WT3000 Manual, Vol 2/3

WT3000
Precision Power Analyzer Communication Interface
USER'S MANUAL

YOKOGAWA
Yokogawa Meters & Instruments Corporation

IM 760301-17E
5th Edition

Thank you for purchasing the WT3000 Precision Power Analyzer.

This Communication Interface User's Manual describes the functions of the GP-IB, RS-232, USB, and Ethernet interfaces and communication commands. To ensure correct use, please read this manual thoroughly before beginning operation. After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation.

Three manuals, including this one, are provided as manuals for the WT3000. Please read all of them.

Manual Title	Manual No.	Description
WT3000 Precision Power Analyzer User's Manual (Vol 1/3)	IM 760301-01E	Explains all functions and procedures of the WT3000 excluding the expansion functions and communication functions.
WT3000 Precision Power Analyzer Communication Interface User's Manual (CD-ROM) (Vol 2/3)	IM 760301-17E	This manual. Explains the functions for controlling the WT3000 using communication commands.
WT3000 Precision Power Analyzer Expansion Function User's Manual (Vol 3/3)	IM 760301-51E	Explains the expansion functions (motor evaluation function and options) of the WT3000 and their operating procedures.

Note

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions. 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 Electric Corporation is strictly prohibited.
- The TCP/IP software of this product and the document concerning the TCP/IP software have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from University of California.

USB Interface and Ethernet Interface

- The items below are needed on the PC to use the communication functions via the USB interface.
  - DL/WT series library (TMCTL)
  - USB connection device driver between the PC and WT3000
- The item below is needed on the PC to use the communication functions via the Ethernet interface.
  - DL/WT series library (TMCTL)

The library and driver above can be downloaded from the following Web page.

http://www.yokogawa.com/tm/

5th Edition : March 2006 (YK)

Trademarks

- Microsoft, Internet Explorer, MS-DOS, Windows, Windows NT, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated.
- For purposes of this manual, the TM and ® symbols do not accompany their respective trademark names or registered trademark names.
- Other company and product names are trademarks or registered trademarks of their respective holders.

Revisions

• 1st Edition	December 2004
• 2nd Edition	June 2005
• 3rd Edition	January 2006
• 4th Edition	December 2006
• 5th Edition	March 2007

Structure of the Manual

This User's Manual consists of the following sections:

Chapter 1 GP-IB Interface
Describes the functions and specifications of the GP-IB interface.

Chapter 2 RS-232 Interface (Option)
Describes the functions and specifications of the RS-232 interface.

Chapter 3 USB Interface (Option)
Describes the functions and specifications of the USB interface.

Chapter 4 Ethernet Interface (Option)
Describes the functions and specifications of the Ethernet interface.

Chapter 5 Before Programming
Describes the syntax used to transmit commands.

Chapter 6 Communication Commands
Describes all the commands one by one.

Chapter 7 Status Reports
Describes the status byte, various registers, queues, and other information.

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

Index

Conventions Used in This Manual

Unit and Note

Type	Symbol	Meaning	
Unit	k K	1000 1024	Example: 100 kHz Example: 640 KB (file data size)
Note	***Note***	Calls attention to information that is important for proper operation of the instrument.	

Subheadings

On pages that describe operating procedures, the following symbols, displayed characters, and terminology are used to distinguish the procedures from their explanations:

Procedure Follow the numbered steps. All procedures are written with inexperienced users in mind; experienced users may not need to carry out all the steps.

Explanation This subsection describes the setup parameters and the limitations on the procedures.

Displayed Characters and Terminology Used in the Procedural Explanations

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

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

Symbols Used in the Syntax

Symbols which are used in the syntax descriptions in Chapter 6 are shown below. These symbols are referred to as BNF (Backus-Naur Form) symbols. For details on the data, see pages 5-5 and 5-6.

