|DC|RMEAN|AC}
:INTEGrate:QMODE:ELEMent<x>?
<x> = 1 to 4 (element)

Example :INTEGRATE:QMODE:ELEMENT1 DC
:INTEGRATE:QMODE:ELEMENT1?
-> :INTEGRATE:QMODE:ELEMENT1 DC

Description This command operates according to the current mode setting for each element, regardless of the on/off status of independent integration by element (:INTEGrate:INDEpendent).

:INTEGrate:RACTION

Sets or queries the integration restart operation when power is restored during integration.

Syntax :INTEGrate:RACTION {START|STOP|ERRor}
:INTEGrate:RACTION?

START = automatically restarts integration.

STOP = enters the integration stop state.

Integration is restarted by manual operation (START key or command).

ERRor = enters the integration error state.

Restarting integration is prohibited.

Example :INTEGRATE:RACTION STOP
:INTEGRATE:RACTION?
-> :INTEGRATE:RACTION STOP

:INTEGrate:RESet

Resets integration values.

Syntax :INTEGrate:RESet {[<NRf>][,<NRf>]
[,<NRf>][,<NRf>]}

<x> = 1 to 4 (element)

Example :INTEGRATE:RESET (executed for all elements)
:INTEGRATE:RESET 1,2,3 (executed by specifying elements)

Description • When independent integration by element (:INTEGrate:INDePendent) is ON (1), parameters (elements to execute) can be specified. If parameters are omitted, all elements are specified.
• When independent integration by element (:INTEGrate:INDePendent) is OFF (0), parameters cannot be specified.

:INTEGrate:RTAll:{START|END}

Sets the reserved integration {start|end} time for all elements at once in 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 2026,1,1,0,0,0

:INTEGrate:RTIME<x>?

Queries the scheduled integration start/end times in real-time integration mode.

Syntax :INTEGrate:RTIME<x>?
<x> = 1 to 4 (element)

:INTEGrate:RTIME<x>:{START|END}

Sets or queries the scheduled integration {start|end} time in real-time integration mode.

Syntax :INTEGrate:RTIME<x>:{START|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
:INTEGrate:RTIME<x>:{START|END}?
<x> = 1 to 4 (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 2026,1,1,0,0,0
:INTEGRATE:RTIME1:START?
-> :INTEGRATE:RTIME1:START 2026,1,1,0,0,0

Description • This setting is valid when the integration mode (:INTEGrate:MODE) is real-time integration mode ({RNORmal|RCONtinuous}).
• When independent integration by element (:INTEGrate:INDePendent) is OFF (0), <x> can be omitted (<x> = 1). All elements operate according to the scheduled time for integration {start|end} setting for element 1.

:INTEGrate:START

Starts integration.

Syntax :INTEGrate:START {[<NRf>][,<NRf>]
[,<NRf>][,<NRf>]}
<x> = 1 to 4 (element)

Example :INTEGRATE:START (executed for all elements)
:INTEGRATE:START 1,2,3 (executed by specifying elements)

Description • When independent integration by element (:INTEGrate:INDePendent) is ON (1), parameters (elements to execute) can be specified. If parameters are omitted, all elements are specified.
• When independent integration by element (:INTEGrate:INDePendent) is OFF (0), parameters cannot be specified.

5.13 INTEGrate Group

:INTEGrate:STATE?

Queries the integration status.

Syntax :INTEGrate:STATE? {<NRf>}
:INTEGrate:STATE?
<x> = 1 to 4 (element)

Example

- When independent integration by element (:INTEGrate:INDEpendent) is OFF (0)
:INTEGrate:STATE? -> RESET
- When independent integration by element (:INTEGrate:INDEpendent) is ON (1)
:INTEGrate:STATE? 1 -> RESET
:INTEGrate:STATE?
-> RESET,RESET,RESET,RESET

Description

- The response contents are as follows.
RESet = integration reset
READy = waiting (real-time integration mode)
STARt = integration in progress
STOP = integration stopped
ERRor = integration abnormal termination
(integration value overflow, power failure)
TIMEup = integration stopped by integration timer time
- When independent integration by element (:INTEGrate:INDEpendent) is OFF (0), the parameter (element to query) cannot be specified.
- When independent integration by element (:INTEGrate:INDEpendent) is ON (1), the parameter can be specified. If the parameter is omitted, this command outputs the integration status for all elements in order from element 1.

:INTEGrate:STOP

Stops integration.

Syntax :INTEGrate:STOP {[<NRf>][,<NRf>]
[,<NRf>][,<NRf>]]
<x> = 1 to 4 (element)

Example :INTEGrate:STOP (executed for all elements)
:INTEGrate:STOP 1,2,3 (executed by specifying elements)

Description

- When independent integration by element (:INTEGrate:INDEpendent) is ON (1), parameters (elements to execute) can be specified. If parameters are omitted, all elements are specified.
- When independent integration by element (:INTEGrate:INDEpendent) is OFF (0), parameters cannot be specified.

:INTEGrate:TIMer<x>

Sets or queries the integration timer duration.

Syntax :INTEGrate:TIMer<x> {<NRf>,<NRf>,<NRf>}
:INTEGrate:TIMer<x>?
<x> = 1 to 4 (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 integration by element (:INTEGrate:INDEpendent) is OFF (0), <x> can be omitted (<x> = 1). All elements operate according to the integration timer time setting for element 1.

:INTEGrate:TMALl

Sets the integration timer duration for all elements at once.

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 computation method by polarity (WP+/WP-) for all elements.

Syntax :INTEGrate:WPTYPE?

:INTEGrate:WPTYPE[:ALL]

Sets the watt-hour computation method by polarity (WP+/WP-) for all elements at once.

Syntax :INTEGrate:WPTYPE[:ALL] {CHARge|SOLD}
Example :INTEGrate:WPTYPE:ALL CHARGE

:INTEGrate:WPTYPE:ELEMent<x>

Sets or queries the watt-hour computation method by polarity (WP+/WP-) for each element.

Syntax :INTEGrate:WPTYPE:
ELEMent<x> {CHARge|SOLD}
:INTEGrate:WPTYPE:ELEMent<x>?
<x> = 1 to 4 (element)
CHARge = charge/discharge
SOLD = import/export power

Example :INTEGrate:WPTYPE:ELEMENT1 CHARGE
:INTEGrate:WPTYPE:ELEMENT1?
-> :INTEGrate:WPTYPE:ELEMENT1 CHARGE

Description This command operates according to the computation method setting for each element, regardless of the on/off status of independent integration by element (:INTEGrate:INDEpendent).

5.14 MEASure Group

The MEASure group is for computation.

You can configure the same settings as on the screens that appear when you select the **Setup** menu icon  > **Computation/Output** > **Efficiency**, **ΔMeasure**, **Update Rate/Averaging**, or **Measure**, and query the settings.

: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
(decay 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 Averaging for harmonic measurement functions is valid only when TYPE = EXPonent (decay constant). For details, see the User's Manual (IM WT1501-01EN).

:MEASure:AVERaging[:STATE]

Sets or queries the averaging on/off status.

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 For averaging of harmonic measurement functions, only EXPonent is valid. For details, see the User's Manual (IM WT1501-01EN).

:MEASure:EFFiciency?

Queries all efficiency computation settings.

Syntax :MEASure:EFFiciency?

:MEASure:EFFiciency:ETA<x>[:FUNCTION]

Sets or queries the efficiency formula.

Syntax :MEASure:EFFiciency:ETA<x>:
FUNCTION {(OFF|P<x>|PA|PB|PM<x>|
UDEF<x>)[,(OFF|P<x>|PA|PB|PM<x>|
UDEF<x>)]}
:MEASure:EFFiciency:ETA<x>:FUNCTION?
<x> of ETA<x> = 1 to 4 (η1 to η4)
OFF = no computation
<x> of P<x> = 1 to 4 (element)
PA, PB = PΣA, PΣB
(available depending on the number of
elements)
<x> of PM<x> = 1 to 4 (motor output, Pm1 to
Pm4)
<x> of UDEF<x> = 1 to 2 (Udef1 to Udef2)

Example :MEASURE:EFFICIENCY:ETA1:
FUNCTION P3,PA
:MEASURE:EFFICIENCY:ETA1:FUNCTION?
-> :MEASURE:EFFICIENCY:ETA1:
FUNCTION P3,PA

Description • Specify the numerator and then the denominator.
• The denominator can be omitted. If omitted, the denominator is set to MAIN.
• In the query response, the denominator is omitted when it is OFF.
• PM<x> is valid only on models with motor evaluation function 1 for encoders (option /EM1), motor evaluation function 1 for resolvers (option /RM1), motor evaluation function 2 for encoders (option /EM2), motor evaluation function 2 for resolvers (option /RM2), or motor evaluation function for encoders/resolvers (option EM1RM1).

5.14 MEASure Group

:MEASure:EFFiciency:ETA<x>:UNIT

Sets or queries the units used in the efficiency formula.

Syntax :MEASure:EFFiciency:ETA<x>:
UNIT {(MAIN|SUB1|SUB2|SUB3|SUB4)
[, (MAIN|SUB1|SUB2|SUB3|SUB4)]}
:MEASure:EFFiciency:ETA<x>:UNIT?
<x> of ETA<x> = 1 to 4 (η_1 to η_4)

Example :MEASURE:EFFICIENCY:ETA1:
UNIT MAIN,SUB1
:MEASURE:EFFICIENCY:ETA1:UNIT?
-> :MEASURE:EFFICIENCY:ETA1:
UNIT MAIN,SUB1

Description

- Specify the numerator and then the denominator.
- The denominator can be omitted. If omitted, the denominator is set to MAIN.
- In the query response, the denominator is omitted when it is MAIN.
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:MEASure:EFFiciency:UDEF<x>[:FUNcti on]

Sets or queries the user-defined parameters used in the efficiency formula.

Syntax MEASure:EFFiciency:UDEF<x>:
FUNCTION {(NONE|P<x>|PA|PB|PM<x>)
[, (NONE|P<x>|PA|PB|PM<x>)]
[, (NONE|P<x>|PA|PB|PM<x>)]
[, (NONE|P<x>|PA|PB|PM<x>)]}
:MEASure:EFFiciency:UDEF<x>:FUNCTION?
<x> of UDEF<x> = 1 to 2 (Udef1 to Udef2)
NONE = no operand
<x> of P<x> = 1 to 4 (element)
PA, PB = $P_{\Sigma A}$, $P_{\Sigma B}$
(available depending on the number of elements)
<x> of PM<x> = 1 to 4 (motor output, Pm1 to Pm4)

Example :MEASURE:EFFICIENCY:UDEF1:
FUNCTION P1,P2,P3
:MEASURE:EFFICIENCY:UDEF1:FUNCTION?
-> :MEASURE:EFFICIENCY:UDEF1:
FUNCTION P1,P2,P3

Description

- Specify the first through fourth terms in order.
- The second to fourth terms can be omitted. Omitted terms are set to NONE.
- In the query response, the second to fourth terms are omitted when that term and all subsequent terms are NONE.
- PM<x> is valid only on models with motor evaluation function 1 for encoders (option /EM1), motor evaluation function 1 for resolvers (option /RM1), motor evaluation function 2 for encoders (option /EM2), motor evaluation function 2 for resolvers (option /RM2), or motor evaluation function for encoders/resolvers (option EM1RM1).

:MEASure:EFFiciency:UDEF<x>:UNIT

Sets or queries the units for the user-defined parameters used in the efficiency formula.

Syntax :MEASure:EFFiciency:UDEF<x>:
UNIT {(MAIN|SUB1|SUB2|SUB3|SUB4)
[, (MAIN|SUB1|SUB2|SUB3|SUB4)]
[, (MAIN|SUB1|SUB2|SUB3|SUB4)]
[, (MAIN|SUB1|SUB2|SUB3|SUB4)]}
:MEASure:EFFiciency:UDEF<x>:UNIT?

Example :MEASURE:EFFICIENCY:UDEF1:UNIT
MAIN, SUB1, SUB2
:MEASURE:EFFICIENCY:UDEF1:UNIT?
-> :MEASURE:EFFICIENCY:UDEF1:UNIT
MAIN, SUB1, SUB2

Description

- Specify the first through fourth terms in order.
- The second to fourth terms can be omitted. Omitted terms are set to MAIN.
- In the query response, the second to fourth terms are omitted when that term and all subsequent terms are MAIN.
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:MEASure:EVENT<x>?

Queries all user-defined event settings.

Syntax :MEASure:EVENT<x>?
<x> = 1 to 8 (Event1 to Event8)

:MEASure:EVENT<x>:EXPRession?

Queries all settings for the user-defined event conditional expression.

Syntax :MEASure:EVENT<x>:EXPRession?
<x> = 1 to 8 (Event1 to Event8)

:MEASure:EVENT<x>:EXPRession:CONDi tion

Sets or queries the user-defined event conditional expression (condition-combination 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:EVENT3:EXPRESSION:
CONDITION 1, AND, 2
:MEASURE:EVENT3:EXPRESSION:
CONDITION?
-> :MEASURE:EVENT3:EXPRESSION:
CONDITION 1, AND, 2

Description This setting is valid when the conditional expression type (:MEASure:EVENT<x>:EXPRession:TYPE) is CONDITION.

:MEASure:EVENT<x>:EXPRession:INVer se

Sets or queries the user-defined event conditional 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 setting is valid when the conditional expression type (:MEASure:EVENT<x>:EXPRession:TYPE) is CONDITION.

5.14 MEASure Group

:MEASure:EVENT<x>:EXPRession:ITEM[:FUNCTION]

Sets or queries the item used in the user-defined event conditional expression (range type).

Syntax :MEASure:EVENT<x>:EXPRession:ITEM:FUNCTION {<Function>[,<Element>][,<Order>]}

:MEASure:EVENT<x>:EXPRession:ITEM:FUNCTION?

<x> = 1 to 8 (Event1 to Event8)

<Function> = {URMS|IRMS|P|S|Q|...}

<Element> = {<NRf>|SIGMa|SIGMB}

(<NRf> = 1 to 4)

<Order> = {TOTa|DC|<NRf>}

(<NRf> = 1 to 100)

Example :MEASURE:EVENT1:EXPRESSION:ITEM:FUNCTION URMS,1

:MEASURE:EVENT1:EXPRESSION:ITEM:FUNCTION?

-> :MEASURE:EVENT1:EXPRESSION:ITEM:FUNCTION URMS,1

Description

- This setting is valid when the conditional expression type (:MEASure:EVENT<x>:EXPRession:TYPE) is RANGE.
- For the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTa is set.
- In responses for functions that do not require <Element> or <Order>, <Element> or <Order> is omitted.
- For motor functions (<Function> = {SPEed|TORQue|SYNCsp|SLIP|PM}), the <Element> part is <Motor>. <Order> is not required.
 - <Motor> = 1 to 4 (motor number; motor 1 if omitted)

:MEASure:EVENT<x>:EXPRession:ITEM:UNIT

Sets or queries the unit for the item used in the user-defined event conditional expression (range type).

Syntax :MEASure:EVENT<x>:EXPRession:ITEM:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}

:MEASure:EVENT<x>:EXPRession:ITEM:UNIT?

<x> = 1 to 8 (Event1 to Event8)

Example :MEASURE:EVENT1:EXPRESSION:ITEM:UNIT MAIN

:MEASURE:EVENT1:EXPRESSION:ITEM:UNIT?

-> :MEASURE:EVENT1:EXPRESSION:ITEM:UNIT MAIN

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:MEASure:EVENT<x>:EXPRession:LIMit<x>

Sets or queries the range limit for the user-defined event conditional expression (range type).

Syntax :MEASure:EVENT<x>:EXPRession:LIMit<x> {<Operand>,<NRf>}

:MEASure:EVENT<x>:EXPRession:LIMit<x>?

<x> of EVENT<x> = 1 to 8 (Event1 to Event8)

<x> of LIMit<x> = 1 to 2

<Operand> = {OFF|LESS|LEQual|EQual|GReat|GEQual|NEQual}

<NRf> = -1.0000E+12 to 1.0000E+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:LIMIT2 OFF

Description

- This setting is valid when the conditional expression type (:MEASure:EVENT<x>:EXPRession:TYPE) is RANGE.
- When <Operand> = OFF, omit <NRf>.

**:MEASure:EVENT<x>:EXPRession:STRi
ng?**

Queries the user-defined event conditional expression in string format.

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 conditional expression, this command returns "No Expression."

:MEASure:EVENT<x>:EXPRession:TYPE

Sets or queries the user-defined event conditional expression type.

Syntax :MEASure:EVENT<x>:EXPRession:
TYPE {RANGE|CONDition}
:MEASure:EVENT<x>:EXPRession:TYPE?
<x> = 1 to 8 (Event1 to Event8)
RANGe = range 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 display data string used when the user-defined event condition is not met (False).

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 user-defined event 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 whether the user-defined event is enabled (ON) or disabled (OFF).

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 display data string used when the user-defined event condition is met (True).

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 user-defined function settings.

Syntax :MEASure:FUNCTION<x>?
<x> = 1 to 20 (F1 to F20)

:MEASure:FUNCTION<x>:EXPRession

Sets or queries the user-defined function expression.

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 user-defined function name.

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"

5.14 MEASure Group

:MEASure:FUNCTION<x>:PRESet[:EXECute]

Executes the user-defined function preset.

Syntax :MEASure:FUNCTION<x>:
PRESet[:EXECute] {DQP3W4|DQV3A3|
REL _ DQP3W4|REL _ DQV3A3}
<x> = 1 to 20 (F1 to F20)
* The input value is ignored, and the operation
applies to all functions.

Example :MEASURE:FUNCTION1:PRESET:
EXECUTE DQV3A3

Description • Executes the user-defined function preset.
• The <x> in FUNCTION<x> does not affect 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, and the operation
applies to all functions.
<String> = File name

Example :MEASURE:FUNCTION1:PRESET:FILE:
LOAD "USERFUNC1"

Description • Loads a user-defined function file.
• The <x> in FUNCTION<x> does not affect the
setting.

:MEASure:FUNCTION<x>:PRESet:FILE:SAVE

Saves the user-defined function contents to a file.

Syntax :MEASure:FUNCTION<x>:PRESet:FILE:
SAVE {<String>}
<x> = 1 to 20 (F1 to F20)
* The input value is ignored, and the operation
applies to all functions.
<String> = File name

Example :MEASURE:FUNCTION1:PRESET:FILE:SAVE
"USERFUNC1"

Description • Saves the user-defined function file.
• The <x> in FUNCTION<x> does not affect the
setting.

:MEASure:FUNCTION<x>[:STATE]

Sets or queries whether the user-defined function is enabled (ON) or disabled (OFF).

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 appended to the user-defined function computation result.

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 does not affect the calculation
result.

:MEASure:MHOLD

Sets or queries whether the MAX HOLD function used in user-defined functions is enabled (ON) or disabled (OFF).

Syntax :MEASure:MHOLD {<Boolean>}
:MEASure:MHOLD?

Example :MEASURE:MHOLD ON
:MEASURE:MHOLD? -> :MEASURE:MHOLD 1

Description • When the MAX HOLD function is specified in
a user-defined function, MAX HOLD operation
starts when :MEASure:MHOLD is set to ON.
• Setting :MEASure:MHOLD to OFF ends MAX
HOLD operation, and the MAX HOLD value
becomes "No data."
• If ON is set while :MEASure:MHOLD is already
ON, the MAX HOLD value is reset once, and
MAX HOLD operation starts again.
• For details on how to specify the MAX
HOLD function, see the User's Manual (IM
WT1501-01EN).

:MEASure:PC?

Queries all settings related to Pc (Corrected Power) computation.

Syntax :MEASure:PC?

:MEASure:PC:FORMula

Sets or queries the Pc (Corrected Power) formula.

Syntax :MEASure:PC:FORMula {TYPE1|TYPE2}
:MEASure:PC:FORMula?

Example :MEASURE:PC:FORMULA TYPE1
:MEASURE:PC:FORMULA?
-> :MEASURE:PC:FORMULA TYPE1

Description For the formulas corresponding to {TYPE1|TYPE2}, see the User's Manual (IM WT1501-01EN).

:MEASure:PC:P<x>

Sets or queries a parameter for Pc (Corrected Power) computation.

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 This parameter is used when :MEASure:PC:FORMula is set to TYPE1.

:MEASure:PHASe

Sets or queries the phase difference display format.

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 difference is displayed as ± 0 to 180° (Lead/Lag). When 360 is selected, it is displayed as 0 to 360° .

:MEASure:PHASe:POLarity

Sets or queries the phase difference polarity.

Syntax :MEASure:PHASe:POLarity {TYPE1|TYPE2}
:MEASure:PHASe:POLarity?

Example :MEASURE:PHASE:POLARITY TYPE1
:MEASURE:PHASE:POLARITY?
-> :MEASURE:PHASE:POLARITY TYPE1

Description The polarity settings corresponding to the choices are as follows.

Type 1: Lead(-)/Lag(+) (*default setting)

Type 2: Lead(+)/Lag(-)

:MEASure:SFORMula

Sets or queries the formula for S (apparent power).

Syntax :MEASure:SFORMula {RMS|MEAN|DC|MRMS|
RMEAN}
:MEASure:SFORMula?

Example :MEASURE:SFORMULA RMS
:MEASURE:SFORMULA?
-> :MEASURE:SFORMULA RMS

Description The formulas corresponding to the choices are as follows.

RMS: $S = U_{rms} \cdot I_{rms}$

MEAN: $S = U_{mean} \cdot I_{mean}$

DC: $S = U_{dc} \cdot I_{dc}$

MRMS: $S = U_{mean} \cdot I_{rms}$

RMEAN: $S = U_{rmean} \cdot I_{rmean}$

:MEASure:SQFormula

Sets or queries the formulas for 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 the formulas corresponding to {TYPE1|TYPE2|TYPE3}, see the User's Manual (IM WT1501-01EN).

:MEASure:SYNChronize

Sets or queries the synchronous measurement mode.

Syntax :MEASure:SYNChronize {MASTER|SLAVE}
:MEASure:SYNChronize?

Example :MEASURE:SYNCHRONIZE MASTER
:MEASURE:SYNCHRONIZE?
-> :MEASURE:SYNCHRONIZE MASTER

:MEASure:UFFormula

Sets or queries the unbalance factor formula.

Syntax :MEASure:UFFormula {FUNDamental|RMS}
:MEASure:UFFormula?

Example :MEASURE:UFFORMULA FUNDAMENTAL
:MEASURE:UFFORMULA?
-> :MEASURE:UFFORMULA FUNDAMENTAL

5.15 MOTO Group

The MOTO group is for motor evaluation functions.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Input (Advanced/Options) > Motor/Aux** and query the settings.

- The commands in this group are valid when motor evaluation function 1 for encoders (option /EM1), motor evaluation function 1 for resolvers (option /RM1), or the motor evaluation function for encoders/resolvers (option /EM1RM1) is installed.
- MOTO3 and MOTO4 are valid on models with motor evaluation function 2 for encoders (option /EM2), motor evaluation function 2 for resolvers (option /RM2), or motor evaluation function for encoders/resolvers (option /EM1RM1).

:MOTOR<x>?

Queries all settings for the motor evaluation function.

Syntax :MOTOR<x>?
<x> = 1 to 4 (motor number)

:MOTOR<x>:EANGLE?

Queries all settings for electrical angle measurement.

Syntax :MOTOR<x>:EANGLE?
<x> = 1, 3 (motor number)

Description Electrical angle measurement applies to Motor1 and Motor3.

:MOTOR<x>:EANGLE:CORRection?