Symbol	Meaning	Example	Example of Input
`< >`	Defined value.	`ELEMent<x>  <x> = 1 to 4`	`->ELEMENT2`
`{ }` |	Select from values given in `{ }`. Exclusive OR	`MODE:{RMS|MEAN|DC|RMEAN}?`	`->MODE:RMS?`
`[ ]`	Can be omitted.	`NUMeric[:NORMal]:VALue?`	`->NUMERIC:VALUE?`

Contents

Chapter 7 Status Reports

Appendix

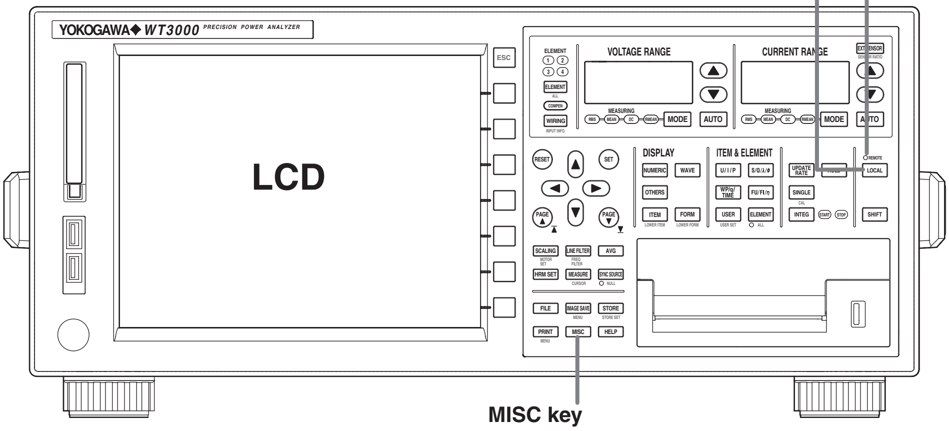
Index

1.1 Names and Functions of Parts

Front Panel

REMOTE indicator
Turns ON when the WT3000 is in the remote mode (controlled via communications).

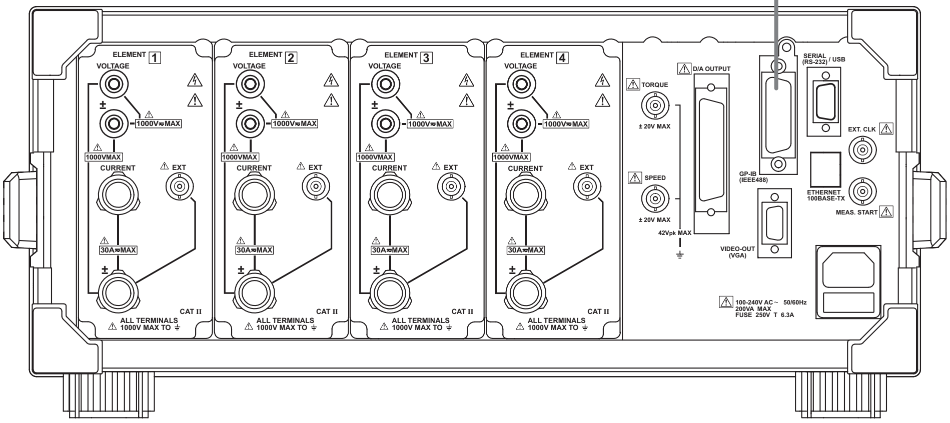
LOCAL key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.



MISC key
Press this key to configure communications.

Rear Panel

GP-IB connector
Connector used to connect the WT3000 to the controller (PC) using a GP-IB cable.



1.2 GP-IB Interface Functions

GP-IB Interface Functions

Listener Capability

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

Talker Capability

- Outputs setup parameters, measured/computed data, waveform data, and other information.

Note

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

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

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

- The REMOTE indicator is turned ON.
- All keys except the LOCAL key are disabled.
- Settings entered in local mode are retained even when the WT3000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing LOCAL key in remote mode puts the instrument in local mode. However, this act is invalid if the instrument has been set to Local Lockout mode (see page 1-6) by the controller.

- The REMOTE indicator is turned OFF.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the WT3000 switches to local mode.

Note

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

1.3 GP-IB Interface Specifications

GP-IB Interface Specifications

Supported device:	National Instruments • AT-GPIB • PCI-GPIB and PCI-GPIB+ • PCMCIA-GPIB and PCMCIA-GPIB + NI-488.2M driver version 1.60 or later
Electrical and mechanical specifications:	Conforms to IEEE St'd 488-1978
Functional specifications:	See table below.
Protocol:	Conforms to IEEE St'd 488.2-1992
Code used:	ISO (ASCII) code
Mode:	Addressable mode
Address setting:	The address can be set in the range from 0 to 30.
Clear remote mode:	Remote mode can be cleared by pressing LOCAL key except when the instrument has been set to Local Lockout mode by the controller.

Functional specifications

Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability.
Acceptor handshaking	AH1	Full acceptor handshaking capability.
Talker	T6	Basic talker capability, serial polling, untalk on MLA (My Listen Address), and no talk-only capability.
Listener	L4	Basic listener capability, unlisten on MTA (My Talk Address), and no listen-only capability.
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel polling	PP0	No parallel polling capability
Device clear	DC1	Full device clear capability
Device trigger	DT1	Full device trigger capability
Controller	C0	No controller capability
Electrical characteristics	E1	Open collector

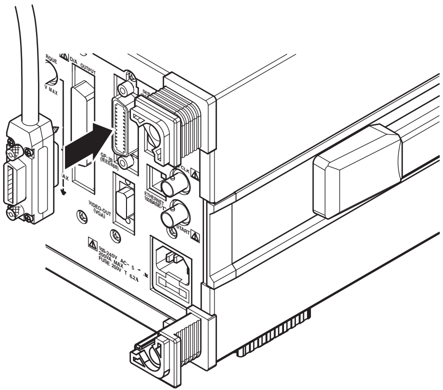
1.4 Connecting the GP-IB Cable

GP-IB Cable

The GP-IB connector used on this instrument is a 24-pin connector that conforms to the IEEE St'd 488-1978. Use a GP-IB cable that conforms to this standard.

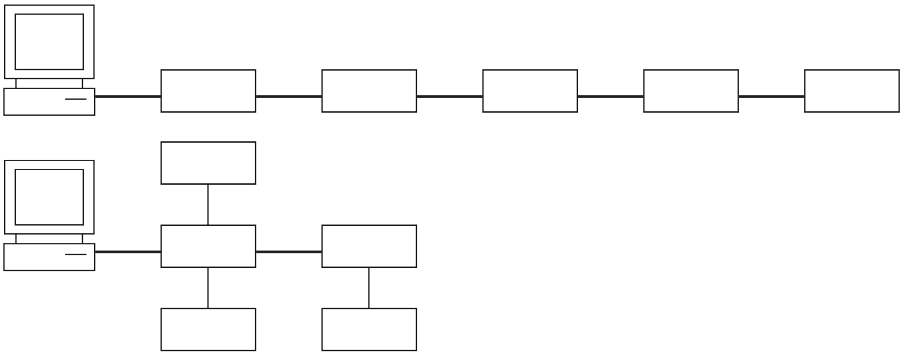
Connection Procedure

Connect the cable as shown below.



Precautions to Be Taken When Making Connections

- Firmly tighten the screws on the GP-IB cable connector.
- Use an NI (National Instruments) model GP-IB port (or card) on the PC side. For details, see section 1.3.
- If a converter is used along the communication cable connecting the WT and PC (for example, a GP-IB-to-USB converter), malfunctions can occur. For details, consult with your Yokogawa dealer or representative.
- Multiple cables can be used to connect multiple devices. However, no more than 15 devices including the controller can be connected on a single bus.
- When connecting multiple devices, each device must have its own unique address.
- Use a cable of length 2 m or less for connecting the devices.
- Make sure the total cable length does not exceed 20 m.
- When communicating, have at least two-thirds of the devices turned ON.
- When connecting multiple devices, connect them in a star or linear configuration (see the figure below). Loop and parallel configurations are not allowed.