Queries all settings related to the electrical angle correction value setting.

Syntax :MOTOR<x>:EANGLE:CORRection?
<x> = 1, 3 (motor number)

:MOTOR<x>:EANGLE:CORRection:AENTER?

Queries all settings related to automatic entry of the electrical angle correction value.

Syntax :MOTOR<x>:EANGLE:CORRection:AENTER?
<x> = 1, 3 (motor number)

:MOTOR<x>:EANGLE:CORRection:AENTER[:EXECute]

Executes automatic entry of the electrical angle correction value.

Syntax :MOTOR<x>:EANGLE:CORRection:
AENTER[:EXECute]
<x> = 1, 3 (motor number)

Example :MOTOR1:EANGLE:CORRection:AENTER:
EXECUTE

Description • The electrical angle measurement value of the target source (:MOTOR<x>:EANGLE:CORRection:AENTER:TARGET) is entered as the correction value.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:EANGLE:CORRection:AENTER:TARGET

Sets or queries the target source for automatic entry of the electrical angle correction value.

Syntax :MOTOR<x>:EANGLE:CORRection:AENTER:
TARGET {U<x>|I<x>}
:MOTOR<x>:EANGLE:CORRection:AENTER:
TARGET?
<x> of MOTOR<x> = 1, 3 (motor number)
<x> of U<x> and I<x> = 1 to 4 (element)

Example :MOTOR1:EANGLE:CORRection:AENTER:
TARGET U1
:MOTOR1:EANGLE:CORRection:AENTER:
TARGET?
-> :MOTOR1:EANGLE:CORRection:AENTER:
TARGET U1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:EANGLE:CORRection:CLEAr

Clears the electrical angle correction value.

Syntax :MOTOR<x>:EANGLE:CORRection:CLEAr
<x> = 1, 3 (motor number)

Example :MOTOR1:EANGLE:CORRection:CLEAr

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:EANGLE:CORRection[:VALue]

Sets or queries the electrical angle correction value.

Syntax :MOTOR<x>:EANGLE:CORRection[:
VALue] {<NRf>}
:MOTOR<x>:EANGLE:CORRection:VALue?
<x> = 1, 3 (motor number)
<NRf> = -180.00 to 180.00

Example :MOTOR1:EANGLE:CORRection:VALue 0
:MOTOR1:EANGLE:CORRection:VALue?
-> :MOTOR1:EANGLE:CORRection:
VALue 0.00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:EANGLE:HTRigger

Sets or queries the trigger for electrical angle measurement.

Syntax :MOTor<x>:EANGLE:HTRigger {(NONE|Z1|Z3|CZPHase),(NONE|Z1|Z3|CZPHase)}
:MOTor<x>:EANGLE:HTRigger?
<x> = 1, 3 (motor number)
Z1 = Z Phase1 (ChD)
(/EM1, /RM1, or /EM1RM1 option required)
Z3 = Z Phase3 (ChH)
(/EM2, /RM2, or /EM1RM1 option required)
CZPHase = common Z Phase (valid only during multi-unit synchronous operation (option /SY1))

Example :MOTOR1:EANGLE:HTRIGGER Z1,Z1
:MOTOR1:EANGLE:HTRIGGER?
-> :MOTOR1:EANGLE:HTRIGGER Z1,Z1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:EANGLE[:STATE]

Sets or queries the electrical angle measurement on/off status.

Syntax :MOTor<x>:EANGLE[:STATE] {<Boolean>}
:MOTor<x>:EANGLE:STATE?
<x> = 1, 3 (motor number)

Example :MOTOR1:EANGLE:STATE ON
:MOTOR1:EANGLE:STATE?
-> :MOTOR1:EANGLE:STATE 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:EANGLE:TRIGger:CZPHase:SOURce

Sets or queries the common Z Phase source for multi-unit synchronization.

Syntax :MOTor<x>:EANGLE:TRIGger:CZPHase:SOURce {Z1|Z3}
:MOTor<x>:EANGLE:TRIGger:CZPHase:SOURce?
<x> = 1 to 4 (motor number)
Z1 = Z Phase1 (ChD)
(/EM1, /RM1, or /EM1RM1 option required)
Z3 = Z Phase3 (ChH)
(/EM2, /RM2, or /EM1RM1 option required)

Example :MOTOR1:EANGLE:TRIGGER:CZPHASE:SOURCE MAIN
:MOTOR1:EANGLE:TRIGGER:CZPHASE:SOURCE?
-> :MOTOR1:EANGLE:TRIGGER:CZPHASE:SOURCE MAIN

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• In multi-unit synchronous operation, specify the unit with :SYSTEM:MUSYnc:SUNit.
• The <x> in MOTor<x> does not affect setting/querying.

:MOTor<x>:EANGLE:TRIGger:CZPHase:UNIT

Sets or queries the common Z Phase unit for multi-unit synchronization.

Syntax :MOTor<x>:EANGLE:TRIGger:CZPHase:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}
:MOTor<x>:EANGLE:TRIGger:CZPHase:UNIT?
<x> = 1, 3 (motor number)

Example :MOTOR1:EANGLE:TRIGGER:CZPHASE:UNIT MAIN
:MOTOR1:EANGLE:TRIGGER:CZPHASE:UNIT?
-> :MOTOR1:EANGLE:TRIGGER:CZPHASE:UNIT MAIN

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• In multi-unit synchronous operation, specify the unit with :SYSTEM:MUSYnc:SUNit.
• The <x> in MOTor<x> does not affect setting/querying.

:MOTor<x>:FILTER?

Queries all input filter settings.

Syntax :MOTor<x>:FILTER?
<x> = 1 to 4 (motor number)

:MOTor<x>:FILTER[:LINE]

Sets or queries the line filter.

Syntax :MOTor<x>:FILTER[:LINE] {OFF|<Frequency>}
:MOTor<x>:FILTER:LINE?
<x> = 1 to 4 (motor number)
OFF = line filter off
<Frequency> = 100, 500 (Hz), 1 (kHz)
(line filter ON, cutoff frequency)

Example :MOTOR1:FILTER:LINE OFF
:MOTOR1:FILTER:LINE?
-> :MOTOR1:FILTER:LINE OFF

Description • This setting is common to rotation signal input and torque signal input.
This setting is valid when the signal input type (:MOTor<x>{SPEed|TORQue}:TYPE) is ANALog (analog input).
• In multi-unit synchronous operation, specify the unit with :SYSTEM:MUSYnc:SUNit.

5.15 MOTor Group

:MOTor<x>:FILTer:NOISe:SPEEd

Sets or queries the noise filter for the rotation signal (pulse input method).

Syntax :MOTor<x>:FILTer:NOISe:SPEEd {OFF|
<Frequency>}
:MOTor<x>:FILTer:NOISe:SPEEd?
<x> = 1 to 4 (motor number)
OFF = Noise filter off
<Frequency> = 10, 100 (kHz), 1 (MHz)
(noise filter ON, cutoff frequency)

Example :MOTOR1:FILTER:NOISE:SPEED OFF
:MOTOR1:FILTER:NOISE:SPEED?
-> :MOTOR1:FILTER:NOISE:SPEED OFF

Description • This setting is valid when the rotation signal input type (:MOTor<x>:SPEEd:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:FILTer:NOISe:TORQue

Sets or queries the noise filter for the torque signal (pulse input method).

Syntax :MOTor<x>:FILTer:NOISe:TORQue {OFF|
<Frequency>}
:MOTor<x>:FILTer:NOISe:TORQue?
<x> = 1 to 4 (motor number)
OFF = Noise filter off
<Frequency> = 10, 100 (kHz), 1 (MHz)
(noise filter ON, cutoff frequency)

Example :MOTOR1:FILTER:NOISE:TORQUE OFF
:MOTOR1:FILTER:NOISE:TORQUE?
-> :MOTOR1:FILTER:NOISE:TORQUE OFF

Description • This setting is valid when the torque signal input type (:MOTor<x>:TORQue:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:PM?

Queries all settings related to motor output (Pm).

Syntax :MOTor<x>:PM?
<x> = 1 to 4 (motor number)

:MOTor<x>:PM:SCALing

Sets or queries the scaling coefficient for motor output computation.

Syntax :MOTor<x>:PM:SCALing {<NRf>}
:MOTor<x>:PM:SCALing?
<x> = 1 to 4 (motor number)
<NRf> = 0.0001 to 99999.9999

Example :MOTOR1:PM:SCALING 1
:MOTOR1:PM:SCALING?
-> :MOTOR1:PM:SCALING 1.0000

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:PM:UNIT

Sets or queries the unit appended to the motor output computation result.

Syntax :MOTor<x>:PM:UNIT {<String>}
:MOTor<x>:PM:UNIT?
<x> = 1 to 4 (motor number)
<String> = Up to 8 characters

Example :MOTOR1:PM:UNIT "W"
:MOTOR1:PM:UNIT?
-> :MOTOR1:PM:UNIT "W"

Description • This command does not affect the calculation result.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:POLE

Sets or queries the number of motor poles.

Syntax :MOTor<x>:POLE {<NRf>}
:MOTor<x>:POLE?
<x> = 1 to 4 (motor number)
<NRf> = 1 to 99

Example :MOTOR1:POLE 2
:MOTOR1:POLE? -> :MOTOR1:POLE 2

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:RESolver?

Queries all resolver calculation settings.

Syntax :MOTor<x>:RESolver?
<x> = 1 to 4 (motor number)

:MOTor<x>:RESolver:AUTO

Sets or queries the auto voltage range on/off status for resolver calculation.

Syntax :MOTor<x>:RESolver:AUTO {<Boolean>}
:MOTor<x>:RESolver:AUTO?
<x> = 1 to 4 (motor number)

Example :MOTOR1:RESOLVER:AUTO ON
:MOTOR1:RESOLVER:AUTO?
-> :MOTOR1:RESOLVER:AUTO 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:EAAVeraging

Sets or queries the electrical angle averaging on/off status for resolver calculation.

Syntax :MOTor<x>:RESolver:
EAAVeraging {<Boolean>}
:MOTor<x>:RESolver:EAAVeraging?
<x> = 1 to 4 (motor number)

Example :MOTOR1:RESOLVER:EAAVERAGING ON
:MOTOR1:RESOLVER:EAAVERAGING?
-> :MOTOR1:RESOLVER:EAAVERAGING 1

:MOTor<x>:RESolver:HYSTeresis

Sets or queries the carrier zero-crossing detection hysteresis for resolver calculation.

Syntax :MOTor<x>:RESolver:
HYSTeresis {HIGH|LOW|MIDDLE}
:MOTor<x>:RESolver:HYSTeresis?
<x> = 1 to 4 (motor number)

Example :MOTOR1:RESOLVER:HYSTERESIS LOW
:MOTOR1:RESOLVER:HYSTERESIS?
-> :MOTOR1:RESOLVER:HYSTERESIS LOW

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:LFIlter

Sets or queries the line filter for resolver calculation.

Syntax :MOTor<x>:RESolver:LFIlter {OFF|
<Frequency>}
:MOTor<x>:RESolver:LFIlter?
<x> = 1 to 4 (motor number)
<Frequency> = 10, 20, 50, 100, 200 (kHz)

Example :MOTOR1:RESOLVER:LFIILTER 100kHz
:MOTOR1:RESOLVER:LFIILTER?
-> :MOTOR1:RESOLVER:LFIILTER 100.0E+03

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:RPPair

Sets or queries the resolver pole pairs for resolver calculation.

Syntax :MOTor<x>:RESolver:RPPair {<NRf>}
:MOTor<x>:RESolver:RPPair?
<x> = 1 to 4 (motor number)
<NRf> = 1 to 15

Example :MOTOR1:RESOLVER:RPPAIR 1
:MOTOR1:RESOLVER:RPPAIR?
-> :MOTOR1:RESOLVER:RPPAIR 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:PRANge

Sets or queries the range for resolver calculation.

Syntax :MOTor<x>:RESolver:
PRANge {<NRf>,<NRf>}
:MOTor<x>:RESolver:PRANge?
<x> = 1 to 4 (motor number)
<NRf> = -99999.9999 to 99999.9999

Example :MOTOR1:RESOLVER:PRANGE 10000,0
:MOTOR1:RESOLVER:PRANGE?
-> :MOTOR1:RESOLVER:PRANGE 10000.0000,0
.0000

Description • Specify the upper limit and then the lower limit.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:RANge

Queries all voltage range settings for resolver calculation.

Syntax :MOTor<x>:RESolver:RANge?
<x> = 1 to 4 (motor number)

:MOTor<x>:RESolver:RANge[:CARRier]

Sets or queries the voltage range of the excitation signal for resolver calculation.

Syntax :MOTor<x>:RESolver:RANge:
CARRier {<Voltage>}
:MOTor<x>:RESolver:RANge:CARRier?
<x> = 1 to 4 (motor number)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :MOTOR1:RESOLVER:RANGE:CARRIER 20V
:MOTOR1:RESOLVER:RANGE:CARRIER?
-> :MOTOR1:RESOLVER:RANGE:
CARRIER 20.0E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.15 MOTor Group

:MOTor<x>:RESolver:RANGe:COsine

Sets or queries the voltage range of the cosine signal for resolver calculation.

Syntax :MOTor<x>:RESolver:RANGe:
COSine {<Voltage>}
:MOTor<x>:RESolver:RANGe:COsine?
<x> = 1 to 4 (motor number)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :MOTOR1:RESOLVER:RANGE:COSINE 20V
:MOTOR1:RESOLVER:RANGE:COsine?
-> :MOTOR1:RESOLVER:RANGE:
COSINE 20.0E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:RANGe:LINKage

Sets or queries the on/off status of voltage range linkage for resolver calculation.

Syntax :MOTor<x>:RESolver:RANGe:
LINKage {<Boolean>}
:MOTor<x>:RESolver:RANGe:LINKage?
<x> = 1 to 4 (motor number)

Example :MOTOR1:RESOLVER:RANGE:LINKAGE ON
:MOTOR1:RESOLVER:RANGE:LINKAGE?
-> :MOTOR1:RESOLVER:RANGE:LINKAGE 1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:RANGe:SINE

Sets or queries the voltage range of the sine signal for resolver calculation.

Syntax :MOTor<x>:RESolver:RANGe:
SINE {<Voltage>}
:MOTor<x>:RESolver:RANGe:SINE?
<x> = 1 to 4 (motor number)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :MOTOR1:RESOLVER:RANGE:SINE 20V
:MOTOR1:RESOLVER:RANGE:SINE?
-> :MOTOR1:RESOLVER:RANGE:
SINE 20.0E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:SCALing

Sets or queries the scaling factor for resolver calculation.

Syntax :MOTor<x>:RESolver:SCALing {<Nrf>}
:MOTor<x>:RESolver:SCALing?
<x> = 1 to 4 (motor number)
<Nrf> = 0.0001 to 99999.9999

Example :MOTOR1:RESOLVER:SCALING 1
:MOTOR1:RESOLVER:SCALING?
-> :MOTOR1:RESOLVER:SCALING 1.0000

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:TFILter

Sets or queries the tracking filter for resolver calculation.

Syntax :MOTor<x>:RESolver:
TFILter {<Frequency>}
:MOTor<x>:RESolver:TFILter?
<x> = 1 to 4 (motor number)
<Frequency> = 100, 250, 1000, 2000 (Hz)

Example :MOTOR1:RESOLVER:TFILTER 100Hz
:MOTOR1:RESOLVER:TFILTER?
-> :MOTOR1:RESOLVER:TFILTER 100.0E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:RESolver:UNIT

Sets or queries the unit appended to resolver calculation results.

Syntax :MOTor<x>:RESolver:UNIT {<String>}
:MOTor<x>:RESolver:UNIT?
<x> = 1 to 4 (motor number)
<String> = Up to 8 characters

Example :MOTOR1:RESOLVER:UNIT "rpm"
:MOTOR1:RESOLVER:UNIT?
-> :MOTOR1:RESOLVER:UNIT "rpm"

Description • This command does not affect the calculation result.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:SPEEd?

Queries all rotation speed (Speed) settings.

Syntax :MOTOR<x>:SPEEd?
<x> = 1 to 4 (motor number)

:MOTOR<x>:SPEEd:AUTO

Sets or queries the auto voltage range on/off status for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:AUTO {<Boolean>}
:MOTOR<x>:SPEEd:AUTO?
<x> = 1 to 4 (motor number)

Example :MOTOR1:SPEED:AUTO ON
:MOTOR1:SPEED:AUTO?
-> :MOTOR1:SPEED:AUTO 1

Description • This setting is valid when the rotation signal input type (:MOTOR<x>:SPEEd:TYPE) is set to ANALog (analog input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:SPEEd:LSCale?

Queries all linear scaling settings for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:LSCale?
<x> = 1 to 4 (motor number)

Description The linear scaling settings apply when the rotation signal input type (:MOTOR<x>:SPEEd:TYPE) is set to ANALog (analog input).

:MOTOR<x>:SPEEd:LSCale:AVALue

Sets or queries slope A of the linear scale for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:LSCale:
AVALue {<Nrf>}
:MOTOR<x>:SPEEd:LSCale:AVALue?
<x> = 1 to 4 (motor number)
<Nrf> = 1.000E-03 to 1.000E+06

Example :MOTOR1:SPEED:LSCALE:AVALUE 1.000
:MOTOR1:SPEED:LSCALE:AVALUE?
-> :MOTOR1:SPEED:LSCALE:
AVALUE 1.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:SPEEd:LSCale:BVALue

Sets or queries offset value B of the linear scale for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:LSCale:
BVALue {<Nrf>}
:MOTOR<x>:SPEEd:LSCale:BVALue?
<x> = 1 to 4 (motor number)
<Nrf> = -1.000E+06 to 1.000E+06

Example :MOTOR1:SPEED:LSCALE:BVALUE 0
:MOTOR1:SPEED:LSCALE:BVALUE?
-> :MOTOR1:SPEED:LSCALE:
BVALUE 0.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:SPEEd:LSCale:CALCulate?

Queries all linear scale parameter calculation settings for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:LSCale:CALCulate?
<x> = 1 to 4 (motor number)

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

Sets or queries the {Point1X|Point1Y|Point2X|Point2Y} data for linear scale parameter calculation for the rotation signal (analog input type).

Syntax :MOTOR<x>:SPEEd:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y} {<Nrf>}
:MOTOR<x>:SPEEd:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y}?
<x> = 1 to 4 (motor number)

<Nrf> = -1.000E+12 to 1.000E+12
Example :MOTOR1:SPEED:LSCALE:CALCULATE:P1X 0
:MOTOR1:SPEED:LSCALE:CALCULATE:P1X?
-> :MOTOR1:SPEED:LSCALE:CALCULATE:
P1X 0.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

5.15 MoTor Group

:MoTor<x>:SPEed:LScale:CALCulate:EXECute

Executes linear scale parameter calculation for the rotation signal (analog input type).

Syntax :MoTor<x>:SPEed:LScale:CALCulate:
EXECute
<x> = 1 to 4 (motor number)

Example :MOTOR1:SPEED:LSCALE:CALCULATE:
EXECUTE

Description • This command calculates and sets linear scale slope A and offset value B from the preset data (Point1X, Point1Y, Point2X, Point2Y).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MoTor<x>:SPEed:PRANge

Sets or queries the range for the rotation signal (pulse input type).

Syntax :MoTor<x>:SPEed:PRANge {<NRf>,<NRf>}
:MoTor<x>:SPEed:PRANge?
<x> = 1 to 4 (motor number)
<NRf> = -99999.9999 to 99999.9999

Example :MOTOR1:SPEED:PRANGE 10000,0
:MOTOR1:SPEED:PRANGE?
-> :MOTOR1:SPEED:PRANGE 10000.0000,
0.0000

Description • Specify the upper limit and then the lower limit.
• This setting is valid when the rotation signal input type (:MoTor<x>:SPEed:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MoTor<x>:SPEed:PULSe

Sets or queries the pulse count for the rotation signal (pulse input type).

Syntax :MoTor<x>:SPEed:PULSe {<NRf>}
:MoTor<x>:SPEed:PULSe?
<x> = 1 to 4 (motor number)
<NRf> = 1 to 9999

Example :MOTOR1:SPEED:PULSE 60
:MOTOR1:SPEED:PULSE?
-> :MOTOR1:SPEED:PULSE 60

Description • This setting is valid when the rotation signal input type (:MoTor<x>:SPEed:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MoTor<x>:SPEed:RANge

Sets or queries the voltage range for the rotation signal (analog input type).

Syntax :MoTor<x>:SPEed:RANge {<Voltage>}
:MoTor<x>:SPEed:RANge?
<x> = 1 to 4 (motor number)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :MOTOR1:SPEED:RANGE 20V
:MOTOR1:SPEED:RANGE?
-> :MOTOR1:SPEED:RANGE 20.0E+00

Description • This setting is valid when the rotation signal input type (:MoTor<x>:SPEed:TYPE) is set to ANALog (analog input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MoTor<x>:SPEed:SCALing

Sets or queries the scaling factor for rotation speed calculation.

Syntax :MoTor<x>:SPEed:SCALing {<NRf>}
:MoTor<x>:SPEed:SCALing?
<x> = 1 to 4 (motor number)
<NRf> = 0.0001 to 99999.9999

Example :MOTOR1:SPEED:SCALING 1
:MOTOR1:SPEED:SCALING?
-> :MOTOR1:SPEED:SCALING 1.0000

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MoTor<x>:SPEed:TYPE

Sets or queries the input type for the rotation signal.

Syntax :MoTor<x>:SPEed:TYPE {ANALog|PULSe}
:MoTor<x>:SPEed:TYPE?
<x> = 1 to 4 (motor number)

Example :MOTOR1:SPEED:TYPE ANALOG
:MOTOR1:SPEED:TYPE?
-> :MOTOR1:SPEED:TYPE ANALOG

Description • The rotation signal input type is determined by the MTR1 and MTR2 configuration ([[:INPut]:MTR<x>[:CONFig] command), so no setting with this command is required.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:SPeEd:UNIT

Sets or queries the unit appended to rotation speed calculation results.

Syntax :MOTor<x>:SPeEd:UNIT {<String>}
:MOTor<x>:SPeEd:UNIT?

<x> = 1 to 4 (motor number)

<String> = Up to 8 characters

Example :MOTOR1:SPEED:UNIT "rpm"

:MOTOR1:SPEED:UNIT?

-> :MOTOR1:SPEED:UNIT "rpm"

Description • This command does not affect the calculation result.

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

:MOTor<x>:SSPeEd

Sets or queries the frequency measurement source for sync speed (SyncSp) calculation.

Syntax :MOTor<x>:SSPeEd {U<x>|I<x>}
:MOTor<x>:SSPeEd?

MOTor<x>: <x> = 1 to 4 (motor number)

<x> of U<x> and I<x> = 1 to 4 (element)

Example :MOTOR1:SSPEED I1

:MOTOR1:SSPEED? -> :MOTOR1:SSPEED I1

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

:MOTor<x>:SYNChronize

Sets or queries the sync source for rotation speed (Speed) and torque (Torque) calculation.

Syntax :MOTor<x>:SYNChronize {U<x>|I<x>|
EXTernal|Z1|Z3|NONE|CZCRoss}
:MOTor<x>:SYNChronize?

MOTor<x>: <x> = 1 to 4 (motor number)

<x> of U<x> and I<x> = 1 to 4 (element)

EXTernal = external clock input (Ext Clk)

Z1 = Z Phase1 (ChD)

(/EM1, /RM1, or /EM1RM1 option required)

Z3 = Z Phase3 (ChH)

(/EM2, /RM2, or /EM1RM1 option required)

NONE = no sync source

CZCRoss = common zero crossing (valid only during multi-unit synchronous operation)

Example :MOTOR1:SYNCHRONIZE NONE

:MOTOR1:SYNCHRONIZE?