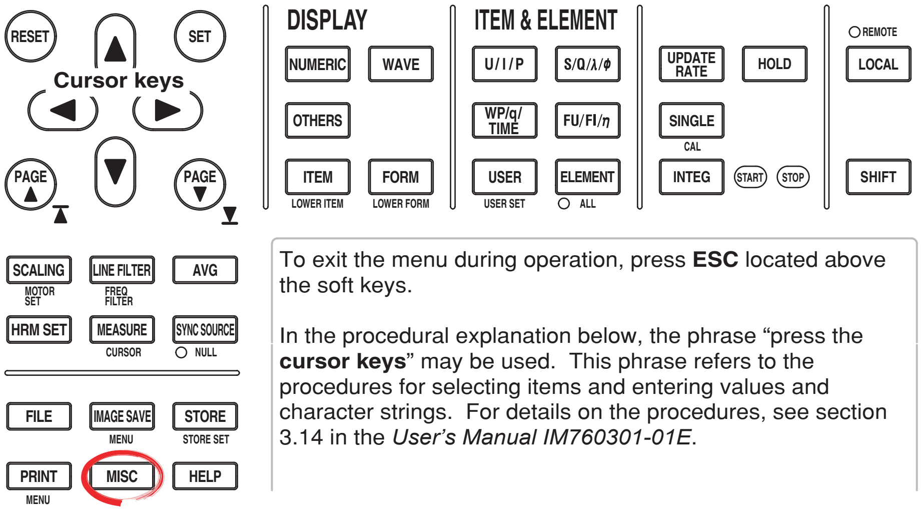


CAUTION

When connecting or disconnecting communication cables, make sure to turn OFF the PC and the WT3000. Otherwise, erroneous operation or damage to the internal circuitry may result.

1.5 Setting the GP-IB Control

Procedure

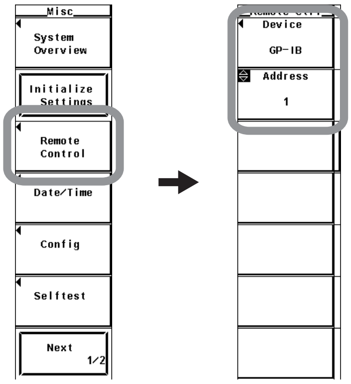


To exit the menu during operation, press **ESC** located above the soft keys.

In the procedural explanation below, the phrase “press the **cursor keys**” may be used. This phrase refers to the procedures for selecting items and entering values and character strings. For details on the procedures, see section 3.14 in the *User’s Manual IM760301-01E*.

1. Press **MISC** to display the Misc menu.
2. Press the **Remote Control** soft key to display the Remote Ctrl menu.
3. Press the **Device** soft key to select GP-IB.

   Only the communication interface selected here is enabled. The WT3000 does not accept commands that are transmitted to other unselected communication interfaces.
4. Press the **cursor keys** to set the address.



Explanation

Enter the following settings when using a controller to set information that can be specified through key operation on the WT3000 or when outputting setup parameters or output waveform display data to the controller.

Address

Set the address of the WT3000 within the following range for the addressable mode.
0 to 30

Each device that can be connected via GP-IB has a unique address within the GP-IB system. This address is used to distinguish the device from others. Therefore, when you connect the WT3000 to a PC, for example, make sure to assign a unique address to the WT3000.

Note

Do not change the address while the controller or other devices are using the GP-IB system.

1.6 Responses to Interface Messages

Responses to Interface Messages

Responses to a Uni-Line Message

- **IFC (Interface Clear)**
  Clears the talker and listener functions. Stops output if data are being output.

- **REN (Remote Enable)**
  Switches between the remote and local modes.

  IDY (Identify) is not supported.

Responses to a Multi-Line Message (Address Command)

- **GTL (Go To Local)**
  Switches to the local mode.

- **SDC (Selected Device Clear)**
  - Clears the program message (command) being received and the output queue (see page 7-9).
  - *OPC and *OPC? commands in execution are void.
  - The *WAI and COMMunicate:WAIT commands are immediately terminated.

- **GET (Group Execute Trigger)**
  Same operation as the *TRG command.

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

Responses to a Multi-Line Message (Universal Command)

- **LLO (Local Lockout)**
  Disables LOCAL on the front panel to prohibit switching to the local mode.

- **DCL (Device Clear)**
  Same operation as the SDC message.

- **SPE (Serial Poll Enable)**
  Sets the talker function on all devices on the bus to serial polling mode. The controller polls the devices in order.

- **SPD (Serial Poll Disable)**
  Clears the serial polling mode of the talker function on all devices on the bus.

  PPU (Parallel Poll Unconfigure) is not supported.

What Is an Interface Message

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

Uni-Line Messages

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

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

Multi-Line Messages

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

- **Address Commands**

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

  - Commands that are valid on an instrument that is designated as a listener
    - GTL (Go To Local)
    - SDC (Selected Device Clear)
    - PPC (Parallel Poll Configure)
    - GET (Group Execute Trigger)
  - Commands that are valid on an instrument that is designated as a talker

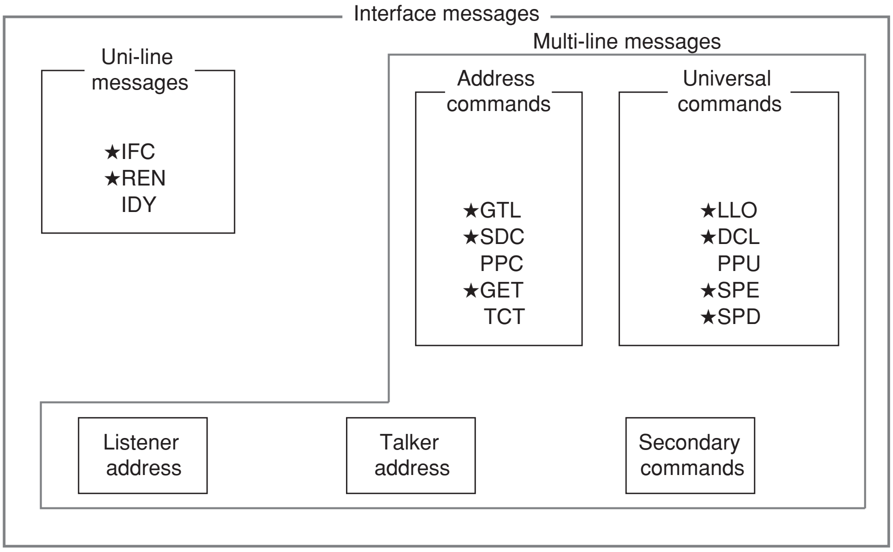
    TCT (Take Control)

- **Universal Commands**

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

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

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



Interface messages that WT3000 supports are indicated with ★ marks.

Note

The differences between SDC and DCL

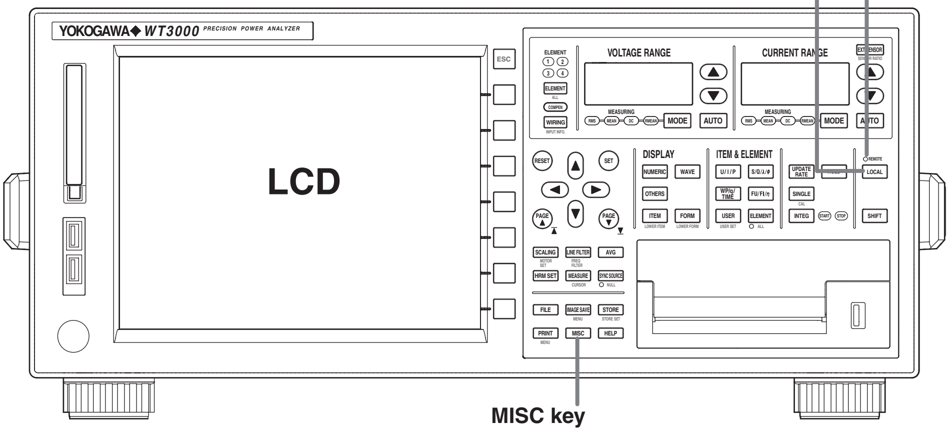
In multi-line messages, SDC messages are those that require talker or listener designation and DCL messages are those that do not require the designation. Therefore, SDC messages are directed at a particular instrument while DCL messages are directed at all instruments on the bus.

2.1 Names and Functions of Parts

Front Panel

REMOTE indicator
Turns ON when the WT3000 is in the remote mode (controlled via communications).