-> :MOTOR1:SYNCHRONIZE NONE

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

:MOTor<x>:TORQue?

Queries all torque (Torque) settings.

Syntax :MOTor<x>:TORQue?
<x> = 1 to 4 (motor number)

:MOTor<x>:TORQue:AUTO

Sets or queries the auto voltage range on/off status for the torque signal (analog input type).

Syntax :MOTor<x>:TORQue:AUTO {<Boolean>}
:MOTor<x>:TORQue:AUTO?

<x> = 1 to 4 (motor number)

Example :MOTOR1:TORQUE:AUTO ON

:MOTOR1:TORQUE:AUTO?

-> :MOTOR1:TORQUE:AUTO 1

Description • This setting is valid when the torque signal input type (:MOTor<x>:TORQue:TYPE) is set to ANALog (analog input).

- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

:MOTor<x>:TORQue:LSCale?

Queries all linear scaling settings for the torque signal (analog input type).

Syntax :MOTor<x>:TORQue:LSCale?
<x> = 1 to 4 (motor number)

Description The linear scaling settings apply when the torque signal input type (:MOTor<x>:TORQue:TYPE) is set to ANALog (analog input).

:MOTor<x>:TORQue:LSCale:AVALue

Sets or queries slope A of the linear scale for the torque signal (analog input type).

Syntax :MOTor<x>:TORQue:LSCale:
AVALue {<NRF>}
:MOTor<x>:TORQue:LSCale:AVALue?
<x> = 1 to 4 (motor number)
<NRF> = 1.000E-03 to 1.000E+06

Example :MOTOR1:TORQUE:LSCALE:AVALUE 1.000

:MOTOR1:TORQUE:LSCALE:AVALUE?

-> :MOTOR1:TORQUE:LSCALE:

AVALUE 1.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSync:SUNit.

5.15 MOTOGroup

:MOTOR<x>:TORQUE:LScale:BVALUE

Sets or queries offset value B of the linear scale for the torque signal (analog input type).

Syntax :MOTOR<x>:TORQUE:LScale:
BVALUE {<NRf>}
:MOTOR<x>:TORQUE:LScale:BVALUE?
<x> = 1 to 4 (motor number)
<NRf> = -1.000E+06 to 1.000E+06

Example :MOTOR1:TORQUE:LSCALE:BVALUE 0
:MOTOR1:TORQUE:LSCALE:BVALUE?
-> :MOTOR1:TORQUE:LSCALE:
BVALUE 0.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:TORQUE:LScale:CALCulate?

Queries all linear scale parameter calculation settings for the torque signal (analog input type).

Syntax :MOTOR<x>:TORQUE:LScale:CALCulate?
<x> = 1 to 4 (motor number)

:MOTOR<x>:TORQUE:LScale:CALCulate:{P1X|P1Y|P2X|P2Y}

Sets or queries the {Point1X|Point1Y|Point2X|Point2Y} data for linear scale parameter calculation for the torque signal (analog input type).

Syntax :MOTOR<x>:TORQUE:LScale:CALCulate:
{P1X|P1Y|P2X|P2Y} {<NRf>}
:MOTOR<x>:TORQUE:LScale:CALCulate:
{P1X|P1Y|P2X|P2Y}?
<x> = 1 to 4 (motor number)
<NRf> = -1.000E+12 to 1.000E+12

Example :MOTOR1:TORQUE:LSCALE:CALCULATE:
P1X 0
:MOTOR1:TORQUE:LSCALE:CALCULATE:P1X?
-> :MOTOR1:TORQUE:LSCALE:CALCULATE:
P1X 0.000E+00

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:TORQUE:LScale:CALCulate:EXECute

Executes linear scale parameter calculation for the torque signal (analog input type).

Syntax :MOTOR<x>:TORQUE:LScale:CALCulate:
EXECute
<x> = 1 to 4 (motor number)

Example :MOTOR1:TORQUE:LSCALE:CALCULATE:
EXECUTE

Description

- This command calculates and sets slope A and offset value B of the linear scale from the previously set data (Point1X, Point1Y, Point2X, Point2Y).
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:TORQUE:PRANge

Sets or queries the range for the torque signal (pulse input type).

Syntax :MOTOR<x>:TORQUE:PRANge {<NRf>,<NRf>}
:MOTOR<x>:TORQUE:PRANge?
<x> = 1 to 4 (motor number)
<NRf> = -10000.0000 to 10000.0000

Example :MOTOR1:TORQUE:PRANGE 50,-50
:MOTOR1:TORQUE:PRANGE?
-> :MOTOR1:TORQUE:PRANGE 50.0000,
-50.0000

Description

- Specify the upper limit and then the lower limit.
- This setting is valid when the torque signal input type (:MOTOR<x>:TORQUE:TYPE) is PULSe (pulse input).
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTOR<x>:TORQUE:RANge

Sets or queries the voltage range for the torque signal (analog input type).

Syntax :MOTOR<x>:TORQUE:RANge {<Voltage>}
:MOTOR<x>:TORQUE:RANge?
<x> = 1 to 4 (motor number)
<Voltage> = 1, 2, 5, 10, 20 (V)

Example :MOTOR1:TORQUE:RANGE 20V
:MOTOR1:TORQUE:RANGE?
-> :MOTOR1:TORQUE:RANGE 20.0E+00

Description

- This setting is valid when the torque signal input type (:MOTOR<x>:TORQUE:TYPE) is set to ANALog (analog input).
- For multi-unit synchronization (option /SY1), specify the target unit with :SYSTEM:MUSYnc:SUNit.

:MOTor<x>:TORQue:RATE?

Queries all rated value settings for the torque signal (pulse input type).

Syntax :MOTor<x>:TORQue:RATE?
<x> = 1 to 4 (motor number)

:MOTor<x>:TORQue:RATE:{UPPer|LOWer}

Sets or queries the {upper|lower} rated value for the torque signal (pulse input type).

Syntax :MOTor<x>:TORQue:RATE:{UPPer|LOWer} {<NRf>,<Frequency>}
:MOTor<x>:TORQue:RATE:{UPPer|LOWer}?
<x> = 1 to 4 (motor number)
<NRf> = -10000.0000 to 10000.0000
Frequency: 1 kHz to 100 MHz

Example :MOTOR1:TORQUE:RATE:UPPER 50,15KHZ
:MOTOR1:TORQUE:RATE:UPPER?
-> :MOTOR1:TORQUE:RATE:
UPPER 50.0000,15.000E+03

Description • This setting is valid when the torque signal input type (:MOTor<x>:TORQue:TYPE) is PULSe (pulse input).
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:TORQue:SCALing

Sets or queries the scaling factor for torque calculation.

Syntax :MOTor<x>:TORQue:SCALing {<NRf>}
:MOTor<x>:TORQue:SCALing?
<x> = 1 to 4 (motor number)
<NRf> = 0.0001 to 99999.9999

Example :MOTOR1:TORQUE:SCALING 1
:MOTOR1:TORQUE:SCALING?
-> :MOTOR1:TORQUE:SCALING 1.0000

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:TORQue:TYPE

Sets or queries the input type for the torque signal.

Syntax :MOTor<x>:TORQue:TYPE {ANALog|PULSe}
:MOTor<x>:TORQue:TYPE?
<x> = 1 to 4 (motor number)

Example :MOTOR1:TORQUE:TYPE ANALOG
:MOTOR1:TORQUE:TYPE?
-> :MOTOR1:TORQUE:TYPE ANALOG

Description For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

:MOTor<x>:TORQue:UNIT

Sets or queries the unit appended to torque calculation results.

Syntax :MOTor<x>:TORQue:UNIT {<String>}
:MOTor<x>:TORQue:UNIT?
<x> = 1 to 4 (motor number)
<String> = Up to 8 characters

Example :MOTOR1:TORQUE:UNIT "Nm"
:MOTOR1:TORQUE:UNIT?
-> :MOTOR1:TORQUE:UNIT "Nm"

Description • This command does not affect the calculation result.
• For multi-unit synchronization (option /SY1), specify the target unit with :SYSTem:MUSYnc:SUNit.

5.16 NUMeric Group

The NUMeric group is for numeric data output.

:NUMeric?

Queries all numeric data output information.

Syntax :NUMeric?

:NUMeric:BYTeorder

Sets or queries the output byte order for numeric data (FLOAT format).

Syntax :NUMeric:BYTeorder {LSBFirst|MSBFirst}
:NUMeric:BYTeorder?

Example :NUMERIC:BYTEORDER LSBFIRST
:NUMERIC:BYTEORDER?
-> :NUMERIC:BYTEORDER LSBFIRST

Description • This setting is valid when :NUMeric:FORMat is set to FLOat.
• The default setting for WT1500 is LSBFirst.

: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 output numeric data changes as follows depending on the :NUMeric:FORMat setting.

- (1) When set to ASCIi
Physical values are output in <NR3> format.
(Elapsed integration time (TIME) only uses <NR1> format)
Data for each item is separated by commas (.).
- (2) When set to FLOat
A header of several bytes (e.g., "#260", "#3208") is added to the beginning of the numeric data block.
Following the header, physical values are output consecutively in IEEE single-precision floating-point (4 byte) format.
The output byte order of data for each item follows the :NUMeric:BYTeorder setting.

• For the format of individual numeric data items, see "Numeric data format" (page 5-109) at the end of this group.

:NUMeric:HOLD

Sets or queries whether to hold (ON) or release (OFF) all numeric data.

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

Example :NUMERIC:HOLD ON
:NUMERIC:HOLD? -> :NUMERIC:HOLD 1

Description • Setting :NUMeric:HOLD to ON before executing :NUMeric[:NORMa]l:VALue? or :NUMeric:LIST:VALue? allows all numeric data at that point to be held internally.

- Even if numeric data is updated on the display, the data remains held until :NUMeric:HOLD is set to OFF.
For example, to acquire various numeric data for each element at the same point in time, proceed as follows.
:NUMeric:HOLD ON
:NUMeric[:NORMa]l:
ITEM1 URMS,1;ITEM2 IRMS,1;...
(Set numeric data items for element 1)
:NUMeric[:NORMa]l:VALue?
(Receive numeric data for element 1)
:NUMeric[:NORMa]l:
ITEM1 URMS,2;ITEM2 IRMS,2;...
(Set numeric data items for element 2)
:NUMeric[:NORMa]l:VALue?
(Receive numeric data for element 2)
:NUMeric[:NORMa]l:
ITEM1 URMS,3;ITEM2 IRMS,3;...
(Set numeric data items for element 3)
:NUMeric[:NORMa]l:VALue?
(Receive numeric data for element 3)
:NUMeric:HOLD OFF
- Setting :NUMeric:HOLD to ON while it is already set to ON releases the held data once and then holds the latest numeric data internally again. When acquiring numeric data continuously, this method eliminates the need to set :NUMeric:HOLD to OFF each time.

:NUMeric:LIST?

Queries all settings related to harmonic measurement numeric list data output.

Syntax :NUMeric:LIST?

Description For settings related to :NUMeric:LIST:ITEM<x>[:FUNCTION], the number of numeric list data output items output equals the count set by :NUMeric:LIST:NUMBER.

:NUMeric:LIST:CLear

Clears (sets to NONE) the output items for harmonic measurement numeric list data.

Syntax :NUMeric:LIST:CLear {ALL|<NRf>[,<NRf>]}

ALL = Clear all items

First <NRf> = 1 to 64 (starting item number for clearing)

Second <NRf> = 1 to 64 (ending item number for clearing)

Example :NUMERIC:LIST:CLEAR ALL

Description If the second <NRf> is omitted, all output items from the starting item number through the last item (64) are cleared.

:NUMeric:LIST:DElete

Deletes output items for harmonic measurement numeric list data.

Syntax :NUMeric:LIST:DElete {<NRf>[,<NRf>]}

First <NRf> = 1 to 64 (starting item number for deletion)

Second <NRf> = 1 to 64 (ending item number for deletion)

Example :NUMERIC:LIST:DELETE 1
(Deletes ITEM1 and shifts ITEM2 onward forward)

:NUMERIC:LIST:DELETE 1,3
(Deletes ITEM1 through ITEM3 and shifts ITEM4 onward forward)

Description • Output items that follow the deleted items are shifted forward to fill the vacated positions, and the remaining empty positions at the end are set to NONE.
• If the second <NRf> is omitted, only the output item at the starting item number is deleted.

:NUMeric:LIST:ITEM<x>[:FUNCTION]

Sets or queries the output item (function and element) for harmonic measurement numeric list data.

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

:NUMeric:LIST:ITEM<x>:FUNCTION?

<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}
(<NRf>=1 to 4)

Example :NUMERIC:LIST:ITEM1:FUNCTION U,1
:NUMERIC:LIST:ITEM1:FUNCTION?
-> :NUMERIC:LIST:ITEM1:FUNCTION U,1

Description For the options for <Function>, see "Numeric list data functions" under "Function options (<Function>)" on page 5-49 at the end of the DISPLAY group.

:NUMeric:LIST:ITEM<x>:UNIT

Sets or queries the unit for harmonic measurement numeric data output items.

Syntax :NUMeric:LIST:ITEM<x>:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}

:NUMeric:LIST:ITEM<x>:UNIT?

<x> = 1 to 64 (item number)

Example :NUMERIC:LIST:ITEM1:UNIT MAIN
:NUMERIC:LIST:ITEM1:UNIT?
-> :NUMERIC:LIST:ITEM1:UNIT MAIN

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:NUMeric:LIST:NUMBER

Sets or queries the number of numeric list data items sent 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 • If the parameter of the :NUMeric:LIST:VALUE? command is omitted, numeric list data for items 1 through (the set value) are output in order.
• The default number of numeric data items is 1.

5.16 NUMeric Group

:NUMeric:LIST:ORDER

Sets or queries the maximum output harmonic order for harmonic measurement numeric list data.

Syntax :NUMeric:LIST:ORDER {<NRf>|ALL}
:NUMeric:LIST:ORDER?
<NRf> = 1 to 100 (ALL)

Example :NUMERIC:LIST:ORDER 100
:NUMERIC:LIST:ORDER?
-> :NUMERIC:LIST:ORDER 100

:NUMeric:LIST:PRESet

Presets the output items for harmonic measurement numeric list data to a fixed pattern.

Syntax :NUMeric:LIST:PRESet {<NRf>}
<NRf> = 1 to 4

Example :NUMERIC:LIST:PRESET 1

Description • For the contents of the preset output items, see “Preset patterns for numeric list data output items for harmonic measurement” (page 5-113) at the end of this group.
• The default setting uses the pattern 2 output items.

:NUMeric:LIST:SElect

Sets or queries the output components for 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 setting is valid only when the harmonic measurement function (option, /G5 or /G6) is installed.
• The options are as follows.
EVEN = Outputs the TOTal, DC, and even-order components
ODD = Outputs the TOTal, DC, and odd-order components
ALL = Outputs all components

:NUMeric:LIST:VALue?

Queries 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,..
(abbreviated) ..,
0.01E+00,0.01E+00 (maximum 102 data items)
• When <NRf> is omitted
(when the :NUMeric:LIST:NUMber setting is 5)
:NUMERIC:LIST:VALUE? -> 103.58E+00,
0.00E+00,103.53E+00,0.09E+00,
2.07E+00,0.04E+00,..(abbreviated) ..,
0.00E+00,0.00E+00
(maximum 102*5 = 510 data items)
• When the :NUMeric:FORMat setting is {FLOat}
:NUMERIC:LIST:VALUE?
-> #N(N-digit byte count)(sequence of data bytes)

Description • One numeric list data set consists of up to 102 numeric data items in the order of TOTal, DC, and 1st through :NUMeric:LIST:ORDER.
• If <NRf> is specified, only the numeric list data for that item number is output. (maximum 102 data items)
• If <NRf> is omitted, numeric list data for item numbers 1 through :NUMeric:LIST:NUMber are output in order. (maximum 102 * :NUMeric:LIST:NUMber data items)
• For the format of individual output numeric data items, see “Numeric data format” (page 5-109) at the end of this group.

:NUMeric:NORMal?

Queries all numeric data output settings.

Syntax :NUMeric:NORMal?

Description For settings related to :NUMeric[:NORMal]:ITE M<x>[:FUNCTION], the number of numeric data output items output equals the count set by :NUMeric[:NORMal]:NUMber.

:NUMERIC[:NORMAL]:CLEAR

Clears (sets to NONE) the numeric data output items.

Syntax :NUMERIC[:NORMAL]:CLEAR {ALL|<NRf>[,<NRf>]}

ALL = Clear all items

First <NRf> = 1 to 1000 (starting item number for clearing)

Second <NRf> = 1 to 1000 (ending item number for clearing)

Example :NUMERIC:NORMAL:CLEAR ALL

Description If the second <NRf> is omitted, all output items from the starting item number through the last item (1000) are cleared.

:NUMERIC[:NORMAL]:DELETE

Deletes numeric data output items.

Syntax :NUMERIC[:NORMAL]:DELETE {<NRf>[,<NRf>]}

First <NRf> = 1 to 1000 (starting item number for deletion)

Second <NRf> = 1 to 1000 (ending item number for deletion)

Example :NUMERIC:NORMAL:DELETE 1
(Deletes ITEM1 and shifts ITEM2 onward forward)

:NUMERIC:NORMAL:DELETE 1,3
(Deletes ITEM1 through ITEM3 and shifts ITEM4 onward forward)

Description

- Output items that follow the deleted items are shifted forward to fill the vacated positions, and the remaining empty positions at the end are set to NONE.
- If the second <NRf> is omitted, only the output item at the starting item number is deleted.

:NUMERIC[:NORMAL]:HEADER?

Queries numeric data headers.

Syntax :NUMERIC[:NORMAL]:HEADER? {<NRf>}

<NRf> = 1 to 1000 (item number)

Example

- When <NRf> is specified
:NUMERIC:NORMAL:HEADER? 1
-> Urms-E1
- When <NRf> is omitted (when :NUMERIC[:NORMAL]:NUMBER is set to 3)
:NUMERIC:NORMAL:HEADER?
-> Urms-E1,Irms-E1,P-E1

Description

- This command outputs the data name (header) of each output item.
- If <NRf> is specified, only the data name of that item number is output.
- If <NRf> is omitted, numeric data for item numbers 1 through :NUMERIC[:NORMAL]:NUMBER are output in order.

:NUMERIC[:NORMAL]:ITEM<x>:FUNCTION

Sets or queries the numeric data output item (function, element, and harmonic order).

Syntax :NUMERIC[:NORMAL]:ITEM<x>:FUNCTION {NONE|<Function>[,<Element>][,<Order>]}

:NUMERIC[:NORMAL]:ITEM<x>:FUNCTION?

<x> = 1 to 1000 (item number)

NONE = no output item

<Function>={URMS|IRMS|P|S|Q|...}

<Element>={<NRf>|SIGMA|SIGMB}
(<NRf>=1 to 4)

<Order>={TOTa|DC|<NRf>}
(<NRf>=1 to 100)

Example

```
:NUMERIC:NORMAL:ITEM1:FUNCTION URMS,1
:NUMERIC:NORMAL:ITEM1:FUNCTION?
-> :NUMERIC:NORMAL:ITEM1:
FUNCTION URMS,1
:NUMERIC:NORMAL:ITEM1:FUNCTION UK,1,1
:NUMERIC:NORMAL:ITEM1:FUNCTION?
->
:NUMERIC:NORMAL:ITEM1:FUNCTION UK,1,1
```

Description

- For the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTa is set.
- In responses for functions that do not require <Element> or <Order>, <Element> or <Order> is omitted.
- For motor functions (<Function> = {SPEed|TORQue|SYNcsp|SLIP|PM}), <Element> is replaced by <Motor>. <Order> is not required.
<Motor> = 1 to 4 (motor number; motor 1 if omitted)

:NUMERIC[:NORMAL]:ITEM<x>:UNIT

Sets or queries the unit for numeric data output items.

Syntax :NUMERIC[:NORMAL]:ITEM<x>:UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}

:NUMERIC[:NORMAL]:ITEM<x>:UNIT?

<x> = 1 to 1000 (item number)

Example

```
:NUMERIC:NORMAL:ITEM1:UNIT MAIN
:NUMERIC:NORMAL:ITEM1:UNIT?
-> :NUMERIC:NORMAL:ITEM1:UNIT MAIN
```

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

5.16 NUMeric Group

:NUMeric[:NORMal]:NUMber

Sets or queries the number of numeric data items sent by :NUMeric[:NORMal]:VALue?.

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 of :NUMeric[:NORMal]:VALue? is omitted, numeric data for items 1 through (the set value) are output in order.
• The default number of numeric data items is 15.

:NUMeric[:NORMal]:PRESet

Presets the numeric data output items to a fixed pattern.

Syntax :NUMeric[:NORMal]:PRESet {<NRf>}
<NRf> = 1 to 4

Example :NUMERIC:NORMAL:PRESET 1

Description • For the contents of the preset output items, see "Preset patterns for numeric data output items" (page 5-111) at the end of this group.
The default setting uses the pattern 2 output items.

:NUMeric[:NORMal]:VALue?

Queries 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,...(abbreviated)...,1.428E+00
• When the :NUMeric:FORMat setting is {FLOat}
:NUMERIC:NORMAL:VALUE?
-> #N(N-digit byte count)(sequence of data bytes)

Description • If <NRf> is specified, only the numeric data for that item number is output.
• If <NRf> is omitted, numeric data for item numbers 1 through :NUMeric[:NORMal]:NUMber are output in order.
• For the format of individual output numeric data items, see "Numeric data format" (page 5-109) at the end of this group.