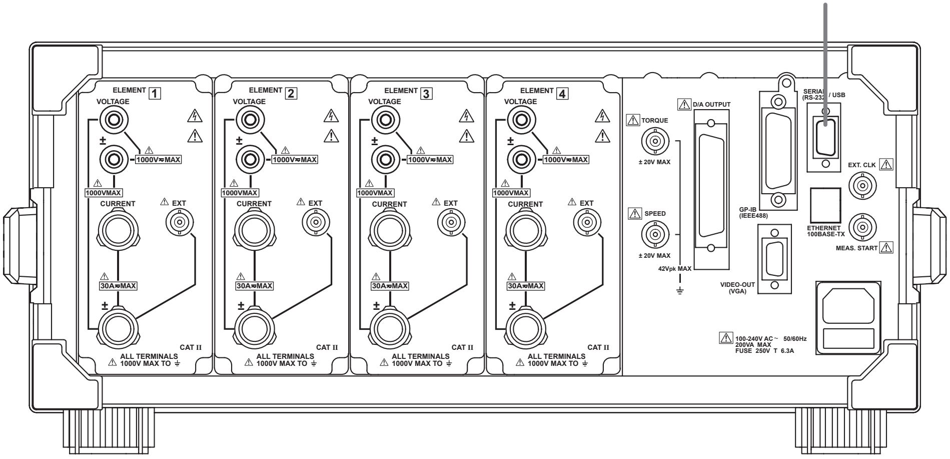
LOCAL key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.



MISC key
Press this key to configure communications.

Rear Panel

RS-232 connector
EIA-574 compliant: For the 9-pin EIA-232 (RS-232)
Connector used to connect the WT3000 to the controller (PC) using an RS-232 interface cable.



2.2 RS-232 Interface Functions and Specifications

Receiving Function

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

Sending Function

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

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

If the WT3000 receives a "`:COMMunicate:REMote ON`" command from the PC when it is in the local mode, it switches to the remote mode.

- The REMOTE indicator is turned ON.
- All keys except the LOCAL key are disabled.
- Settings entered in local mode are retained even when the WT3000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing LOCAL key in remote mode puts the instrument in local mode. However, this is void when the WT3000 has received a ":`COMMunicate:LOCKout ON`" command from the PC (local lockout condition). When the WT3000 receives a "`:COMMunicate:REMote OFF`" command from the PC, the WT3000 switches to the local mode regardless of the local lockout condition.

- The REMOTE indicator is turned OFF.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the WT3000 switches to local mode.

Note

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

RS-232 Interface Specifications

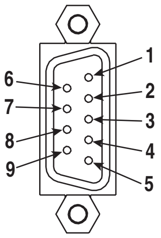
Electrical characteristics:	Conforms to EIA-574 (9-pin EIA-232 (RS-232))
Connection:	Point-to-point
Transmission mode:	Full-duplex
Synchronization:	Start-stop synchronization
Baud rate:	1200, 2400, 4800, 9600, 19200, and 38400
Start bit:	Fixed to 1 bit
Data length:	7 or 8 bits
Parity:	Even, odd, or no parity
Stop bit:	1 or 2 bits
Connector:	DELC-J9PAF-13L6 (JAE or equivalent)
Hardware handshaking:	Select whether to fix the CA and CB signals to TRUE or use the signals for flow control.
Software handshaking:	Select whether to use the X-ON and X-OFF signals to control the transmission data or both transmission and reception data. X-ON (ASCII 11H) X-OFF (ASCII 13H)
Received buffer length:	256 bytes

2.3 Connection via the RS-232 Interface

When you connect the WT3000 to a PC, you must set the WT3000 so that the handshaking method, baud rate, data format, and other parameters match those on the PC side.

For details on the settings, see the following pages. In addition, use an interface cable that meets the specifications of the WT3000.

Connector and Signal Names



2 RD (Received Data): Received data from the PC.
Signal direction ..... input

3 SD (Send Data): Transmitted data to the PC.
Signal direction ..... output

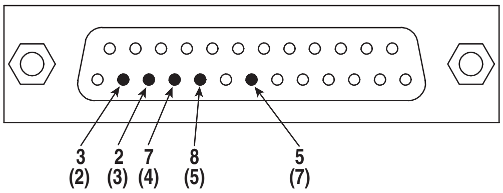
5 SG (Signal Ground): Signal ground.