Numeric data format

Applicable commands

```
:NUMeric:FORMat
:NUMeric:LIST:VALue?
:NUMeric[:NORMal]:VALue?
```

Normal data

- **Frequency (FU, FI, FPLL1, FPLL2)**
 ASCII: <NR3> format (mantissa: up to 5 significant digits, exponent: 2 digits, example: 50.000E+00)
 FLOAT: IEEE single-precision floating-point (4 byte) format
- **Elapsed integration time (ITIME)**
 ASCII: Output in seconds in <NR2> format to ms (example: for 1 hour (1:00:00), 3600.000)
 FLOAT: In seconds in IEEE single-precision floating-point (4 byte) format (example: for 1 hour (1:00:00), 0x45610000)
- **User-defined event (EV1 to EV8)**
 ASCII: Character strings set for true and false, respectively (default values are "True" and "False")
 FLOAT: 0x3F800000 (1) when true, 0x00000000 (0) when false
- **Measurement range**
 ASCII: <NR3> format (mantissa: up to 4 significant digits, exponent: 2 digits, example: 1.000E+03 for 1000 V)
 FLOAT: IEEE single-precision floating-point (4 byte) format
- **Timestamp**
 - Date (TSDate, NPDate, NMDate, NADate, HPDate)**
 ASCII: Example: 2026/01/23; no data: --/--/--
 FLOAT: Unix Time (4 byte)
 * However, 0 indicates no data. Same value as the time of each timestamp (next item).
 - Time (TSTime, NPTime, NMTime, NATime, HPTIME)**
 ASCII: Example: 12:34:56; no data: --/--/--
 FLOAT: Unix Time (4 byte)
 * However, 0 indicates no data. Same value as the date of each timestamp (previous item).
 - Fractional seconds (TSSubsec, NPSubsec, NMSubsec, NASubsec, HPSubsec)**
 ASCII: <NR1> format in microseconds; no data: "NAN" (Not A Number)
 FLOAT: Tick (4 bytes, in ns (0 to 999999999); however, 0xFFFFFFFF indicates no data)
- **No item (NONE)**
 ASCII: NAN (Not A Number)
 FLOAT: 0x7E951BEE(9.91E+37)
- **Other than the above**
 ASCII: <NR3> format (mantissa: up to 5 significant digits, exponent: 2 digits, example: [-]123.45E+00)
 FLOAT: IEEE single-precision floating-point (4 byte) format

Note

- For the phase difference phi (PHI) for elements 1 to 4, when 180 degree (Lead/Lag) display is used, the value is output in the range of –180.00 to 180.00 with Lead (D) as negative and Lag (G) as positive.
 - For sum power values (P, S, Q), the number of mantissa digits may be up to 7 significant digits depending on the combination of the voltage range and current range (power range). For a list of power ranges, see the User's Manual (IM WT1501-01EN).
 - Efficiency (ETA1, ETA2, ETA3, ETA4), content rates (UHDFk, IHDFk, PHDFk), and distortion factors (UTHD, ITHD, PTHD) are fixed at three decimal places. When the value is 100 [%] or greater, the number of mantissa digits becomes 6.
 - The conventional function (TIME) can also be used for elapsed integration time. The ASCII format is the same as before, in <NR1> format in seconds.
 - For details about each timestamp, see the User's Manual (IM WT1501-01EN).
-

Error data

- **No data exists (display: “-----”)**
ASCII: NAN (Not A Number)
FLOAT: 0x7E951BEE(9.91E+37)
- **Overrange (display: “---O L---”)**
- **Overflow (display: “---O F---”)**
- **Data over (display: “ Error “)**
ASCII: INF (INFinity)
FLOAT: 0x7E94F56A(9.9E+37)

List of preset numeric data output items

This list shows the measurement function (Function) and element (Element) assigned to each item number (ITEM<x>).

- * A comparison list of the function names <Function> used in commands and the function names used on this instrument's menu screens is provided in "Numeric data functions" (page 5-45) in the DISPlay group.

Preset patterns for numeric data output 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	SIGMa
50	NONE	
51 to 59	URMS to FI	SIGMb
60	NONE	
61 to 1000	NONE	

Pattern 2

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	SIGMa
76 to 90	URMS to FI	SIGMb
91 to 1000	NONE	

5.16 NUMeric Group

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	SIGMa
100	NONE	
101 to 119	URMS to IMPeak	SIGMb
120	NONE	
121 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	SIGMa
101 to 120	URMS to AHM	SIGMb
121 to 1000	NONE	

Preset patterns for numeric list data output items for harmonic measurement

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 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 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 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 64	NONE	

5.17 STATUS Group

The STATUS group is used to configure and query settings related to status reports.

There is no setup screen corresponding to this group. For details on status reports, see chapter 6.

:STATUS?

Queries all settings for the communication status function.

Syntax :STATUS?

:STATUS:CONDition?

Queries the contents of the condition register.

Syntax :STATUS:CONDition?

Example :STATUS:CONDITION? -> 16

Description For details on the condition register, see Status Reports in chapter 6.

:STATUS:EESE

Sets or queries the extended event enable register.

Syntax :STATUS:EESE <Register>

:STATUS:EESE?

<Register> = 0 to 65535

Example :STATUS:EESE #B00000000000000000

:STATUS:EESE?

-> :STATUS:EESE 0

Description For details on the extended event enable register, see Status Reports in chapter 6.

:STATUS:EESR?

Queries the contents of the extended event register and clears the register.

Syntax :STATUS:EESR?

Example :STATUS:EESR? -> 0

Description For details on the extended event register, see Status Reports in chapter 6.

:STATUS:ERRor?

Queries the code and message of the error that occurred (the first entry in the error queue).

Syntax :STATUS:ERRor?

Example :STATUS:ERROR?

-> 113,"Underfined Header"

Description • When no error has occurred, "0,"Noerror"" is returned.
• You can use STATUS:QMESSAGE to set whether to include the message.

: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 • This setting determines how each bit in the condition register must change for an event to be set. When RISE is specified, an event is set when the bit changes from 0 to 1.
• For details on transition filters, see Status Reports in chapter 6.

:STATUS:QENable

Sets or queries whether messages other than errors are stored in the error queue (ON) or not stored (OFF).

Syntax :STATUS:QENable {<Boolean>}

:STATUS:QENable?

Example :STATUS:QENABLE ON

:STATUS:QENABLE?

-> :STATUS:QENABLE 1

:STATUS:QMESsage

Sets or queries whether to include the message in the response to :STATUS:ERRor? (ON) or not include it (OFF).

Syntax :STATUS:QMESsage {<Boolean>}

:STATUS:QMESsage?

Example :STATUS:QMESSAGE ON

:STATUS:QMESSAGE?

-> :STATUS:QMESSAGE 1

:STATUS:SPOLl?

Executes a serial poll.

Syntax :STATUS:SPOLl?

Example :STATUS:SPOLL? -> :STATUS:SPOLL 0

5.18 STORe Group

The STORe group is for storage.

You can configure the same settings as on the screen that appears when you select the **Setup** menu icon  > **Computation/Output > Store** and query the settings.

:STORe?

Queries all numeric data storage settings.

Syntax :STORe?

:STORe:CONDition?

Queries the storage state.

Syntax :STORe:CONDition?

Example :STORe:CONDITION? -> END

Description • The response contents are as follows.

- END = storage reset state
- READy = storage standby
- RECOrd = storing
- PAUSE = storage paused
- COMPLete = storage complete (or ended due to an error)
- CONVert = converting to a CSV format file
- CLOSE = closing a WTS/HDS format file
- The legacy command :STORe:STATE? can also be used in the same way. (RECOrd = START, PAUSE = STOP, END = RESEt)

: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:END

Ends numeric data storage.

Syntax :STORe:END

Example :STORe:END

Description The legacy command :STORe:RESEt can also be used in the same way.

:STORe:FILE?

Queries all settings for saving data that has been stored.

Syntax :STORe:FILE?

:STORe:FILE:ANAMing

Sets or queries the auto naming function for saving stored numeric data 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 save destination directory for stored numeric data.

Syntax :STORe:FILE:CDIRECTory {<String>}
<String> = Directory name

Example :STORe:FILE:CDIRECTORY "STORE"

Description To move to the parent directory, specify "..".

:STORe:FILE:CONVert?

Queries all settings for converting stored numeric data files to CSV format.

Syntax :STORe:FILE:CONVert?

:STORe:FILE:CONVert:ABORT

Aborts conversion of stored numeric data files to CSV format.

Syntax :STORe:FILE:CONVert:ABORT
Example :STORe:FILE:CONVERT:ABORT

:STORe:FILE:CONVert:AUTO

Sets or queries the automatic conversion on/off status for 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

5.18 STORE Group

:STORE:FILE:CONVERT:EXECute

Executes conversion of a 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 the extension.
• This command is an overlap command.

:STORE:FILE:DRIVE

Sets the save destination drive for stored numeric data.

Syntax :STORE:FILE:DRIVE {USER|USB[,<Nrf>]}
USER = internal memory drive
USB = USB memory drive
<Nrf> = 0, 1, or 2 (drive number)

Example :STORE:FILE:DRIVE USER

:STORE:FILE:FREE?

Queries the free space (byte) on the save destination drive for stored numeric data.

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

:STORE:FILE:NAME

Sets or queries the save file name for stored numeric data.

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 save destination for stored numeric data.

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 setting is valid when the storage mode (:STORE:SMODE) is {MANual|RTIME|INTEGRate}.

:STORE:NUMeric?

Queries all settings for numeric data storage items.

Syntax :STORE:NUMeric?

:STORE:NUMeric:ESElect?

Queries all settings for numeric data storage items (element-specific selection method).

Syntax :STORE:NUMeric:ESElect?

Description This setting is valid when the stored item selection method (:STORE:NUMeric:ITEM) is the element-specific selection method (ESElect).

:STORE:NUMeric:ESElect:ALL

Sets the output elements for all functions at once when storing numeric data.

Syntax :STORE:NUMeric:ESElect:
ALL {<Boolean>}

Example :STORE:NUMERIC:ESELECT:ALL ON

:STORE:NUMeric:ESElect:<Function>

Sets or queries the output elements for each function when storing numeric data.

Syntax :STORE:NUMeric:ESElect:
<Function> {ALL|<Element>[,Element>]
[,Element>][,Element>]...
[,Element>]}
:STORE:NUMeric:ESElect:<Function>?
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<Nrf>|SIGMa|SIGMB}
(<Nrf> = 1 to 4)
ALL = All elements and Σ

Example :STORE:NUMERIC:ESELECT:URMS 1,2,3,
SIGMA
:STORE:NUMERIC:ESELECT:URMS?
-> :STORE:NUMERIC:ESELECT:URMS 1,2,
3,SIGMA

Description • For the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.
• List all elements and Σ to turn output on for in the parameter. To turn output on for all elements and Σ , specify ALL as the parameter.

:STORE:NUMeric:ESElect:PRESet<x>

Presets the output elements of functions to a predefined pattern when storing numeric data.

Syntax :STORE:NUMeric:ESElect:PRESet<x>
<x> = 1 or 2 (preset pattern number)

Example :STORE:NUMERIC:ESELECT:PRESET1

Description For the stored item setting patterns used when a preset is executed, see the User's Manual (IM WT1501-01EN).

:STORE:NUMERIC:ITEM

Sets or queries how numeric data storage items are selected.

Syntax :STORE:NUMERIC:ITEM {SElected}
ESElect}
:STORE:NUMERIC:ITEM?
SElected = manual selection method
ESElect = manual selection method (element-specific selection method)

Example :STORE:NUMERIC:ITEM SELECTED
:STORE:NUMERIC:ITEM?
-> :STORE:NUMERIC:ITEM SELECTED

Description The options are as follows.

SElected = saves to a file the numeric items individually configured with the commands under :STORE:NUMERIC:NORMAL:...

ESElect = saves to a file the numeric items individually configured with the commands under :STORE:NUMERIC:ESElect:...

:STORE:NUMERIC:NORMAL?

Queries all settings for numeric data storage items (manual selection method).

Syntax :STORE:NUMERIC:NORMAL?

Description This setting is valid when the stored item selection method (:STORE:NUMERIC:ITEM) is the manual selection method (SElected).

:STORE:NUMERIC[:NORMAL]:ALL

Sets the output on/off status for all elements and functions at once when storing numeric data.

Syntax :STORE:NUMERIC[:NORMAL]:
ALL {<Boolean>}

Example :STORE:NUMERIC:NORMAL:ALL ON

:STORE:NUMERIC[:NORMAL]:{ELEMENT<x>|SIGMA|SIGMB}

Sets or queries the output on/off status for each element or wiring unit {ΣA|ΣB} when storing numeric data.

Syntax :STORE:NUMERIC[:NORMAL]:{ELEMENT<x>|
SIGMA|SIGMB} {<Boolean>}
:STORE:NUMERIC[:NORMAL]:{ELEMENT<x>|
SIGMA|SIGMB}?
<x> = 1 to 4

Example :STORE:NUMERIC:NORMAL:ELEMENT1 ON
:STORE:NUMERIC:NORMAL:ELEMENT1?
-> :STORE:NUMERIC:NORMAL:ELEMENT1 1

:STORE:NUMERIC[:NORMAL]:<Function>

Sets or queries the output on/off status for each function when storing numeric data.

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 the options for <Function>, see "Numeric data functions" under "Function options (<Function>)" on page 5-45 at the end of the DISPlay group.

:STORE:NUMERIC[:NORMAL]:MOTOR<x>

Sets or queries the output on/off status for each motor when storing numeric data.

Syntax :STORE:NUMERIC[:NORMAL]:
MOTOR<x> {<Boolean>}
:STORE:NUMERIC[:NORMAL]:MOTOR<x>?
<x> = 1 to 4

Example :STORE:NUMERIC:NORMAL:MOTOR1 ON
:STORE:NUMERIC:NORMAL:MOTOR1?
-> :STORE:NUMERIC:NORMAL:MOTOR1 1

:STORE:NUMERIC[:NORMAL]:PRESet<x>

Presets the output on/off status for elements and functions to a predefined pattern when storing numeric data.

Syntax :STORE:NUMERIC[:NORMAL]:PRESet<x>
<x> = 1 or 2 (preset pattern number)

Example :STORE:NUMERIC:NORMAL:PRESET1

Description For the stored item setting patterns used when a preset is executed, see the User's Manual (IM WT1501-01EN).

:STORE:NUMERIC[:NORMAL]:UNIT

Sets or queries the unit for storing numeric data.

Syntax :STORE:NUMERIC:NORMAL:
UNIT {MAIN|SUB1|SUB2|SUB3|SUB4}
:STORE:NUMERIC:NORMAL:UNIT?

Example :STORE:NUMERIC:NORMAL:UNIT MAIN
:STORE:NUMERIC:NORMAL:UNIT?
-> :STORE:NUMERIC:NORMAL:UNIT MAIN

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

5.18 STORE Group

:STORE:NUMERIC:TSTamp

Sets or queries the timestamp acquisition timing when storing numeric data.

Syntax :STORE:NUMERIC:TSTamp {STORE|MEASure}
:STORE:NUMERIC:TSTamp?
STORE = use the time at storage
MEASure = use the time at measurement
Example :STORE:NUMERIC:TSTAMP MEASURE
:STORE:NUMERIC:TSTAMP?
-> :STORE:NUMERIC:TSTAMP MEASURE

:STORE:PAUSE

Stops numeric data storage.

Syntax :STORE:PAUSE
Example :STORE:PAUSE

Description The legacy command :STORE:STOP can also be used in the same way.

:STORE:RECORD

Starts numeric data storage.

Syntax :STORE:RECORD
Example :STORE:RECORD

Description • When :STORE:SMODE is MANual, storage is executed. When it is {RTIME|INTEGRate|EVENT}, the instrument enters storage standby. When it is SINGLE, storage is executed.
• The legacy command :STORE:START can also be used in the same way.

:STORE:RTIME?

Queries the scheduled time for starting/ending storage in real-time mode.

Syntax :STORE:RTIME?

:STORE:RTIME:{START|END}

Sets or queries the scheduled time for storage {start|end} in real-time mode.

Syntax :STORE:RTIME:{START|END} {<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 2018,1,1,0,0,0
:STORE:RTIME:START?
-> :STORE:RTIME:START 2018,1,1,0,0,0

Description This setting is valid when the storage mode (:STORE:SMODE) is RTIME.

:STORE:SASStart

Sets or queries whether to store numeric data at the start of storage.

Syntax :STORE:SASStart {<Boolean>}
:STORE:SASStart?
Example :STORE:SASSTART OFF
:STORE:SASSTART? -> :STORE:SASSTART 0

Description This setting is valid when the storage mode (:STORE:SMODE) is {MANual|RTIME|INTEGRate} ({MANual|RTIME} are valid when the storage interval is not 0).

: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:TEvent

Sets or queries the event that triggers event-synchronized storage mode.

Syntax :STORE:TEvent {<NRf>}
:STORE:TEvent?
<NRf> = 1 to 8 (event number)

Example :STORE:TEVENT 1
:STORE:TEVENT? -> :STORE:TEVENT 1

Description This setting is valid when the storage mode (:STORE:SMODE) is EVENT.

5.19 SYSTem Group

The SYSTem group is for system settings.

You can configure the same settings as on the **System Overview** and **System Configuration** screens that appear when you select the **Setup** menu icon  > **Utility**, and the settings controlled by the **KEY LOCK/LOCAL** key on the front panel, and query the settings.

:SYSTem?

Queries all system settings.

Syntax :SYSTem?

:SYSTem:CLOCK?

Queries all date and time settings.

Syntax :SYSTem:CLOCK?

:SYSTem:CLOCK:DISPlay

Sets or queries the on/off status of the date and time display.

Syntax :SYSTem:CLOCK:DISPlay {<Boolean>}
:SYSTem:CLOCK:DISPlay?

Example :SYSTEM:CLOCK:DISPLAY ON
:SYSTEM:CLOCK:DISPLAY?
-> :SYSTEM:CLOCK:DISPLAY 1

:SYSTem:CLOCK:FORMat

Sets or queries the date display order.

Syntax :SYSTem:CLOCK:FORMat {<NRF>}
:SYSTem:CLOCK:FORMat?
<NRF> = 1 to 4

Example :SYSTEM:CLOCK:FORMAT 1
:SYSTEM:CLOCK:FORMAT?
-> :SYSTEM:CLOCK:FORMAT 1

Description Specifies the format for screen display, CSV files, and communication numeric data output. (The response format for communication settings is fixed.)

- 1: year/month (numeric)/day
 - 2: day/month (numeric)/year
 - 3: day-month (English abbreviation)-year (last two digits)
 - 4: day month (English abbreviation) year
- Default value: 1

:SYSTem:CLOCK:SNTP?

Queries all SNTP date and time settings.

Syntax :SYSTem:CLOCK:SNTP?

:SYSTem:CLOCK:SNTP[:EXECute]

Sets the date and time using SNTP.

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"

Description This is the same setting and query as :SYSTem:TZONE.

:SYSTem:CLOCK:TYPE

Sets or queries the method for setting the date and time.

Syntax :SYSTem:CLOCK:TYPE {MANual|SNTP}
:SYSTem:CLOCK:TYPE?

Example :SYSTEM:CLOCK:TYPE MANUAL
:SYSTEM:CLOCK:TYPE?
-> :SYSTEM:CLOCK:TYPE MANUAL

:SYSTem:CTPower:PPOWER:ELEMENT<x>:STATUS?

Queries the current probe power supply status for the element.

Syntax :SYSTem:CTPower:PPOWER:ELEMENT<x>:
STATUS?
<x> = 1 to 4 (element)

Example :SYSTEM:CTPOWER:PPOWER:ELEMENT1:
STATUS?
-> :SYSTEM:CTPOWER:PPOWER:ELEMENT1:
STATUS 1

Description Returns the current probe power supply status.
0: current probe power supply has stopped abnormally
1: current probe power supply is operating normally

5.19 SYSTem Group

:SYSTem:CTPower:SCONnect:ELEment<x>:STATus?

Queries the current sensor connection status.

Syntax :SYSTem:CTPower:SCONnect:ELEment<x>:
STATus?

<x> = 1 to 4 (element)

Example :SYSTEM:CTPOWER:SCONNECT:ELEMENT1:
STATUS?
-> :SYSTEM:CTPOWER:SCONNECT:
ELEMENT1:STATUS 1

Description Returns the current sensor connection status.
0: sensor not connected
1: sensor connected and power being supplied

:SYSTem:CTPower:SPOWer:ELEment<x>:STATus?

Queries the current sensor power supply status for the element.

Syntax :SYSTem:CTPower:SPOWer:ELEment<x>:
STATus?

<x> = 1 to 4 (element)

Example :SYSTEM:CTPOWER:SPOWER:ELEMENT1:
STATUS?
-> :SYSTEM:CTPOWER:SPOWER:ELEMENT1:
STATUS 1

Description Returns the current sensor power supply status.
0: current sensor power supply has stopped abnormally
1: current sensor power supply is operating normally

: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 "26/01/01"
:SYSTEM:DATE? -> "26/01/01"

Description The year is the last two digits of the Gregorian year.

:SYSTem:DFlow:FREquency

Sets or queries how frequency data is displayed during low-frequency input (no input).

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 how motor data is displayed when there is no pulse input.

Syntax :SYSTem:DFlow:MOTor {0|ERROR}
:SYSTem:DFlow:MOTor?

Example :SYSTEM:DFlow:MOTOR ERROR
:SYSTEM:DFlow:MOTOR?
-> :SYSTEM:DFlow:MOTOR ERROR

Description This command is valid on models with the motor evaluation function 1 for encoders (option /EM1), motor evaluation function 1 for resolvers (option /RM1), motor evaluation function 2 for encoders (option /EM2), motor evaluation function 2 for resolvers (option /RM2), or motor evaluation function for encoders/resolvers (option /EM1RM1).

:SYSTem:DPOint

Sets or queries the type of decimal point used when saving various types of data in ASCII (CSV) format.

Syntax :SYSTem:DPOint {PERiod|COMMa}
:SYSTem:DPOint?

Example :SYSTEM:DPOINT PERIOD
:SYSTEM:DPOINT?
-> :SYSTEM:DPOINT PERIOD

:SYSTem:EClear

Clears the error message displayed on the screen.

Syntax :SYSTem:EClear

Example :SYSTEM:ECLEAR

:SYSTem:ELEment<x>?

Queries all element settings.

Syntax :SYSTem:ELEment<x>?
<x> = 1 to 4 (element)

:SYSTem:ELEment<x>:CTID?

Queries the ID of the CT series installed in the element.

Syntax :SYSTem:ELEment<x>:CTID?
<x> = 1 to 4 (element)

Example :SYSTEM:ELEMENT1:CTID?
-> :SYSTEM:ELEMENT1:CTID CT1000A

Description Returns the detected CTID.

CT2000A = CT2000A

CT1000A = CT1000A

UNKNOWN = unknown

NONE = none

:SYSTem:HOSTname

Sets or queries the host name.

Syntax :SYSTem:HOSTname {<String>}
:SYSTem:HOSTname?
<String> = Up to 32 characters
Example :SYSTEM:HOSTNAME "WT1500 _ AB12CD34EF"
:SYSTEM:HOSTNAME?
-> :SYSTEM:
HOSTNAME "WT1500 _ AB12CD34EF"

:SYSTem:KLOCK

Sets or queries the key lock on/off status.

Syntax :SYSTem:KLOCK {<Boolean>}
:SYSTem:KLOCK?
Example :SYSTEM:KLOCK OFF
:SYSTEM:KLOCK? -> :SYSTEM:KLOCK 0

:SYSTem:LANGuage?

Queries all screen display language settings.

Syntax :SYSTem:LANGuage?

:SYSTem:LANGuage:MENU

Sets or queries the menu language.

Syntax :SYSTem:LANGuage:MENU {JAPANEse|
ENGLish|CHINEse|GERMan}
: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}
:SYSTem:LANGuage:MESSage?
Example :SYSTEM:LANGUAGE:MESSAGE ENGLISH
:SYSTEM:LANGUAGE:MESSAGE?
-> :SYSTEM:LANGUAGE:MESSAGE ENGLISH

:SYSTem:LCD?

Queries all screen settings used when operating this instrument with an external monitor or the Web server function.

Syntax :SYSTem:LCD?

:SYSTem:LCD:COLor?

Queries all display color settings for the screen used when operating this instrument with an external monitor or the Web server function.

Syntax :SYSTem:LCD:COLor?

:SYSTem:LCD:COLor:GRAPH?

Queries all waveform display color settings.

Syntax :SYSTem:LCD:COLor:GRAPH?

:SYSTem:LCD:COLor:GRAPH:CHANnel<x>

Sets or queries the display color of each waveform.

Syntax :SYSTem:LCD:COLor:GRAPH:
CHANnel<x> {YELLow|GREEn|MAGenta|
CYAN|RED|ORANge|LBLue|PURPle|BLUE|
PINK|LGReen|VIOLet|TEAL|SPINK|MGRen|
OLive}
: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
VIOLet = violet
TEAL = teal
SPINK = salmon pink
MGRen = mild green
OLive = olive
Example :SYSTEM:LCD:COLOR:GRAPH:
CHANNEL1 YELLOW
:SYSTEM:LCD:COLOR:GRAPH:CHANNEL1?
-> :SYSTEM:LCD:COLOR:GRAPH:
CHANNEL1 YELLOW

: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

5.19 SYSTem Group

:SYSTem:MLCD?