7 RS (Request to Send): Handshaking signal to receive data from the PC.
Signal direction ..... output

8 CS (Clear to Send): Handshaking signal to send data to the PC.
Signal direction ..... input

* Pins 1, 4, 6, and 9 are not used.

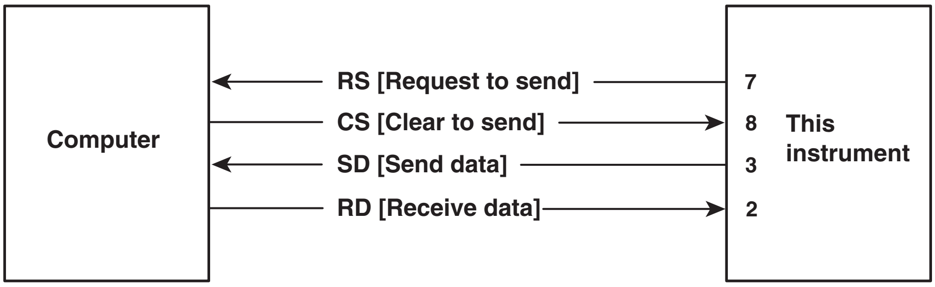
9-Pin to 25-Pin Adapter and Signal Names



The numbers inside the parentheses are pin numbers for the 25-pin connector.

Signal Direction

The following figure shows the direction of the signals used by the RS-232 interface of the WT3000.



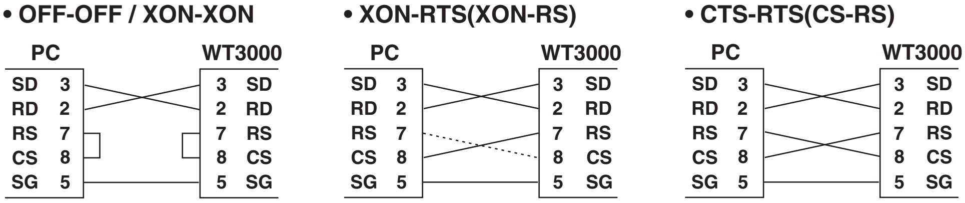
RS-232 Standard Signals and Their JIS and CCITT Abbreviations

Pin No. (9-pin connector)	Abbreviation			Description
	Serial (RS-232)	CCITT	JIS	
5	AB (GND)	102	SG	Signal ground
3	BA (TXD)	103	SD	Transmitted data
2	BB (RXD)	104	RD	Received data
7	CA (RTS)	105	RS	Request to send
8	CB (CTS)	106	CS	Clear to send

Signal Wiring Example

The pin numbers are for the 9-pin connector.

In general, use a cross cable.



2.4 Combination of Handshaking Methods

When using the RS-232 interface for transferring data, it is necessary for equipment on both sides to agree on a set of rules to ensure the proper transfer of data. The set of rules is called handshaking. Because there are various handshaking methods that can be used between the WT3000 and the PC, one must make sure that the same method is chosen by both the WT3000 and the PC.

You can choose any of the four methods in the table below.

Handshake format Descriptions→○

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

OFF-OFF

Data Transmission Control

There is no handshaking between the WT3000 and the PC. The “X-OFF” and “X-ON” signals are treated as data, and the CS signal is ignored.

Data Reception Control

There is no handshaking between the WT3000 and the PC. When the received buffer becomes full, all overflow data are discarded.
RS = True (fixed).

XON-XON

Data Transmission Control

Software handshaking is performed between the WT3000 and the PC. When an “X-OFF” code is received while sending data to the PC, the WT3000 stops the data transmission. When the WT3000 receives the next “X-ON” code, the WT3000 resumes the data transmission. The CS signal received from the PC is ignored.

Data Reception Control

Software handshaking is performed between the WT3000 and the PC. When the free area of the receive buffer decreases to 64 bytes, the WT3000 sends an “X-OFF” code. When the free area increases to 192 bytes, it sends an “X-ON” code.
RS = True (fixed).