Queries all settings for the monochrome LCD on the WT1500 instrument.

Syntax :SYSTem:MLCD?

:SYSTem:MLCD:CONTRast

Sets or queries the monochrome LCD contrast.

Syntax :SYSTem:MLCD:CONTRast {<NRf>}
:SYSTem:MLCD:CONTRast?
<NRf> = 1 to 10

Example :SYSTEM:MLCD:CONTRAST 5
:SYSTEM:MLCD:CONTRAST?
-> :SYSTEM:MLCD:CONTRAST 5

:SYSTem:MLCD:DMODE

Sets the screen content of the monochrome LCD.

Syntax :SYSTem:MLCD:DMODE {MEASurement|MULTi|
NETWork|OTHeRs|STATus|SYSTem}

Example :SYSTEM:MLCD:DMODE MEASUREMENT

Description {MULTi} is available only when the multi-unit synchronization interface option (/SY1) is installed.

:SYSTem:MLCD:MODE

Sets or queries the monochrome LCD backlight on/off status.

Syntax :SYSTem:MLCD:MODE {<Boolean>}
:SYSTem:MLCD:MODE?

Example :SYSTEM:MLCD:MODE ON
:SYSTEM:MLCD:MODE?
-> :SYSTEM:MLCD:MODE 1

:SYSTem:MODEL?

Queries the model code (model name).

Syntax :SYSTem:MODEL? {ALL}

Example • Example when ALL is omitted
:SYSTEM:MODEL?
-> :SYSTEM:MODEL "WT1500"
• Example when ALL is specified (valid only when the /SY1 option is installed)
:SYSTEM:MODEL? ALL
-> :SYSTEM:MODEL "WT1500","WT1500",
"WT1500","WT1500","NONE"

Description • This command returns the Model item on the system overview screen (System Overview).
• The following apply when ALL is specified and only when the /SY1 option is installed.
• Query this command when the synchronized operation status is "OPERating".
• If the synchronized operation status is OFF or CONNecting, this command returns NONE.
• If an invalid subunit (no SFP) is queried, this command returns NONE.
• This command returns all of MAIN, SUB1, SUB2, SUB3, and SUB4.

:SYSTem:MSGLog:SAVE[:EXECute]

Saves the message log.

Syntax :SYSTem:MSGLog:SAVE[:EXECute]

Example :SYSTEM:MSGLOG:SAVE:EXECUTE

Description The file name and save path are fixed.
(User/MSGLOG+"Date".CSV)

:SYSTem:MUSYnc?

Queries all multi-unit synchronization status settings.

Syntax :SYSTem:MUSYnc?

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:SYSTem:MUSYnc:ABORT

Aborts multi-unit synchronous operation.

Syntax :SYSTem:MUSYnc:ABORT

Example :SYSTEM:MUSYNC:ABORT

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• This command is available only while multi-unit synchronous operation is in progress or while connection for multi-unit synchronous operation is in progress.

:SYSTem:MUSYnc:CNUMBER?

Queries the number of connected units for multi-unit synchronization (Communication Number).

Syntax :SYSTem:MUSYnc:CNUMBER?
1 to 4 = number of connected units

Example :SYSTEM:MUSYNC:CNUMBER?
-> :SYSTEM:MUSYNC:CNUMBER OFF

Description • This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
• If the multi-unit synchronous operation status (:SYSTem:MUSYnc:SState?) is anything other than OPERating, this command returns an error.

:SYSTem:MUSYnc:CStAtE:SUBUnit<x>?

Queries the connection status (Communication State) between the main unit and a connected subunit in multi-unit synchronization.

Syntax :SYSTem:MUSYnc:CStAtE:SUBUnit<x>?

<x> = 1 to 4 (subunit 1 to 4)

Example :SYSTEM:MUSYNC:CSTATE:SUBUNIT?

-> :SYSTEM:MUSYNC:CSTATE:SUBUNIT NONE

- Description
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
 - The response contents are as follows.
NONE: synchronized operation off; no connection status
NOTUsed: unused subunit (no optical transceiver module in the port)
CONNEcting: connection with the subunit in progress
OK: connection complete
ERRor: Connection errors

:SYSTem:MUSYnc:EXECute

Executes multi-unit synchronous operation.

Syntax :SYSTem:MUSYnc:EXECute

Example :SYSTEM:MUSYNC:EXECUTE

- Description
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
 - This command is available when multi-unit synchronous operation is not in progress and the multi-unit synchronization mode (:SYSTem:MUSYnc:MODE) is MAIN or SUB.

:SYSTem:MUSYnc:MODE

Sets or queries the multi-unit synchronization mode.

Syntax :SYSTem:MUSYnc:MODE {OFF|MAIN|SUB}

:SYSTem:MUSYnc:MODE?

OFF = does not use the multi-unit synchronization function

MAIN = main unit mode for multi-unit synchronization

SUB = subunit mode for multi-unit synchronization

Example :SYSTEM:MUSYNC:MODE MAIN

:SYSTEM:MUSYNC:MODE?

-> :SYSTEM:MUSYNC:MODE MAIN

- Description
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:SYSTem:MUSYnc:SStAtE?

Queries the synchronized operation status (Synchronous State) for multi-unit synchronization.

Syntax :SYSTem:MUSYnc:SStAtE?

Example :SYSTEM:MUSYNC:SSTATE?

-> :SYSTEM:MUSYNC:SSTATE OFF

- Description
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.
 - The response contents are as follows.
OFF: synchronized operation off; no connection status
STANby: standby waiting for connection processing
OPERating: synchronized operation in progress

:SYSTem:MUSYnc:SUNit

Sets or queries the unit used to configure measurement conditions for multi-unit synchronization and the unit to which queries are sent (Selected Unit).

Syntax :SYSTem:MUSYnc:SUNit {MAIN|SUB1|SUB2|SUB3|SUB4}

:SYSTem:MUSYnc:SUNit?

MAIN = main unit

SUB1 = subunit 1

SUB2 = subunit 2

SUB3 = subunit 3

SUB4 = subunit 4

Example :SYSTEM:MUSYNC:SUNIT MAIN

:SYSTEM:MUSYNC:SUNIT?

-> :SYSTEM:MUSYNC:SUNIT MAIN

- Description
- This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

:SYSTem:RZERo

Sets or queries the on/off status of rounding to zero (Rounding to Zero) for numeric data.

Syntax :SYSTem:RZERo {<Boolean>}

:SYSTem:RZERo?

Example :SYSTEM:RZERO ON

:SYSTEM:RZERO? -> :SYSTEM:RZERO 1

5.19 SYSTem Group

:SYSTem:SERial?

Queries the serial number.

Syntax :SYSTem:SERial? {ALL}

- Example
- Example when ALL is omitted
:SYSTEM:SERIAL?
-> :SYSTEM:SERIAL "AB12CD34EF"
 - Example when ALL is specified (valid only when the /SY1 option is installed)
:SYSTEM:SERial? ALL
-> :SYSTEM:SERIAL "AB12CD34EF",
"AB12CD34EF", "AB12CD34EF",
"AB12CD34EF", "NONE"
- Description
- This command returns the string in the No. item on the system overview screen (System Overview).
 - The following apply when ALL is specified and only when the /SY1 option is installed.
 - Query this command when the synchronized operation status is "OPERating".
 - If the synchronized operation status is OFF or CONNecting, this command returns NONE.
 - If an invalid subunit (no SFP) is queried, this command returns NONE.
 - This command returns all of MAIN, SUB1, SUB2, SUB3, and SUB4.

:SYSTem:SUFFix?

Queries the suffix (suffix code).

Syntax :SYSTem:SUFFix? {ALL}

- Example
- Example when ALL is omitted
:SYSTEM:SUFFIX?
-> :SYSTEM:SUFFIX "-HE/M1/EM1"
 - Example when ALL is specified (valid only when the /SY1 option is installed)
:SYSTEM:MODEL? ALL
-> :SYSTEM:SUFFIX "-HE/M1/EM1",
"-HE/M1/EM1", "-HE/M1/EM1",
"-HE/M1/EM1", "NONE"

- Description
- This command returns the string in the Suffix item on the system overview screen (System Overview).
 - The following apply when ALL is specified and only when the /SY1 option is installed.
 - Query this command when the synchronized operation status is "OPERating".
 - If the synchronized operation status is OFF or CONNecting, this command returns NONE.
 - If an invalid subunit (no SFP) is queried, this command returns NONE.
 - This command returns all of MAIN, SUB1, SUB2, SUB3, and SUB4.

:SYSTem:TIME

Sets or queries the time.

Syntax :SYSTem:TIME {<String>}

:SYSTem:TIME?

<String> = "HH:MM:SS"
(HH = hours, MM = minutes, SS = seconds)

- Example
- ```
:SYSTEM:TIME "14:30:00"
:SYSTEM:TIME? -> "14:30:00"
```

### :SYSTem:TLOCK

Sets or queries the touch lock on/off status.

Syntax :SYSTem:TLOCK {<Boolean>}

:SYSTem:TLOCK?

- Example
- ```
:SYSTEM:TLOCK OFF  
:SYSTEM:TLOCK? -> :SYSTEM:TLOCK 0
```

:SYSTem:TSYNchro:IEEE1588?

Queries all IEEE 1588 time synchronization settings.

Syntax :SYSTem:TSYNchro:IEEE1588?

:SYSTem:TSYNchro:IEEE1588:DMEChani sm

Sets or queries the IEEE 1588 time synchronization delay mechanism (time difference calculation method).

Syntax :SYSTem:TSYNchro:IEEE1588:

DMEChanism {E2E|P2P}

:SYSTem:TSYNchro:IEEE1588:

DMEChanism?

- Example
- ```
:SYSTEM:TSYNCHRO:IEEE1588:
DMECHANISM E2E
:SYSTEM:TSYNCHRO:IEEE1588:
DMECHANISM?
-> :SYSTEM:TSYNCHRO:IEEE1588:
DMECHANISM E2E
```

- Description
- This setting can be configured when IEEE 1588 time synchronization (:SYSTem:TSYNchro:IEEE1588:STATe) is OFF (0).

### :SYSTem:TSYNchro:IEEE1588:DNUMBER

Sets or queries the IEEE 1588 time synchronization domain number.

Syntax :SYSTem:TSYNchro:IEEE1588:

DNUMBER {<NRf>}

:SYSTem:TSYNchro:IEEE1588:DNUMBER?

<NRf> = 0 to 255

- Example
- ```
:SYSTEM:TSYNCHRO:IEEE1588:DNUMBER 0  
:SYSTEM:TSYNCHRO:IEEE1588:DNUMBER?  
-> :SYSTEM:TSYNCHRO:IEEE1588:  
DNUMBER 0
```

- Description
- This setting can be configured when IEEE 1588 time synchronization (:SYSTem:TSYNchro:IEEE1588:STATe) is OFF (0).

:SYSTem:TSYNchro:IEEE1588:NLAYer

Sets or queries the IEEE 1588 time synchronization network layer.

Syntax :SYSTem:TSYNchro:IEEE1588:
NLAYer {LAYER2|LAYER3}
:SYSTem:TSYNchro:IEEE1588:NLAYer?

Example :SYSTEM:TSYNCHRO:IEEE1588:
NLAYER LAYER3
:SYSTEM:TSYNCHRO:IEEE1588:NLAYER?
-> :SYSTEM:TSYNCHRO:IEEE1588:
NLAYER LAYER3

Description This setting can be configured when IEEE 1588 time synchronization (:SYSTem:TSYNchro:IEEE1588:STATE) is OFF (0).

:SYSTem:TSYNchro:IEEE1588:PState?

Queries the IEEE 1588 time synchronization reception status.

Syntax :SYSTem:TSYNchro:IEEE1588:PState?

Example :SYSTEM:TSYNCHRO:IEEE1588:PSTATE?
-> STABLE

Description The response contents are as follows.
OFF: time synchronization off
LOCK: normal reception
UNLOCK: reception unavailable
STABLE: normal reception after synchronization
UNSTable: reception unstable after synchronization
SERRor: synchronization failed

:SYSTem:TSYNchro:IEEE1588[:STATE]

Sets or queries IEEE 1588 time synchronization on/off.

Syntax :SYSTem:TSYNchro:
IEEE1588[:STATE] {<Boolean>}
:SYSTem:TSYNchro:IEEE1588:STATE?

Example :SYSTEM:TSYNCHRO:IEEE1588:STATE ON
:SYSTEM:TSYNCHRO:IEEE1588:STATE?
-> :SYSTEM:TSYNCHRO:IEEE1588:STATE 1

:SYSTem:TZONE

Sets or queries the time zone.

Syntax :SYSTem:TZONE {<String>}
:SYSTem:TZONE?
<String> = "HH:MM" (HH = hours, MM = minutes)

Example :SYSTEM:TZONE "09:00"
:SYSTEM:TZONE?
-> :SYSTEM:TZONE "09:00"

Description This is the same setting and query as :SYSTem:CLOCK:SNTP:GMTTime.

: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

5.20 UDP Group

The UDP group is for UDP Data Output.

You can configure the same settings as on the UDP Data Output settings screen that appears when you select the **Setup** menu icon  > **Computation/Output** > **UDP** and query the settings.

:UDP:IDENTifier

Sets or queries the identifier of the header output through UDP.

Syntax :UDP:IDENTifier {<String>} <NRf>
<String> = hexadecimal string from 0 to ffff
:UDP:IDENTifier?
Example :UDP:IDENTIFIER 87,845754
:UDP:IDENTIFIER?
-> :UDP:IDENTIFIER PORT 87,845754

:UDP:IPAddress

Sets or queries the destination IP address for UDP output.

Syntax :UDP:IPAddress {<NRf>,<NRf>,<NRf>,<NRf>}
:UDP:IPAddress?
Example :UDP:IPADDRESS 255,255,255,255
:UDP:IPADDRESS?
-> :UDP:IPADDRESS 255,255,255,255

:UDP:MODE

Sets or queries whether UDP output starts automatically when the power is turned on.

Syntax :UDP:MODE <NRf>
:UDP:MODE?
Example :UDP:MODE 1
:UDP:MODE? -> :UDP:MODE 1

:UDP:PORT

Sets or queries the destination port number for UDP output.

Syntax :UDP:PORT <NRf>
:UDP:PORT?
Example :UDP:PORT 50000
:UDP:PORT? -> :UDP:PORT 50000

:UDP:START

Starts UDP output.

Syntax :UDP:START
Example :UDP:START

:UDP:STOP

Stops UDP output.

Syntax :UDP:STOP
Example :UDP:STOP

Related communication commands

The output items (ITEM) for UDP data output depend on the settings of the following commands.

However, the format of numeric data output through UDP is fixed at FLOAT (IEEE single-precision floating-point (4-byte) format), regardless of the command settings.

:NUMeric:BYTeorder
:NUMeric[:NORMal]:VALue?
:NUMeric[:NORMal]:CLEar
:NUMeric[:NORMal]:DELeTe
:NUMeric[:NORMal]:HEADer?
:NUMeric[:NORMal]:ITEM<x>[:FUNctIon]
:NUMeric[:NORMal]:NUMber
:NUMeric[:NORMal]:PRESet
:NUMeric[:NORMal]:VALue?

5.21 UPDate Group

The UPDate group is for data updates.

You can configure the same settings as on the data update setup screen that appears when you select the **Setup** menu icon  > **Computation/Output** > **Update Rate/Averaging** and query the settings.

:UPDate?

Queries all data update settings.

Syntax :UPDate?

:UPDate:AUTO?

Queries all auto update mode settings.

Syntax :UPDate:AUTO?

Description This setting is valid when the update mode (:UPDate[:MODE]) is AUTO.

:UPDate:AUTO:RATE

Sets or queries the data update interval in auto update mode.

Syntax :UPDate:AUTO:RATE {<Time>}
:UPDate:AUTO:RATE?
<Time> = 10 or 50 (ms)

Example :UPDATE:AUTO:RATE 50MS
:UPDATE:AUTO:RATE?
-> :UPDATE:AUTO:RATE 50.0E-03

:UPDate:AUTO:TIMEout

Sets or queries the timeout value in auto update mode.

Syntax :UPDate:AUTO:TIMEout {<NRf>}
:UPDate:AUTO:TIMEout?
<NRf> = 1, 5, 10, 20

Example :UPDATE:AUTO:TIMEOUT 1
:UPDATE:AUTO:TIMEOUT?
-> :UPDATE:AUTO:TIMEOUT 1

:UPDate[:MODE]

Sets or queries the update mode.

Syntax :UPDate[:MODE] {CONSTant|TRIGger|
AUTO}
:UPDate:MODE?

Example :UPDATE:MODE CONSTANT
:UPDATE:MODE?
-> :UPDATE:MODE CONSTANT

:UPDate:RATE

Sets or queries the data update interval.

Syntax :UPDate:RATE {<Time>}
:UPDate:RATE?
<Time> = 10, 50, 100, 200, or 500 (ms), 1, 2, 5,
10, or 20 (s)

Example :UPDATE:RATE 500MS
:UPDATE:RATE?
-> :UPDATE:RATE 500.0E-03

Description This setting is valid when the update mode (:UPDate[:MODE]) is {CONSTant|TRIGger}.

:UPDate:TRIGger?

Queries all trigger update mode settings.

Syntax :UPDate:TRIGger?

Description This setting is valid when the update mode (:UPDate[:MODE]) is TRIGger.

:UPDate:TRIGger:LEVel

Sets or queries the trigger level.

Syntax :UPDate:TRIGger:LEVel {<NRf>}
:UPDate:TRIGger:LEVel?
<NRf> = -100.0 to 100.0(%)

Example :UPDATE:TRIGGER:LEVEL 0
:UPDATE:TRIGGER:LEVEL?
-> :UPDATE:TRIGGER:LEVEL 0.0

Description Set as a percentage of the displayed full-scale value.

:UPDate:TRIGger:MODE

Sets or queries the trigger mode.

Syntax :UPDate:TRIGger:MODE {AUTO|NORMal}
:UPDate:TRIGger:MODE?
Example :UPDATE:TRIGGER:MODE AUTO
:UPDATE:TRIGGER:MODE?
-> :UPDATE:TRIGGER:MODE AUTO

:UPDate:TRIGger:SLOPe

Sets or queries the trigger slope.

Syntax :UPDate:TRIGger:SLOPe {RISE|FALL|
BOTH}
:UPDate:TRIGger:SLOPe?
Example :UPDATE:TRIGGER:SLOPE RISE
:UPDATE:TRIGGER:SLOPE?
-> :UPDATE:TRIGGER:SLOPE RISE

5.21 UPDate Group

:UPDate:TRIGger:SOURce

Sets or queries the trigger source.

Syntax :UPDate:TRIGger:SOURce {U<x>|I<x>|
 EXTErnal}
 :UPDate:TRIGger:SOURce?
 <x> = 1 to 4 (element)
 EXTErnal = external trigger input (Ext Clk)

Example :UPDATE:TRIGGER:SOURCE U1
 :UPDATE:TRIGGER:SOURCE?
 -> :UPDATE:TRIGGER:SOURCE U1

:UPDate:TRIGger:UNIT

Sets or queries the trigger unit (Trigger Unit).

Syntax :UPDate:TRIGger:UNIT {MAIN|SUB1|SUB2|
 SUB3|SUB4}
 :UPDate:TRIGger:UNIT?

Example :UPDATE:TRIGGER:UNIT MAIN
 :UPDATE:TRIGGER:UNIT?
 -> :UPDATE:TRIGGER:UNIT MAIN

Description This setting is valid only when the multi-unit
 synchronization interface option (/SY1) is
 installed.

5.22 WAVEform Group

The WAVEform group is for acquired waveform data.

:WAVEform?

Queries all waveform display data output settings.

Syntax :WAVEform?

:WAVEform:BYTEorder

Sets or queries the output byte order of waveform display data (FLOAT format) sent with :WAVEform:SEND?.

Syntax :WAVEform:BYTEorder {LSBFirst|MSBFirst}
:WAVEform:BYTEorder?
Example :WAVEFORM:BYTEORDER MSBFIRST
:WAVEFORM:BYTEORDER?
-> :WAVEFORM:BYTEORDER MSBFIRST

Description This setting is valid when :WAVEform:FORMat is FLOAT.

:WAVEform:END

Sets or queries the output end point of waveform display data sent with :WAVEform:SEND?.

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 waveform display data sent with :WAVEform:SEND?.

Syntax :WAVEform:FORMat {AScii|FLOat}
:WAVEform:FORMat?
Example :WAVEFORM:FORMAT FLOAT
:WAVEFORM:FORMAT?
-> :WAVEFORM:FORMAT FLOAT

Description For differences in waveform display data output according to the format setting, see the description for :WAVEform:SEND?.

:WAVEform:HOLD

Sets or queries whether to hold (ON) or release (OFF) all waveform display data.

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

Example :WAVEFORM:HOLD ON
:WAVEFORM:HOLD?
-> :WAVEFORM:HOLD 1

Description • Before executing :WAVEform:SEND?, setting :WAVEform:HOLD to ON allows all waveform data at that point to be held internally.
• Even if the waveform display is updated on the display screen, the waveform data is held until :WAVEform:HOLD is set to OFF.

Example To obtain the U1 and I1 waveform display data at the same point in time, do the following.

```
:WAVEform:HOLD ON
:WAVEform:TRACe U1
:WAVEform:SEND?
(Receive the U1 waveform display data)
:WAVEform:TRACe I1
:WAVEform:SEND?
(Receive the I1 waveform display data)
:WAVEform:HOLD OFF
```

- If ON is set while :WAVEform:HOLD is already set to ON, the waveform display data is released once, and the latest waveform data is held internally again. When obtaining waveform display data continuously, this method eliminates the need to set :WAVEform:HOLD to OFF each time.

:WAVEform:LENGth?

Queries the total number of data points in the waveform specified by :WAVEform:TRACe.

Syntax :WAVEform:LENGth?
Example :WAVEFORM:LENGTH? -> 1602

Description The number of data points is fixed. This command always returns 1602.

5.22 WAVEform Group

:WAVEform:SEND?

Queries the waveform display data specified by :WAVEform:TRACe.

Syntax :WAVEform:SEND?

Example

- When :WAVEform:FORMat is {ASCIi}
:WAVEFORM:SEND? -> <NR3>,<NR3>,...
- When :WAVEform:FORMat is {FLOat}
:WAVEFORM:SEND?
-> #4(4-digit byte count)(sequence of data bytes)

Description The format of the output waveform display data changes as follows according to the :WAVEform:FORMat setting.

(1) When ASCIi is selected
Physical values are output in <NR3> format.
The data for each point is separated by a comma (.).

(2) When set to FLOat
Physical values are output in IEEE single-precision floating-point (4-byte) format.
The output byte order of the data for each point follows the :WAVEform:BYTeorder setting.

: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 waveform display data sent with :WAVEform:SEND?.

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 waveform to which :WAVEform:SEND? applies.

Syntax :WAVEform:TRACe {U<x>|I<x>|SPEed|
TORQue|AUX<x>}

:WAVEform:TRACe?

<x> of U<x> and I<x> = 1 to 4 (element)

<x> of SPEed<x> and TORQue<x> = 1 to 4
(MOTOR number)

<x> of AUX<x> = 1 to 8 (AUX input channel)

Example :WAVEFORM:TRACE U1

:WAVEFORM:TRACE?

-> :WAVEFORM:TRACE U1

Description

- {SPEed|TORQue|AUX<x>} is valid when the encoder motor evaluation function 1 option (/EM1), resolver motor evaluation function 1 option (/RM1), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.
- The values <x> = 3 or 4 for {SPEed<x>|TORQue<x>} and <x> = 5 to 8 for AUX<x> are available when the encoder motor evaluation function 2 option (/EM2), resolver motor evaluation function 2 option (/RM2), or encoder/resolver motor evaluation function option (/EM1RM1) is installed.

:WAVEform:TRIGger?

Queries the trigger position of the acquired waveform.

Syntax :WAVEform:TRIGger?

Example :WAVEFORM:TRIGGER? -> 0

Description The trigger position is always the beginning of the waveform display data, so this command returns 0.

:WAVEform:UNIT

Sets or queries the unit for the waveform to which :WAVEform:SEND? applies.

Syntax :WAVEform:UNIT {MAIN|SUB1|SUB2|SUB3|
SUB4}

:WAVEform:UNIT?

Example :WAVEFORM:UNIT MAIN

:WAVEFORM:UNIT?

-> :WAVEFORM:UNIT MAIN

Description This setting is valid only when the multi-unit synchronization interface option (/SY1) is installed.

5.23 Common Command Group

This group contains commands defined in IEEE 488.2-1992 that are independent of the instrument's individual functions.

*CAL?

Executes calibration (zero-level compensation, the same operation as selecting Misc > Cal [Execute]) and queries the result.

Syntax *CAL?

Example *CAL? -> 0

Description When calibration finishes normally, 0 is returned. If there is an abnormality, 1 is returned.

*CLS

Clears the standard event register, extended event register, and error queue.

Syntax *CLS

Example *CLS

Description

- When the *CLS command is immediately after the program message terminator, the output queue is also cleared.
- For details on each register and queue, see chapter 6.

*ESE

Sets or queries the value of the standard event enable register.

Syntax *ESE {<NRf>}

*ESE?

<NRf> = 0 to 255

Example *ESE 251

*ESE? -> 251

Description

- Set this register by using the decimal sum of each bit.
Example If *ESE 251 is specified, the standard event enable register is set to 11111011. In other words, bit 2 of the standard event register is disabled, and 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).
- Querying with *ESE? does not clear the contents of the standard event enable register.
- For details on the standard event enable register, see section 6.3.

*ESR?

Queries the value of the standard event register and clears it at the same time.

Syntax *ESR?

Example *ESR? -> 32

Description

- The decimal sum of each bit is returned.

- When an SRQ has occurred, you can determine what type of event occurred.
Example If 32 is returned, this indicates that the standard event register is set to 00100000. In other words, you can see that an SRQ occurred because a command syntax error occurred.
- Querying with *ESR? clears the contents of the standard event register.
- For details on the standard event register, see section 6.3.

*IDN?

Queries the model.

Syntax *IDN?

Example *IDN?

-> YOKOGAWA,WT1504-10-WB2-HV2,
1234567890,F1.01

Description

- This command returns information in the format <Manufacturer>,<Model>,<Serial No.>,<Firmware version>.
- The format of <Model> is model name (6 digits)-specifications (2 digits)-WB element configuration (3 digits)-HV element configuration (3 digits).
For details on the model name, specifications, and element configuration, see "Checking the package contents" in the Getting Started Guide (IM WT1501-02EN).

*OPC

Sets bit 0 (OPC bit) of the standard event register to 1 when the specified overlap command is completed.

Syntax *OPC

Example *OPC

Description

- For details on how to synchronize using *OPC, see page 4-12.
- Use :COMMunicate:OPSE to specify overlap commands.
- Operation of *OPC that is not at the end of a message is not guaranteed.

5.23 Common Command Group

*OPC?

Returns ASCII code 1 if the specified overlap command has been completed.

Syntax *OPC?

Example *OPC? -> 1

Description

- For details on how to synchronize using *OPC, see page 4-12.
- Use :COMMunicate:OPSE to specify overlap commands.
- Operation of *OPC? that is not at the end of a message is not guaranteed.

*OPT?

Queries the installed options.

Syntax *OPT?

Example *OPT? -> M1,EM1

Description The availability of the following options is returned.

- 32 GB internal memory (M1),
- Motor Evaluation for Encoder 1 (EM1),
- Motor Evaluation for Resolver 1 (RM1),
- Motor Evaluation for Encoder 2 (EM2),
- Motor Evaluation for Resolver 2 (RM2),
- Motor Evaluation for Encoder/Resolver (EM1RM1),
- Multi-unit synchronization interface (SY1),
- CAN/CAN FD interface (CN1)

*RST

Initializes the settings.

Syntax *RST

Example *RST

Description

- Previously sent *OPC and *OPC? commands are also canceled.
- All settings except communication settings are set to their factory default settings.

*SRE

Sets or queries the value of the service request enable register.

Syntax *SRE {<NRf>}

*SRE?

<NRf> = 0 to 255

Example *SRE 239

*SRE? -> 175 (because the setting of bit 6 (MSS) is ignored)

Description

- Set this register by using the decimal sum of each bit.
- For example, if *SRE 239 is specified, the service request enable register is set to 11101111. In other words, bit 4 of the service request enable register is disabled, and bit 4 (MAV) of the status byte register is not set to 1 even when the output queue is not empty.
- However, bit 6 (MSS) of the status byte register is the MSS bit itself, so it is ignored.
- The default value is *SRE 0 (all bits disabled).
- Querying with *SRE? does not clear the contents of the service request enable register.
- For details on the service request enable register, see section 6.2.

*STB?

Queries the value of the status byte register.

Syntax *STB?

Example *STB? -> 4

Description

- The sum of each bit is returned as a decimal number.
- Because the register is read without executing a serial poll, bit 6 is the MSS bit, not RQS.

Example If 4 is returned, this indicates that the status byte register is set to 00000100. In other words, you can see that the error queue is not empty (an error occurred).

- Querying with *STB? does not clear the contents of the status byte register.
- For details on the status byte register, see section 6.2.

*TRG

Executes single measurement (the same operation as selecting Misc > Single [Execute]).

Syntax *TRG

Example *TRG

Description The multiline message GET (Group Execute Trigger) also performs the same operation as this command.

***TST?**

Executes a self-test and queries the result.

Syntax *TST?

Example *TST? -> 0

Description • The self-test consists of testing each internal memory.

- When the self-test result is normal, 0 is returned. If there is an abnormality, 1 is returned. Also, if the self-test cannot be executed due to the state of the instrument (integration in progress or storage in progress), the corresponding error code is returned.

***WAI**

Waits to execute instructions that follow *WAI until the specified overlap command is completed.

Syntax *WAI

Example *WAI

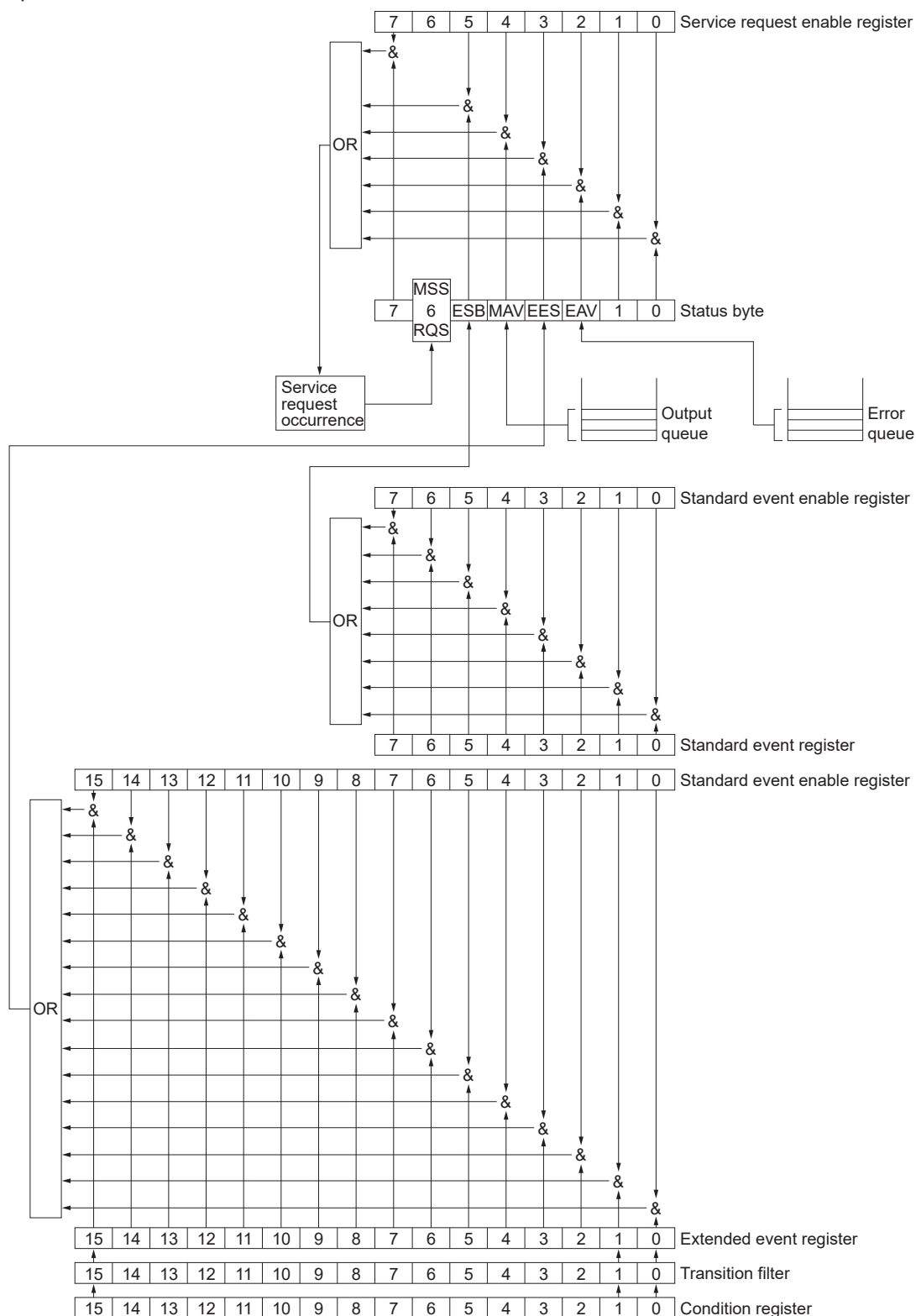
Description • For details on how to synchronize using *WAI, see page 4-11.

- Use :COMMunicate:OPSE to specify overlap commands.

6.1 About Status Reports

Status reports

The figure below shows the status reports read by a serial poll. This is an extension of what is specified in IEEE 488.2-1992.



Overview of registers and queues

Name	Function	Write	Read
Status byte	–	–	Serial poll (RQS), *STB? (MSS)
Service request enable register	Status byte mask	*SRE	*SRE?
Standard event register	Changes in the state of the instrument	–	*ESR?
Standard event enable register	Masks the standard event register	*ESE	*ESE?
Extended event register	Changes in the state of the instrument	–	:STATUS:EESR?
Extended event enable register	Masks the extended event register	:STATUS:ESEE	:STATUS:ESEE?
Condition register	Current state of the instrument	–	:STATUS:CONDition?
Transition filter	Conditions for changes in the extended event register	:STATUS:FILTer<x>	:STATUS:FILTer<x>?
Output queue	Stores response messages to queries	Query commands	
Error queue	Stores error numbers and messages	–	:STATUS:ERRor?

Registers and queues that affect the status byte

The registers that affect the status byte bits are summarized below.

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 registers used to mask bits so that they do not affect the status byte, even when set to 1, are summarized below.

Register to be masked	Register used for masking
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 bits of the standard event enable register to 1 or 0. Use the *ESE? command to check whether the bits of the standard event enable register are set to 1 or 0. These commands are described in detail in chapter 5.

6.2 Status Byte

Status byte

	RQS						
7	6	ESB	MAV	EES	EAV	1	0
	MSS						

Bits 0, 1, and 7

Unused (always 0)

Bit 2 EAV (Error Available)

This bit is set to 1 when the error queue is not empty. In other words, it is set to 1 when an error occurs. See page 6-8.

Bit 3 EES (Extend Event Summary Bit)

This bit is set to 1 when the logical product of the extended event register and the extended event enable register is not 0. In other words, it is set to 1 when an event occurs inside the instrument. See page 6-7.

Bit 4 MAV (Message Available)

This bit is set to 1 when the output queue is not empty. In other words, it is set to 1 when a query has been made and there is data to output. See page 6-8.

Bit 5 ESB (Event Summary Bit)

This bit is set to 1 when the logical product of the standard event register and the standard event enable register is not 0. In other words, it is set to 1 when an event occurs inside the instrument. See page 6-5.

Bit 6 RQS (Request Service)/MSS (MasterStatus Summary)

This bit is set to 1 when the logical product of the status byte other than bit 6 and the service request enable register is not 0. In other words, it is set to 1 when the instrument is requesting service from the controller.

RQS is set to 1 when MSS changes from 0 to 1 and is cleared when a serial poll is performed or when MSS becomes 0.

Masking bits

To mask a bit in the status byte so that it is not a cause of SRQ, set the corresponding bit in the service request enable register to 0.

For example, to mask bit 2 (EAV) so that service is not requested even if an error occurs, set bit 2 of the service request enable register to 0. Use the *SRE command to do this. Use *SRE? to query whether the bits of the service request enable register are set to 1 or 0. For details on the *SRE command, see chapter 5.

Status byte operation

When bit 6 of the status byte becomes 1, a service request is generated. When any bit other than bit 6 becomes 1, bit 6 becomes 1 (when the corresponding bit of the service request enable register is also 1). For example, when an event occurs and the logical OR of the bits of the standard event register and its enable register becomes 1, bit 5 (ESB) is set to 1. At this time, if bit 5 of the service request enable register is 1, bit 6 (MSS) is set to 1, and the instrument requests service from the controller.

You can also check what type of event occurred by reading the contents of the status byte.

Reading the status byte

There are the following two methods for reading the contents of the status byte.

Querying with *STB?

When you query with *STB?, bit 6 is MSS. Therefore, MSS is read. After reading, no bits in the status byte are cleared.

Serial poll

When you perform a serial poll, bit 6 is RQS. Therefore, RQS is read. After reading, only RQS is cleared. MSS cannot be read by a serial poll.

Clearing the status byte

There is no method for forcibly clearing all bits of the status byte. The bits cleared by each operation are shown below.

When queried with *STB?

No bits are cleared.

When a serial poll is performed

Only the RQS bit is cleared.

When the *CLS command is received

When the *CLS command is received, the status byte itself is not cleared, but the contents of the standard event register and other registers that affect the bits are cleared. As a result, the corresponding bits of the status byte are cleared. Because the output queue cannot be cleared with the *CLS command, bit 4 (MAV) of the status byte is not affected. However, if the *CLS command is received immediately after the program message terminator, the output queue is also cleared.

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) Power on

This bit is set to 1 when the instrument is turned on.

Bit 6 URQ (User Request) User request

Unused (always 0)

Bit 5 CME (Command Error) Command syntax error

This bit is set to 1 when there is an error in the command syntax.

Example The command name is misspelled, or character data that is not one of the choices is received.

Bit 4 EXE (Execution Error) Command 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 A parameter is outside the setting range, or a command related to an option that is not installed is received.

Bit 3 DDE (Device Error) Instrument-specific error

This bit is set to 1 when the command cannot be executed due to an internal instrument cause other than a command syntax error or command execution error.

Bit 2 QYE (Query Error) Query error

This bit is set to 1 when a query command has been sent but the output queue is empty or data has been lost.

Example There is no response data, or the output queue overflowed and data was lost.

Bit 1 RQC (Request Control) Request control

Unused (always 0)

Bit 0 OPC (Operation Complete) Operation complete

This bit is set to 1 when the operation specified by the *OPC command (see chapter 6-5) is completed.

Masking bits

To mask a bit in the standard event register so that it is not a cause of bit 5 (ESB) of the status byte, set the corresponding bit in the standard event enable register to 0.

For example, to mask bit 2 (QYE) so that ESB is not set to 1 even if a query error occurs, set bit 2 of the standard event enable register to 0. Use the *ESE command to do this. Use *ESE? to query whether the bits of the standard event enable register are set to 1 or 0. For details on the *ESE command, see chapter 5.

Standard event register operation

The standard event register records eight types of events that occur inside the instrument. When any bit set to 1, bit 5 (ESB) of the status byte is set to 1 (when the corresponding bit of the standard event enable register is also 1).

Example

1. A query error occurs
2. Bit 2 (QYE) is set to 1
3. If bit 2 of the standard event enable register is 1, bit 5 (ESB) of the status byte is set to 1

You can also check the type of event that occurred inside the instrument by reading the contents of the standard event register.

Reading the standard event register

The contents of the standard event register can be read with `*ESR?`. After the register is read, it is cleared.

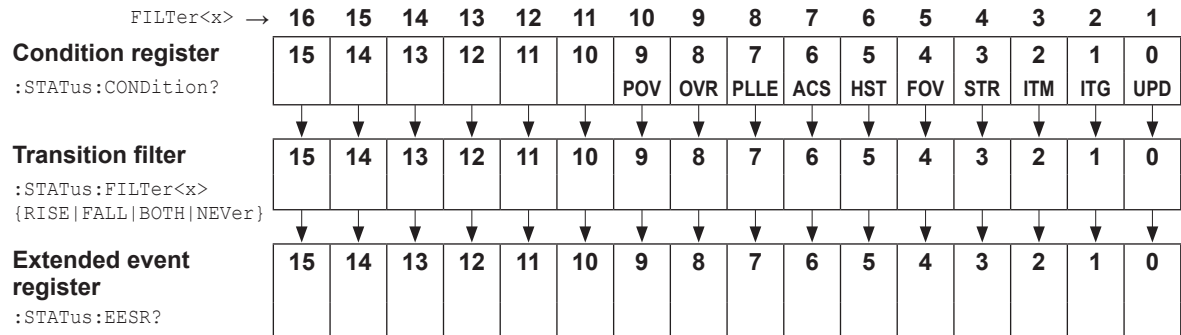
Clearing the standard event register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read with `*ESR?`
- When the `*CLS` command is received
- When the power is turned on again

6.4 Extended Event Register

The extended event register contains the results obtained when changes in the condition register, which indicates the instrument's internal state, are edge-detected by the transition filter.



The condition register bits are described below.

Bit 0	UPD (Updating)	Set to 1 while measurement data is being updated. The falling edge of UPD (1 → 0) indicates the end of updating.
Bit 1	ITG (Integrate Busy)	Set to 1 during integration.
Bit 2	ITM (Integrate Timer Busy)	Set to 1 while the integration timer is running.
Bit 3	STR (Store Busy)	Set to 1 during storage.
Bit 4	FOV (Frequency Over)	Set to 1 when there is a frequency error.
Bit 5	HST (HISTory)	Set to 1 while history data is being updated. The falling edge of HST (1 → 0) indicates the end of updating. * See Appendix 2, "History Function."
Bit 6	ACS (Accessing)	Set to 1 while any drive is being accessed.
Bit 7	PLLE (PLL Source Input Error)	Set to 1 during harmonic measurement when there is no input to the PLL source and synchronization cannot be established.
Bit 8	OVR (ElementX Measured Data Over)	Set to 1 when the voltage or current value of any element is overrange.
Bit 9	POV (ElementX Input Peak Over)	Set to 1 when peak-over is detected in any element.

The transition filter parameters extract changes in the specified condition register bits (numeric suffixes 1 to 16) as follows and update the extended event register.

RISE	Sets the specified bit of the extended event register to 1 when a 0 → 1 change occurs.
FALL	Sets the specified bit of the extended event register to 1 when a 1 → 0 change occurs.
BOTH	Sets the specified bit of the extended event register to 1 when a 0 → 1 or 1 → 0 change occurs.
NEVer	Always 0.

6.5 Output and Error Queues

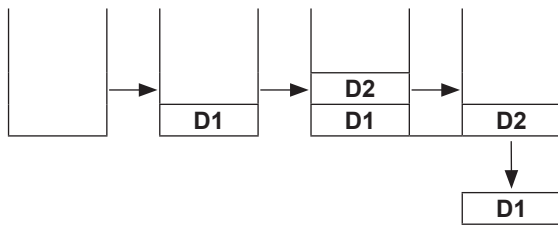
Output queue

The output queue stores response messages to queries. For example, when `:WAVeform:SEND?` is sent to request output of acquired waveform data, the data is stored in the output queue until it is read.

As shown in the figure below, data is stored in order and read from oldest to newest. The output queue is also cleared in the following cases.

- When a new message is received from the controller
- When a deadlock state occurs (see page 4-3)
- When a device clear (DCL or SDC) is received
- When the power is turned on again

Note that the output queue cannot be cleared with the `*CLS` command. You can check whether the output queue is empty with bit 4 (MAV) of the status byte.



Error queue

The error queue stores the error number and message when an error occurs. For example, if the controller sends an incorrect program message, the error number and error message 113, “Undefined header” are stored in the error queue when the error is displayed.

The contents of the error queue can be read with the `:STATus:ERRor?` query. As with the output queue, the error queue is read from oldest to newest.

When the error queue overflows, the last message is replaced with the message 350, “Queue overflow.”

The error queue is also cleared in the following cases.

- When the `*CLS` command is received
- When the power is turned on again

You can check whether the error queue is empty with bit 2 (EAV) of the status byte.

7.1 Overview of Modbus/TCP Communication

Modbus/TCP is a communication protocol used to communicate with PCs, PLCs (sequencers), and other devices over networks such as Ethernet by using the TCP/IP protocol. Through this protocol, data is exchanged with connected devices by reading from and writing to the internal registers of this instrument.

Hereafter, higher-level devices such as PCs are referred to as “client devices.”

This instrument can be connected to IEEE 802.3-compliant networks (1000BASE-T/100BASE-TX/10BASE-T).

In communication using the Modbus/TCP protocol, port number 502 is generally used.

Part names and functions

For details on the names and functions of the parts related to the Ethernet interface used for Modbus/TCP communication, see section 1.1.

Modbus/TCP functions and specifications

Specifications of the Ethernet interface used for Modbus/TCP communication

Item	Specification
Number of ports	1
Connector type	RJ-45
Electrical and mechanical specifications	Complies with IEEE 802.3
Transmission system	Ethernet 1000BASE-T/100BASE-TX/10BASE-T
Communication protocol	TCP/IP
Supported services	DHCP, DNS, SNTP client, FTP server and client, remote control (VXI-11, Modbus/TCP)
Port number	VXI-11: 11024/tcp Modbus/TCP: 502/tcp

This instrument operates as a Modbus server. The number of simultaneous connections is 1.

Connection procedure

Connect the STP (Shielded Twisted-Pair) cable connected to a hub or similar device to the Ethernet port on the rear panel of this instrument. For details, see section 1.3.

Instrument settings

Setting TCP/IP 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	Read holding registers	Can be read consecutively from 0001 to 0010
04	Read input registers	Up to 125 registers can be read consecutively from 0001 to 5648.
06	Write holding registers	Only one register from 0001 to 0180 can be written.

Specifying registers

The methods for specifying registers from client devices are as follows.

- When using commercially available SCADA or similar software, specify the Ref No. (reference number) given in section 7.3, “Register Functions and Uses.”
- When using a user-created communication program, specify the H No. (relative number) given in section 7.3, “Register Functions and Uses.”

Example: When specifying voltage RMS data for element 1 (lower-order float), input register 0101

- For a request using commercially available SCADA or similar software, specify 30101 as the Ref No.
- For a request using a user-created communication program, specify 0064 as the H No.

7.3 Register Functions and Uses

In this instrument, measurement data, setup data, and other data are assigned to internal registers for Modbus/TCP. By sending commands to this instrument through Modbus/TCP communication and reading from and writing to the internal registers of this instrument, client devices can obtain measurement data and perform operations such as starting integration.

Register configuration

	Register number	Category	Description
Input registers	0001 to 0070	Measurement data, status	Data and status independent of elements and Σ
	0071 to 0086	Measurement range information	Voltage and current ranges of the specified element
	0101 to 0200	Measurement data (element 1)	Normal and harmonic measurement data for element 1
	0201 to 0300	Measurement data (element 2)	Normal and harmonic measurement data for element 2
	0301 to 0400	Measurement data (element 3)	Normal and harmonic measurement data for element 3
	0401 to 0500	Measurement data (element 4)	Normal and harmonic measurement data for element 4
	1001 to 1100	Measurement data (ΣA)	Normal measurement data for wiring unit ΣA
	1101 to 1200	Measurement data (ΣB)	Normal measurement data for wiring unit ΣB
	1501 to 1664	Measurement data (motor)	Motor input measurement data
	1701 to 1732	Measurement data (auxiliary signal)	Auxiliary input measurement data
	2001 to 4000	Communication output item data	Measurement data linked to communication output item settings (:NUMERIC[:NORMAL]:ITEM<x>[:FUNCTION] Commands)
	4001 to 4096	Numeric display item data (4 items display)	Measurement data linked to screen display item settings Numeric display (Numeric): 4 items display
	4101 to 4292	Numeric display item data (8 items display)	Measurement data linked to screen display item settings Numeric display (Numeric): 8 items display
	4301 to 4684	Numeric display item data (16 items display)	Measurement data linked to screen display item settings Numeric display (Numeric): 16 items display
	5001 to 5648	Numeric display item data (Matrix display)	Measurement data linked to screen display item settings Numeric display (Numeric): Matrix display
Hold registers	0001 to 0010	Control data	Register value retention and integration operation control
Other		Reserved area (blank)	Cannot be used; writing is not guaranteed.

Register map (input register)

Reg No.	Ref No.	H No.	Register name		Register content		Remarks
Status and data independent of elements and wiring units							
0001	30001	0000	Update Count		Data update counter	(uint 16)	0 to 65535
0002	30002	0001	Peak Over		Peak over status	(uint 16)	0 to 255
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 content	Remarks
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)

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)
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)

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content		Remarks
0117	30117	0074	fl1	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	(float, lower 2 bytes)	
0134	30134	0085		H	watt hours 1	(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	(float, lower 2 bytes)	
0140	30140	008B		H	watt hours 1	(float, upper 2 bytes)	
0141	30141	008C	q+1	L	Positive ampere hours 1	(float, lower 2 bytes)	
0142	30142	008D		H		(float, upper 2 bytes)	
0143	30143	008E	q-1	L	Negative ampere hours 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	(float, lower 2 bytes)	
0150	30150	0095		H	calibrated to the rms value 1	(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	(float, lower 2 bytes)	
0158	30158	009D		H	calibrated to the rms value 1	(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)	
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 Power1	(float, lower 2 bytes)	
0170	30170	00A9		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register content	Remarks
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 $\lambda 1(1)$ of the 1st harmonic (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 harmonic (fundamental wave), $\phi 1(1)$	(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						
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						

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content	Remarks
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						
Wiring unit ΣA normal measurement data and delta computation data						
1001	31001	03E8	Urms ΣA	L	True rms voltage of ΣA	(float, lower 2 bytes)
1002	31002	03E9		H		(float, upper 2 bytes)
1003	31003	03EA	Irms ΣA	L	True rms current of ΣA	(float, lower 2 bytes)
1004	31004	03EB		H		(float, upper 2 bytes)
1005	31005	03EC	P ΣA	L	Active power of ΣA	(float, lower 2 bytes)
1006	31006	03ED		H		(float, upper 2 bytes)
1007	31007	03EE	S ΣA	L	Apparent power of ΣA	(float, lower 2 bytes)
1008	31008	03EF		H		(float, upper 2 bytes)
1009	31009	03F0	Q ΣA	L	Reactive power of ΣA	(float, lower 2 bytes)
1010	31010	03F1		H		(float, upper 2 bytes)
1011	31011	03F2	Lambda ΣA	L	Power factor of ΣA (λ)	(float, lower 2 bytes)
1012	31012	03F3		H		(float, upper 2 bytes)
1013	31013	03F4	Phi ΣA	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 ($\Delta U1$)	(float, lower 2 bytes)
1016	31016	03F7		H		(float, upper 2 bytes)
1017	31017	03F8	DeltaU2	L	ΣA delta voltage computation 2 ($\Delta U2$)	(float, lower 2 bytes)
1018	31018	03F9		H		(float, upper 2 bytes)
1019	31019	03FA	DeltaU3	L	ΣA delta voltage computation 3 ($\Delta U3$)	(float, lower 2 bytes)
1020	31020	03FB		H		(float, upper 2 bytes)
1021	31021	03FC	DeltaU Σ	L	ΣA delta voltage computation 1 ($\Delta U\Sigma$)	(float, lower 2 bytes)
1022	31022	03FD		H		(float, upper 2 bytes)
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 ($\Delta P1$)	(float, lower 2 bytes)
1026	31026	0401		H		(float, upper 2 bytes)
1027	31027	0402	DeltaP2	L	ΣA delta power computation 2 ($\Delta P2$)	(float, lower 2 bytes)
1028	31028	0403		H		(float, upper 2 bytes)
1029	31029	0404	DeltaP3	L	ΣA delta power computation 3 ($\Delta P3$)	(float, lower 2 bytes)
1030	31030	0405		H		(float, upper 2 bytes)
1031	31031	0406	DeltaP Σ	L	ΣA delta power computation Σ ($\Delta P\Sigma$)	(float, lower 2 bytes)
1032	31032	0407		H		(float, upper 2 bytes)
1033	31033	0408	WP ΣA	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	q ΣA	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 hours of ΣA	(float, lower 2 bytes)
1044	31044	0413		H		(float, upper 2 bytes)
1045	31045	0414	WS ΣA	L	Volt-ampere hours of ΣA	(float, lower 2 bytes)
1046	31046	0415		H		(float, upper 2 bytes)
1047	31047	0416	WQ ΣA	L	Var hours of ΣA	(float, lower 2 bytes)
1048	31048	0417		H		(float, upper 2 bytes)

Reg No.	Ref No.	H No.	Register name		Register content	Remarks
1049	31049	0418	UmnΣA	L	Rectified mean voltage calibrated (float, lower 2 bytes)	
1050	31050	0419		H	to the rms value of ΣA (float, upper 2 bytes)	
1051	31051	041A	UdcΣA	L	DC voltage of ΣA (Simple (float, lower 2 bytes)	
1052	31052	041B		H	average) (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 (float, lower 2 bytes)	
1058	31058	0421		H	to the rms value of ΣA (float, upper 2 bytes)	
1059	31059	0422	IdcΣA	L	DC current of ΣA (simple (float, lower 2 bytes)	
1060	31060	0423		H	average) (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 (float, lower 2 bytes)	
1072	31072	042F		H	components of voltage ΣA (float, upper 2 bytes)	
1073	31073	0430	UΣA(1)	L	1st harmonic value of voltage (float, lower 2 bytes)	
1074	31074	0431		H	ΣA (fundamental wave) (float, upper 2 bytes)	
1075	31075	0432	IΣA(Total)	L	Total value of all harmonic (float, lower 2 bytes)	
1076	31076	0433		H	components of current ΣA (float, upper 2 bytes)	
1077	31077	0434	IΣA(1)	L	1st harmonic value of current (float, lower 2 bytes)	
1078	31078	0435		H	ΣA (fundamental wave) (float, upper 2 bytes)	
1079	31079	0436	PΣA(Total)	L	Total value of all harmonic (float, lower 2 bytes)	
1080	31080	0437		H	components of active power ΣA (float, upper 2 bytes)	
1081	31081	0438	PΣA(1)	L	1st harmonic value of active (float, lower 2 bytes)	
1082	31082	0439		H	power ΣA (fundamental wave) (float, upper 2 bytes)	
1083	31083	043A	SΣA(Total)	L	Total value of all harmonic (float, lower 2 bytes)	
1084	31084	043B		H	components of apparent power ΣA (float, upper 2 bytes)	
1085	31085	043C	SΣA(1)	L	1st harmonic value of apparent (float, lower 2 bytes)	
1086	31086	043D		H	power ΣA (fundamental wave) (float, upper 2 bytes)	
1087	31087	043E	QΣA(Total)	L	Total value of all harmonic (float, lower 2 bytes)	
1088	31088	043F		H	components of reactive power ΣA (float, upper 2 bytes)	
1089	31089	0440	QΣA(1)	L	1st harmonic value of reactive (float, lower 2 bytes)	
1090	31090	0441		H	power ΣA (fundamental wave) (float, upper 2 bytes)	
1091	31091	0442	PhiUi-Uj	L	Phase difference between (float, lower 2 bytes)	
1092	31092	0443		H	Ui(1) and Uj(1) of ΣA (float, upper 2 bytes)	
1093	31093	0444	PhiUi-Uk	L	Phase difference between (float, lower 2 bytes)	
1094	31094	0445		H	Ui(1) and Uk(1) of ΣA (float, upper 2 bytes)	
1095	31095	0446	PhiUi-li	L	Phase difference between (float, lower 2 bytes)	
1096	31096	0447		H	Ui(1) and li(1) of ΣA (float, upper 2 bytes)	
1097	31097	0448	PhiUj-lj	L	Phase difference between (float, lower 2 bytes)	
1098	31098	0449		H	Uj(1) and lj(1) of ΣA (float, upper 2 bytes)	
1099	31099	044A	PhiUk-lk	L	Phase difference between (float, lower 2 bytes)	
1100	31100	044B		H	Uk(1) and lk(1) of ΣA (float, upper 2 bytes)	

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name	Register content	Remarks
---------	---------	-------	---------------	------------------	---------

Wiring unit Σ B normal measurement data and delta computation data

1101	31101	044C	Urms Σ B	L	True rms voltage of Σ B	(float, lower 2 bytes)	
1102	31102	044D		H		(float, upper 2 bytes)	
to 1170							

Wiring unit Σ B harmonic measurement data

1171	31171	0492	U Σ B(Total)	L	Total value of all harmonic	(float, lower 2 bytes)	
1172	31172	0493		H	components of voltage Σ B	(float, upper 2 bytes)	
to 1200							

Motor input 1 measurement data

1501	31501	05DC	Speed1	L	Rotation speed (Motor1)	(float, lower 2 bytes)	
1502	31502	05DD		H		(float, upper 2 bytes)	
1503	31503	05DE	Torque1	L	Torque (Motor1)	(float, lower 2 bytes)	
1504	31504	05DF		H		(float, upper 2 bytes)	
1505	31505	05E0	SyncSp1	L	Sync speed (Motor1)	(float, lower 2 bytes)	
1506	31506	05E1		H		(float, upper 2 bytes)	
1507	31507	05E2	Slip1	L	Slip (Motor1)	(float, lower 2 bytes)	
1508	31508	05E3		H		(float, upper 2 bytes)	
1509	31509	05E4	Pm1	L	Mechanical output of the	(float, lower 2 bytes)	
1510	31510	05E5		H	motor (Motor1)	(float, upper 2 bytes)	
1511	31511	05E6	SpeedRange1	L	Rotation speed range value	(float, lower 2 bytes)	
1512	31512	05E7		H	(Motor1)	(float, upper 2 bytes)	
1513	31513	05E8	TorqueRange1	L	Torque range value (Motor1)	(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)	

Motor input 2 measurement data

1551	31551	060E	Speed2	L	Rotation speed (Motor2)	(float, lower 2 bytes)	
1552	31552	060F		H		(float, upper 2 bytes)	
1553	31553	0610	Torque2	L	Torque (Motor2)	(float, lower 2 bytes)	
1554	31554	0611		H		(float, upper 2 bytes)	
1555	31555	0612	SyncSp2	L	Sync speed (Motor2)	(float, lower 2 bytes)	
1556	31556	0613		H		(float, upper 2 bytes)	
1557	31557	0614	Slip2	L	Slip (Motor2)	(float, lower 2 bytes)	
1558	31558	0615		H		(float, upper 2 bytes)	
1559	31559	0616	Pm2	L	Mechanical output of the	(float, lower 2 bytes)	
1560	31560	0617		H	motor (Motor2)	(float, upper 2 bytes)	
1561	31561	0618	SpeedRange2	L	Rotation speed range value	(float, lower 2 bytes)	
1562	31562	0619		H	(Motor2)	(float, upper 2 bytes)	

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content	Remarks
1563	31563	061A	TorqueRange2	L	Torque range value (Motor2)	(float, lower 2 bytes)
1564	31564	061B		H		(float, upper 2 bytes)

Motor input 3 measurement data

1601	31601	0640	Speed3	L	Rotation speed (Motor3)	(float, lower 2 bytes)
1602	31602	0641		H		(float, upper 2 bytes)
to 1614						

Motor input 4 measurement data

1651	31651	0672	Speed4	L	Rotation speed (Motor4)	(float, lower 2 bytes)
1652	31652	0673		H		(float, upper 2 bytes)
to 1664						

Auxiliary input measurement data

1701	31701	06A4	Aux1	L	Auxiliary input 1	(float, lower 2 bytes)
1702	31702	06A5		H		(float, upper 2 bytes)
1703	31703	06A6	Aux2	L	Auxiliary input 2	(float, lower 2 bytes)
1704	31704	06A7		H		(float, upper 2 bytes)
1705	31705	06A8	AuxRange1	L	AUX1 range value	(float, lower 2 bytes)
1706	31706	06A9		H		(float, upper 2 bytes)
1707	31707	06AA	AuxRange2	L	AUX2 range value	(float, lower 2 bytes)
1708	31708	06AB		H		(float, upper 2 bytes)
1709	31709	06AC	Aux3	L	Auxiliary input 3	(float, lower 2 bytes)
1710	31710	06AD		H		(float, upper 2 bytes)
1711	31711	06AE	Aux4	L	Auxiliary input 4	(float, lower 2 bytes)
1712	31712	06AF		H		(float, upper 2 bytes)
1713	31713	06B0	AuxRange3	L	AUX3 range value	(float, lower 2 bytes)
1714	31714	06B1		H		(float, upper 2 bytes)
1715	31715	06B2	AuxRange4	L	AUX4 range value	(float, lower 2 bytes)
1716	31716	06B3		H		(float, upper 2 bytes)
1717	31717	06B4	Aux5	L	Auxiliary input 5	(float, lower 2 bytes)
1718	31718	06B5		H		(float, upper 2 bytes)
1719	31719	06B6	Aux6	L	Auxiliary input 6	(float, lower 2 bytes)
1720	31720	06B7		H		(float, upper 2 bytes)
1721	31721	06B8	AuxRange5	L	AUX5 range value	(float, lower 2 bytes)
1722	31722	06B9		H		(float, upper 2 bytes)
1723	31723	06BA	AuxRange6	L	AUX6 range value	(float, lower 2 bytes)
1724	31724	06BB		H		(float, upper 2 bytes)
1725	31725	06BC	Aux7	L	Auxiliary input 7	(float, lower 2 bytes)
1726	31726	06BD		H		(float, upper 2 bytes)
1727	31727	06BE	Aux8	L	Auxiliary input 8	(float, lower 2 bytes)
1728	31728	06BF		H		(float, upper 2 bytes)
1729	31729	06C0	AuxRange7	L	AUX7 range value	(float, lower 2 bytes)
1730	31730	06C1		H		(float, upper 2 bytes)
1731	31731	06C2	AuxRange8	L	AUX8 range value	(float, lower 2 bytes)
1732	31732	06C3		H		(float, upper 2 bytes)

Measurement data assigned to communication output items (:NUMERIC[:NORMAL]:ITEM<x>[:FUNCTION] commands)

2001+ (X-1)×2		Item	L	Measurement data assigned	(float, lower 2 bytes)	Default setting
2001+ (X-1)×2+1			H	to ItemX	(float, upper 2 bytes)	
2001	32001	07D0	L	Measurement data assigned	(float, lower 2 bytes)	Urms1
2002	32002	07D1	H	to ITEM1	(float, upper 2 bytes)	
2003	32003	07D2	L	Measurement data assigned	(float, lower 2 bytes)	Umn1
2004	32004	07D3	H	to ITEM2	(float, upper 2 bytes)	
2005	32005	07D4	L	Measurement data assigned	(float, lower 2 bytes)	Udc1
2006	32006	07D5	H	to ITEM3	(float, upper 2 bytes)	

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content	Remarks
2007	32007	07D6	Item4	L	Measurement data assigned	(float, lower 2 bytes)
2008	32008	07D7		H	to ITEM4	(float, upper 2 bytes)
2009	32009	07D8	Item5	L	Measurement data assigned	(float, lower 2 bytes)
2010	32010	07D9		H	to ITEM5	(float, upper 2 bytes)
2011	32011	07DA	Item6	L	Measurement data assigned	(float, lower 2 bytes)
2012	32012	07DB		H	to ITEM6	(float, upper 2 bytes)
2013	32013	07DC	Item7	L	Measurement data assigned	(float, lower 2 bytes)
2014	32014	07DD		H	to ITEM7	(float, upper 2 bytes)
2015	32015	07DE	Item8	L	Measurement data assigned	(float, lower 2 bytes)
2016	32016	07DF		H	to ITEM8	(float, upper 2 bytes)
2017	32017	07E0	Item9	L	Measurement data assigned	(float, lower 2 bytes)
2018	32018	07E1		H	to ITEM9	(float, upper 2 bytes)
2019	32019	07E2	Item10	L	Measurement data assigned	(float, lower 2 bytes)
2020	32020	07E3		H	to ITEM10	(float, upper 2 bytes)
2021	32021	07E4	Item11	L	Measurement data assigned	(float, lower 2 bytes)
2022	32022	07E5		H	to ITEM11	(float, upper 2 bytes)
2023	32023	07E6	Item12	L	Measurement data assigned	(float, lower 2 bytes)
2024	32024	07E7		H	to ITEM12	(float, upper 2 bytes)
2025	32025	07E8	Item13	L	Measurement data assigned	(float, lower 2 bytes)
2026	32026	07E9		H	to ITEM13	(float, upper 2 bytes)
2027	32027	07EA	Item14	L	Measurement data assigned	(float, lower 2 bytes)
2028	32028	07EB		H	to ITEM14	(float, upper 2 bytes)
2029	32029	07EC	Item15	L	Measurement data assigned	(float, lower 2 bytes)
2030	32030	07ED		H	to ITEM15	(float, upper 2 bytes)
to						
3999	33999	0F9E	Item1000	L	Measurement data assigned	(float, lower 2 bytes)
4000	34000	0F9F		H	to ITEM1000	(float, upper 2 bytes)

Measurement data corresponding to numeric data display items (4 Items display)

4001+ (X-1)×2			4Items_X	L	Measurement data for 4 Items	(float, lower 2 bytes)	Default value (for the 4 element model)
4001+ (X-1)×2+1				H	display, item number = X	(float, upper 2 bytes)	
4001	34001	0FA0	4Items_1	L	Measurement data for 4 Items	(float, lower 2 bytes)	Urms1
4002	34002	0FA1		H	display, item number = 1 (page 1)	(float, upper 2 bytes)	
4003	34003	0FA2	4Items_2	L	Measurement data for 4 Items	(float, lower 2 bytes)	Irms1
4004	34004	0FA3		H	display, item number = 2 (page 1)	(float, upper 2 bytes)	
4005	34005	0FA4	4Items_3	L	Measurement data for 4 Items	(float, lower 2 bytes)	P1
4006	34006	0FA5		H	display, item number = 3 (page 1)	(float, upper 2 bytes)	
4007	34007	0FA6	4Items_4	L	Measurement data for 4 Items	(float, lower 2 bytes)	λ1
4008	34008	0FA7		H	display, item number = 4 (page 1)	(float, upper 2 bytes)	
4009	34009	0FA8	4Items_5	L	Measurement data for 4 Items	(float, lower 2 bytes)	Urms2
4010	34010	0FA9		H	display, item number = 5 (page 2)	(float, upper 2 bytes)	
4011	34011	0FAA	4Items_6	L	Measurement data for 4 Items	(float, lower 2 bytes)	Irms2
4012	34012	0FAB		H	display, item number = 6 (page 2)	(float, upper 2 bytes)	
4013	34013	0FAC	4Items_7	L	Measurement data for 4 Items	(float, lower 2 bytes)	P2
4014	34014	0FAD		H	display, item number = 7 (page 2)	(float, upper 2 bytes)	
4015	34015	0FAE	4Items_8	L	Measurement data for 4 Items	(float, lower 2 bytes)	λ2
4016	34016	0FAF		H	display, item number = 8 (page 2)	(float, upper 2 bytes)	
4017	34017	0FB0	4Items_9	L	Measurement data for 4 Items	(float, lower 2 bytes)	Urms3
4018	34018	0FB1		H	display, item number = 9 (page 3)	(float, upper 2 bytes)	
to							
4095	34096	0FFE	4Items_48	L	Measurement data for 4 Items	(float, lower 2 bytes)	
4096	34096	0FFF		H	display, item number = 48 (page 12)	(float, upper 2 bytes)	

Measurement data corresponding to numeric data display items (8 Items display)

4101+ (X-1)×2			8Items_X	L	Measurement data for 8 Items	(float, lower 2 bytes)	Default value (for the 4 element model)
4101+ (X-1)×2+1				H	display, item number = X	(float, upper 2 bytes)	

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content		Remarks
4101	34101	1004	8Items_1	L	Measurement data for 8 Items	(float, lower 2 bytes)	Urms1
4102	34102	1005		H	display, item number = 1 (page 1)	(float, upper 2 bytes)	
4103	34103	1006	8Items_2	L	Measurement data for 8 Items	(float, lower 2 bytes)	Irms1
4104	34104	1007		H	display, item number = 2 (page 1)	(float, upper 2 bytes)	
4105	34105	1008	8Items_3	L	Measurement data for 8 Items	(float, lower 2 bytes)	P1
4106	34106	1009		H	display, item number = 3 (page 1)	(float, upper 2 bytes)	
4107	34107	100A	8Items_4	L	Measurement data for 8 Items	(float, lower 2 bytes)	S1
4108	34108	100B		H	display, item number = 4 (page 1)	(float, upper 2 bytes)	
4109	34109	100C	8Items_5	L	Measurement data for 8 Items	(float, lower 2 bytes)	Q1
4110	34110	100D		H	display, item number = 5 (page 1)	(float, upper 2 bytes)	
4111	34111	100E	8Items_6	L	Measurement data for 8 Items	(float, lower 2 bytes)	λ1
4112	34112	100F		H	display, item number = 6 (page 1)	(float, upper 2 bytes)	
4113	34113	1010	8Items_7	L	Measurement data for 8 Items	(float, lower 2 bytes)	φ1
4114	34114	1011		H	display, item number = 7 (page 1)	(float, upper 2 bytes)	
4115	34115	1012	8Items_8	L	Measurement data for 8 Items	(float, lower 2 bytes)	fU1
4116	34116	1013		H	display, item number = 8 (page 1)	(float, upper 2 bytes)	
4117	34117	1014	8Items_9	L	Measurement data for 8 Items	(float, lower 2 bytes)	Urms2
4118	34118	1015		H	display, item number = 9 (page 2)	(float, upper 2 bytes)	
to							
4291	34291	10C2	8Items_96	L	Measurement data for 8 Items	(float, lower 2 bytes)	
4292	34292	10C3		H	display, item number = 96 (page 12)	(float, upper 2 bytes)	

Measurement data corresponding to numeric data display items (16 Items display)

4301+ (X-1)×2			16Items_X	L	Measurement data for 16 Items	(float, lower 2 bytes)	Default value (for the 4 element model)
4301+ (X-1)×2+1				H	display, item number = X	(float, upper 2 bytes)	
4301	34301	10CC	16Items_1	L	Measurement data for 16 Items	(float, lower 2 bytes)	Urms1
4302	34302	10CD		H	display, item number = 1 (page 1)	(float, upper 2 bytes)	
4303	34303	10CE	16Items_2	L	Measurement data for 16 Items	(float, lower 2 bytes)	Irms1
4304	34304	10CF		H	display, item number = 2 (page 1)	(float, upper 2 bytes)	
4305	34305	10D0	16Items_3	L	Measurement data for 16 Items	(float, lower 2 bytes)	P1
4306	34306	10D1		H	display, item number = 3 (page 1)	(float, upper 2 bytes)	
4307	34307	10D2	16Items_4	L	Measurement data for 16 Items	(float, lower 2 bytes)	S1
4308	34308	10D3		H	display, item number = 4 (page 1)	(float, upper 2 bytes)	
4309	34309	10D4	16Items_5	L	Measurement data for 16 Items	(float, lower 2 bytes)	Q1
4310	34310	10D5		H	display, item number = 5 (page 1)	(float, upper 2 bytes)	
4311	34311	10D6	16Items_6	L	Measurement data for 16 Items	(float, lower 2 bytes)	λ1
4312	34312	10D7		H	display, item number = 6 (page 1)	(float, upper 2 bytes)	
4313	34313	10D8	16Items_7	L	Measurement data for 16 Items	(float, lower 2 bytes)	φ1
4314	34314	10D9		H	display, item number = 7 (page 1)	(float, upper 2 bytes)	
4315	34315	10DA	16Items_8	L	Measurement data for 16 Items	(float, lower 2 bytes)	Pc1
4316	34316	10DB		H	display, item number = 8 (page 1)	(float, upper 2 bytes)	
4317	34317	10DC	16Items_9	L	Measurement data for 16 Items	(float, lower 2 bytes)	fU1
4318	34318	10DD		H	display, item number = 9 (page 1)	(float, upper 2 bytes)	
4319	34319	10DE	16Items_10	L	Measurement data for 16 Items	(float, lower 2 bytes)	fl1
4320	34320	10DF		H	display, item number = 10	(float, upper 2 bytes)	
4321	34321	10E0	16Items_11	L	Measurement data for 16 Items	(float, lower 2 bytes)	U+pk1
4322	34322	10E1		H	display, item number = 11 (page 1)	(float, upper 2 bytes)	
4323	34323	10E2	16Items_12	L	Measurement data for 16 Items	(float, lower 2 bytes)	U-pk1
4324	34324	10E3		H	display, item number = 12 (page 1)	(float, upper 2 bytes)	
4325	34325	10E4	16Items_13	L	Measurement data for 16 Items	(float, lower 2 bytes)	I+pk1
4326	34326	10E5		H	display, item number = 13 (page 1)	(float, upper 2 bytes)	
4327	34327	10E6	16Items_14	L	Measurement data for 16 Items	(float, lower 2 bytes)	I-pk1
4328	34328	10E7		H	display, item number = 14 (page 1)	(float, upper 2 bytes)	
4329	34329	10E8	16Items_15	L	Measurement data for 16 Items	(float, lower 2 bytes)	CfU1
4330	34330	10E9		H	display, item number = 15 (page 1)	(float, upper 2 bytes)	

7.3 Register Functions and Uses

Reg No.	Ref No.	H No.	Register name		Register content		Remarks
4331	34331	10EA	16Items_16	L	Measurement data for 16 Items	(float, lower 2 bytes)	Cfl1
4332	34332	10EB		H	display, item number = 16 (page 1)	(float, upper 2 bytes)	
4333	34333	10EC	16Items_17	L	Measurement data for 16 Items	(float, lower 2 bytes)	Urms2
4334	34334	10ED		H	display, item number = 17 (page 2)	(float, upper 2 bytes)	
to							
4683	34683	124A	16Items_192	L	Measurement data for 16 Items	(float, lower 2 bytes)	
4684	34684	124B		H	display, item number = 192 (page 12)	(float, upper 2 bytes)	

Measurement data corresponding to numeric data display items (Matrix display)

5001+(4×(X-1)+(C-1))×2			Matrix_X_C	L	Measurement data for item number = X, column = C	(float, lower 2 bytes)	Default value (for the 4 element model)
5001+(4×(X-1)+(C-1))×2+1				H		(float, upper 2 bytes)	
5001	35001	1388	Matrix_1_1	L	Measurement data for item 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	Measurement data for item 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	Measurement data for item 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	Measurement data for item number = 1, column = 4 (page 1)	(float, lower 2 bytes)	Urms4
5008	35008	138F		H		(float, upper 2 bytes)	
5009	35009	1390	Matrix_2_1	L	Measurement data for item number = 2, column = 1	(float, lower 2 bytes)	Irms1
5010	35010	1391		H		(float, upper 2 bytes)	
5011	35011	1392	Matrix_2_2	L	Measurement data for item number = 2, column = 2	(float, lower 2 bytes)	Irms2
5012	35012	1393		H		(float, upper 2 bytes)	
5013	35013	1394	Matrix_2_3	L	Measurement data for item number = 2, column = 3	(float, lower 2 bytes)	Irms3
5014	35014	1395		H		(float, upper 2 bytes)	
5015	35015	1396	Matrix_2_4	L	Measurement data for item number = 2, column = 4	(float, lower 2 bytes)	Irms4
5016	35016	1397		H		(float, upper 2 bytes)	
5017	35017	1398	Matrix_3_1	L	Measurement data for item number = 3, column = 1 (page 1)	(float, lower 2 bytes)	P1
5018	35018	1399		H		(float, upper 2 bytes)	
to							
5647	35647	160E	Matrix_81_4	L	Measurement data for item number = 81, column = 4 (page 9)	(float, lower 2 bytes)	
5648	35648	160F		H		(float, upper 2 bytes)	

Peak-over status (input register: 0003)

This data consists of peak-over information for each element assigned to bits as shown below. The bit corresponding to an input where a peak-over condition occurs is set to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								I4	U4	I3	U3	I2	U2	I1	U1

Floating-point data

IEEE single-precision floating-point data. No auxiliary units are used.

Data under abnormal conditions is as follows.

- When no data exists (display: "-----")
"NAN" 0x7FC00000
- When overrange (display: "--OL-"), calculation overflow (display: "--OF-"), or an error (display: "Error") occurs:
"INF" 0x7F800000

Register map (hold registers)

Reg No.	Ref No.	H No.	Register name	Register content		Valid range	Default value	BackUp	R/W
Control data									
0001	40001	0000	Register Hold	Register value retention/ release	(uint 16)	0: release, 1: hold	0	No	R/W
0002	40002	0001							
0003	40003	0002	INTEG:START/ STOP	Start/stop integration (all elements)	(uint 16)	0: stop, 1: start	0	No	R/W
0004	40004	0003	INTEG:RESET	Reset integrated values (all elements)	(uint 16)	1: reset integrated values, other than 1: invalid	—	No	W
0005	40005	0004	INTEG:INDEP	Element-specific (independent) integration ON/OFF	(uint 16)	0: OFF, 1: ON	0	Yes	R/W
0006	40006	0005	INTEG:START	Start integration* (target elements)	(uint 16)	1 to 15 (target elements)	—	No	W
0007	40007	0006	INTEG:STOP	Stop integration* (target elements)	(uint 16)	1 to 15 (target elements)	—	No	W
0008	40008	0007	INTEG:RESET	Reset integrated values* (target elements)	(uint 16)	1 to 15 (target elements)	—	No	W
0009	40009	0008	HOLD	Measurement hold ON/ OFF	(uint 16)	0: OFF, 1: ON	0	Yes	R/W
0010	40010	0009	SINGLE (*TRG)	Execute single measurement	(uint 16)	1: SINGLE execution, other than 1: invalid	—	No	W

* Use when element-specific (independent) integration is ON. It is invalid when OFF.

Specifying elements for integration control (hold registers: 0006, 0007, 0008)

The elements are assigned to bits as shown below.

Set the bit for the element to be controlled for integration to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												EL4	EL3	EL2	EL1

Appendix 1 Error Messages

This section explains communication error messages.

- If the instrument needs servicing, contact your nearest YOKOGAWA dealer to have it repaired.
 - This section lists only communication error messages. For details on other error messages, see the User's Manual (IM WT1501-01EN).
 - Communication syntax errors 100 to 199
 - Communication execution errors 200 to 299
 - System errors (communication) 300 to 399
 - Communication query error 400 to 499
 - Warning (communication) 50
 - Other errors 350
- } Listed below

Note

Error message display language

- When operating the instrument from an external monitor or Web server
Messages can be displayed in the language set below, but they are displayed in English when read by a PC or other device.
- **Setup** menu icon  > **Utility** > **System Configuration** > **Language** > **Message Language**
- On the instrument's LCD screen, messages are displayed in English.

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 mnemonic.	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

Appendix 1 Error Messages

Code	Message	Corrective action	Page
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>.	
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 syntax 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 DATA> is incorrect. 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.	

System Errors (Communication) (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 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?.	–

Warnings (communication) (50)

Code	Message	Corrective action	Page
50	*OPC/? exists in message.	Write *OPC or *OPC? at the end of program messages.	–

Other errors (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 Function

The history function stores a history of the latest measured numeric data for up to the past 40 updates.

Note

The history function applies only to numeric data in normal measurement (Normal) mode.

Overview of the history function

- Maximum number of history entries: 40
After the number of history entries reaches the maximum, if new update data is stored in the history, the oldest update data is removed from the history.
- History data can be output over communication.
- History communication output sends numeric data for multiple updates together.

History communication output

When you execute the history communication numeric data query (`:HISTory:NUMeric:VALue?`) or the history communication numeric list data query (`:HISTory:NUMeric:LIST:VALue?`), the instrument outputs the specified number of numeric data updates starting from the history output start position, according to the history communication output mode. For details, see the later “Explanation.”

History communication output mode

The range of history to be output over communication and the control method for the status register (bit 5: HST) are fixed to the following **N update (N Update)** mode.

When measurement data updating is complete and the status register (bit 5: HST) changes from 1 to 0, the output start position for history communication is set.

N update (N Update)

After measurement starts, the status register (bit 5: HST) changes from 1 to 0 after the specified N data updates.

- The range of updated numeric data to be transferred is the specified N updates, counted backward from the notification time.
N: Up to 20
- Setting example
This is useful mainly when communicating a large amount of numeric data and the update interval is short, so data cannot be transferred for each measurement update. Communication efficiency is improved by transferring one second of numeric data together every second. When the update interval is 50 ms, one second (20 updates) of numeric data is transferred together every second.

For details, see the later “Explanation.”

Explanation

N update (N Update) mode

Update completion is reported every N measurement data updates. Set the number of updates N in advance. In this mode, the numeric data for the automatically reported history portion is output together from the history.

[Operation]

The status register (bit 5: HST) is controlled as follows.

- When the status of the HST status flag (HST) is 1, it is “updating.” It changes to 1 when reception of the first measurement data starts.
- When the status of the HST status flag (HST) is 0, it is “not updating.” It changes to 0 when the Nth measurement data update from the first measurement data is complete.

History communication output count

In the history communication output mode “N update (N Update),” numeric data is reported every N updates, and N updates are output together. Set the number of updates N.

* This cannot be changed during integration.

Setting range: 1 to 20

Set the number of history data points when history communication output is performed continuously using :HISTory:NUMeric:VALue? and :HISTory:NUMeric:LIST:VALue?.

When the history communication output mode is N update (N Update), specify the number of measurement updates N used to report “update complete” with the status register (bit 5: HST).

Status register (bit 5: HST)

HST is cleared/reset when measurement is paused, including when settings are changed or measurement hold is turned on/off.

[Operation]

- If Single measurement is executed while measurement hold is ON, HST is not cleared. HST becomes “updating” (1) at the first Single measurement update, and after the Nth Single measurement update, it repeatedly indicates “update complete.”
- If Single measurement is executed while measurement hold is OFF, HST is cleared.

History communication output

This works only in normal measurement (Normal) mode. If the following query is executed in another mode, “no output” = “#10” (0 bytes of <Block data>) is output.

[Operation]

According to the history communication output mode, the numeric data in the scheduled history communication output range at the time history communication output is executed is captured and output. Therefore, even if the history is updated during communication output, the previously described “numeric data in the scheduled history communication output range (data in units of one update cycle)” that has started being output once is not overwritten (updated) midway and is output to the end.

However, when history communication hold is ON, the numeric data captured when hold was turned ON is output.

Note

For example, after measurement has been stopped and then restarted, such as after the range is changed, until the first history output data is updated, the history output data is “no output” = “#10” (0 bytes of <Block data>).

History communication numeric data query

(:HISTory:NUMeric:VALue?)

- The same ITEMS as :NUMeric[:NORMal]:VALue? are output together for multiple updates according to the history communication output mode.
- The following commands are common to :NUMeric[:NORMal]:VALue? and :HISTory:NUMeric:VALue?.

:NUMeric:BYTeorder	Sets or queries the output byte order of numeric data (FLOAT format)
:NUMeric[:NORMal]:CLear	Clears numeric data output items
:NUMeric[:NORMal]:DElete	Deletes numeric data output items
:NUMeric[:NORMal]:ITEM<x>	Sets or queries numeric data output items
:NUMeric[:NORMal]:NUMber	Sets or queries the number of numeric data items to be transmitted
:NUMeric[:NORMal]:PRESet	Presets numeric data output items to a predefined pattern

History communication numeric list data query

(:HISTory:NUMeric:LIST:VALue?)

- This command outputs the same ITEMS as :NUMeric:LIST:VALue? together for multiple updates according to the history communication output mode.
- The following commands are common to :NUMeric:LIST:VALue? and :HISTory:NUMeric:LIST:VALue?.

:NUMeric:BYTeorder	Sets or queries the output byte order of numeric data (FLOAT format)
:NUMeric:LIST:CLear	Clears harmonic measurement numeric list data output items
:NUMeric:LIST:DElete	Deletes harmonic measurement numeric list data output items
:NUMeric:LIST:ITEM<x>	Sets or queries harmonic measurement numeric list data output items
:NUMeric:LIST:NUMBER	Sets or queries the number of numeric list data items to be transmitted
:NUMeric:LIST:ORder	Sets or queries the highest harmonic order to output for harmonic measurement numeric list data
:NUMeric:LIST:PRESet	Presets harmonic measurement numeric list data output items to a predefined pattern
:NUMeric:LIST:SElect	Sets or queries harmonic measurement numeric list data output components

Extended event register (EESR)

Bit 5: HST of the extended event register is used for update notifications for history communication output.

History communication numeric data output format

FLOAT only (fixed).

History communication hold

According to the history communication output mode, this captures history communication output data for the specified count starting from the start position at the time history communication hold is turned ON.

Appendix 3 Port Numbers

This section explains the port numbers used for communication through the instrument's Ethernet interface and, when necessary, the port numbers available on the 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	Dynamic	TCP	This instrument→PC
HiSLIP	4880	TCP	See section 1.2.
RawSocket	10002	TCP	See section 1.2.
Modbus/TCP	502	TCP	See section 7.1.

Port numbers available on the instrument

The following table lists service names commonly used by measuring instruments and generally known port numbers.

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

TCP and UDP use port numbers 1 to 65535 to identify destination applications. A single application can also use multiple port numbers. Some port numbers are assigned to specific application services as “well-known port numbers” (well-known ports).

Index

Numerics

	Page
1P2W.....	5-83
1P3W.....	5-83
3P3W.....	5-83
3P3W(3V3A).....	5-83
3P3W(3V3A)→3P4W.....	5-83
3P3W(3V3AR).....	5-83
3P3W(3V3AR)→3P4W.....	5-83
3P4W.....	5-83
3P4W→3P3W(3V3AS).....	5-83

A

	Page
abbreviated form.....	4-7
amplitude correction.....	5-61
amplitude correction ratio.....	5-61
anti-aliasing filter.....	5-72
apparent power formula.....	5-93
attenuation constant.....	5-87
auto current range.....	5-64
auto harmonic filter.....	5-73
auto update mode.....	5-127
auto voltage range.....	5-80
averaging.....	5-87
averaging coefficient.....	5-87
averaging type.....	5-87

B

	Page
block data.....	4-10
Boolean.....	4-10

C

	Page
Cal [Execute].....	5-131
calibration.....	5-131
CAN data output.....	5-26
carrier ZC detection hysteresis.....	5-97
character data.....	4-10
character notations.....	iv
commands.....	4-4
common command header.....	4-4
common Z Phase.....	5-95
condition register.....	5-114
continuous integration mode.....	5-84
Corrected Power.....	5-93
cosine signal.....	5-98
crest factor.....	5-61
CSV conversion.....	5-115
CT-ID.....	5-69
CT preset.....	5-65
CT ratio.....	5-78
current.....	4-9
current mode for current integration.....	5-84
current probe power supply.....	5-119
current probe preset.....	5-67
current range.....	5-68
current sensor connection status.....	5-120

D

	Page
Data.....	4-8
data update interval.....	5-127
date and time.....	5-119

DBC files.....	5-26
DC offset correction.....	5-62
deadlock.....	4-3
decimal point type.....	5-120
delay mechanism.....	5-124
digital line filter.....	5-73
display unit.....	5-42
domain number.....	5-124

E

	Page
efficiency.....	5-87
efficiency formula.....	5-87
electrical angle.....	5-94
electrical angle, automatic offset insertion.....	5-94
electrical angle averaging.....	5-97
electrical angle, offset.....	5-94
element.....	5-120
error messages.....	App-1
error queue.....	6-8
Ethernet interface.....	1-2
event-synchronized storage mode.....	5-118
excitation signal.....	5-97
extended event enable register.....	5-114
extended event register.....	5-114, 6-7
external current sensor range.....	5-66

F

	Page
file list filter.....	5-50
files, saving.....	5-52
frequency.....	4-9
frequency filter.....	5-71
frequency (phase correction computation).....	5-63
front panel.....	1-1, 3-1
function options.....	5-45

G

	Page
Greenwich Mean Time.....	5-119
grid intensity.....	5-121
group.....	4-5

H

	Page
harmonic measurement source channel.....	5-54
history communication numeric data.....	5-57, App-6
history communication numeric list data.....	5-57, App-7
history communication output.....	5-56, App-4, App-6
history communication output count.....	5-56, App-5
history communication output mode.....	5-56, App-4
hold.....	5-58

I

	Page
IEEE 1588 time synchronization.....	5-124
image data color tone.....	5-59
image data, saving.....	5-59
independent element integration.....	5-84
initialization.....	5-132
input element types.....	5-74
input filter.....	5-95
input resistance.....	5-65
integrated value, resetting.....	5-85
integration.....	5-84

Index

integration auto calibration	5-84
integration mode	5-84
integration, starting	5-85
integration, stopping	5-86
integration-synchronized storage mode	5-118
integration time	5-85
integration timer	5-86

J	Page
jump destination range	5-66, 5-81

K	Page
k and K	iv
key lock	5-121

L	Page
line filter	5-72, 5-95
linkage	5-98
low-pass filter	5-71

M	Page
manual storage mode	5-118
MAX HOLD	5-92
measurement range	5-70
measurement range linkage	5-77
menu language	5-121
message	4-1
Modbus/TCP	7-1
model	5-131
model code	5-122
models	5-122
motor evaluation function	5-94
motor output	5-96
motor output computation, scaling factor	5-96
moving average count	5-87
multi-unit synchronization mode	5-123

N	Page
network layer	5-125
noise filter	5-96
normal integration mode	5-84
NRF	4-10
NULL	5-76
number of motor poles	5-96
numeric data	5-104
numeric data, format	5-104
numeric data, output item	5-107
numeric data, storing at the start of storage	5-118
NUPDate	5-56
N update mode	5-56, App-4, App-5

O	Page
option	5-132
output queue	6-8
output voltage rate	5-68
overlap commands	4-11

P	Page
Pc	5-93
peak over-range	5-77
phase correction	5-63, 5-64
phase correction value	5-63
phase difference between I/O	5-63
phase difference display format	5-93

PLL source	5-55
polarity of phase difference	5-93
power coefficient	5-78
program messages	4-1
pulse integration mode	5-84

Q	Page
query	4-1

R	Page
range Σ link	5-77
RawSocket	1-2
real-time continuous integration mode	5-84
real-time normal integration mode	5-84
real-time storage mode	5-118
rear panel	1-1, 3-1
Register	4-9
remote and local modes, switching	1-2, 2-2, 3-2
resolver operation	5-97
resolver pole-pair count	5-97
response	4-7
response messages	4-2
response without headers	4-7
revolution signal input type	5-100
rotating speed	5-99
rounding to zero	5-123

S	Page
sample rate	5-130
scaling	5-78
scaling coefficient (torque computation)	5-103
scaling factor	5-98
scheduled storage time	5-118
screen, display color	5-121
self-test	5-133
sensor model	5-69
sensor output type	5-69
serial number	5-124
serial polling	5-114
service request enable register	5-132
sine signal	5-98
Single [Execute]	5-132
single measurement	5-132
Single-phase three-wire	5-83
Single-phase two-wire system	5-83
single-shot storage mode	5-118
SNTP	5-119
Speed	5-99
standard event enable register	5-131
standard event register	5-131, 6-5
status byte	6-3
status byte register	5-132
status reports	5-114, 6-1
storage	5-115
storage interval	5-116
storage mode	5-118
stored items	5-116
stored items, selection method	5-117
string data	4-10
suffix	5-124
suffix code	5-124
synchronization source	5-78
synchronized measurement mode	5-93
synchronized operation status	5-123
SyncSp	5-101
sync speed	5-101

T	Page
TCP/IP	1-6
THD	5-55
three-phase four-wire	5-83
three-phase three-wire system	5-83
three-voltage three-current measurement method	5-83
time	4-9
time difference between input and output	5-63
timeout period	1-6
timestamp	5-118
time zone	5-125
TMCTL	ii
torque	5-101
Torque	5-101
torque signal, input type	5-103
total harmonic distortion	5-55
touch lock	5-124
tracking filter	5-98
transition filter	5-114, 6-7
trigger level	5-127
trigger mode	5-127
trigger position	5-130
trigger slope	5-127
trigger source	5-128
trigger update mode	5-127
U	Page
UDP data output	5-126
unbalance factor	5-93
update mode	5-127
upper-level query	4-6
USB cable	3-3
USB hub	3-3
USB interface	3-2
USB keyboard type	5-125
user-defined event, expression	5-89
user-defined event, expression type	5-91
user-defined event, name	5-91
user-defined events	5-89
user-defined function, equation	5-91
user-defined function, name	5-91
user-defined functions	5-91
V	Page
valid electric current range	5-65
valid voltage range	5-81
voltage	4-9
voltage range	5-82
VT ratio	5-78
W	Page
watt-hour integration method for each polarity, computation method	5-86
waveform display data	5-129
waveform display data, display format	5-129
wiring pattern	5-83
Z	Page
zero-level compensation	5-131