XON-RS

Data Transmission Control

Software handshaking is performed between the WT3000 and the PC. When an “X-OFF” code is received while sending data to the PC, the WT3000 stops the data transmission. When the WT3000 receives the next “X-ON” code, the WT3000 resumes the data transmission. The CS signal received from the PC is ignored.

Data Reception Control

Hardware handshaking is performed between the WT3000 and the PC. When the free area of the receive buffer decreases to 64 bytes, the instrument sets “RS = False.” When the free area increases to 192 bytes, it sets “RS = True.”

CS-RS

Data Transmission Control

Hardware handshaking is performed between the WT3000 and the PC. When the CS signal becomes False while sending data to the PC, the WT3000 stops the data transmission. When the CS signal becomes True, the WT3000 resumes the data transmission. The “X-OFF” and “X-ON” signals are treated as data.

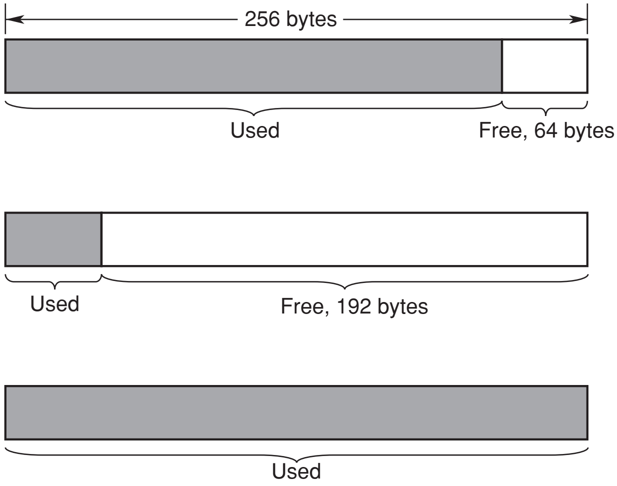
Data Reception Control

Hardware handshaking is performed between the WT3000 and the PC. When the free area of the receive buffer decreases to 64 bytes, the instrument sets “RS = False.” When the free area increases to 192 bytes, it sets “RS = True.”

Precautions Regarding Data Reception Control

When handshaking is used to control the reception of data, data may still be sent from the PC even if the free space in the receive buffer drops below 64 bytes. In this case, after the receive buffer becomes full, the excess data will be lost, whether or not handshaking is in effect. Data storage of data resumes when there is free space in the buffer.

Data Receiving Control using Handshaking



When handshaking is in use, reception of data will stop when the free space in the buffer drops to 64 bytes since data cannot be passed to the main program fast enough to keep up with the transmission.

After reception of data stops, data continues to be passed to the internal program. Reception of data starts again when the free space in the buffer increases to 192 bytes.

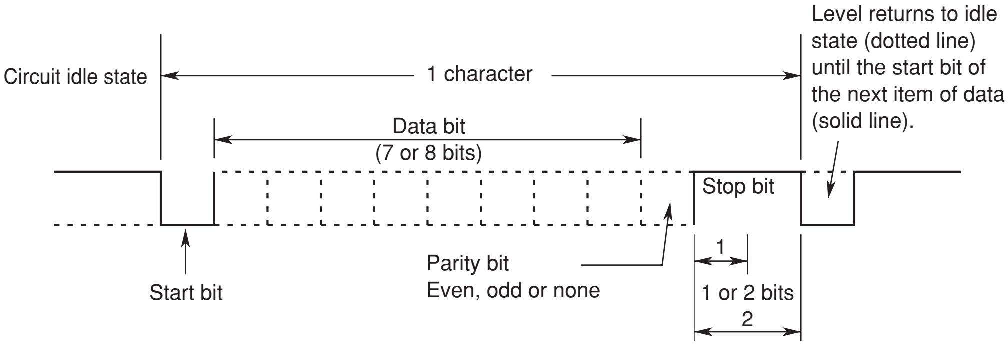
Whether handshaking is in use or not, if the buffer becomes full, any additional data received is no longer stored and is lost.

Note

The PC program must be designed so that the received buffers of both the WT3000 and the PC do not become full.

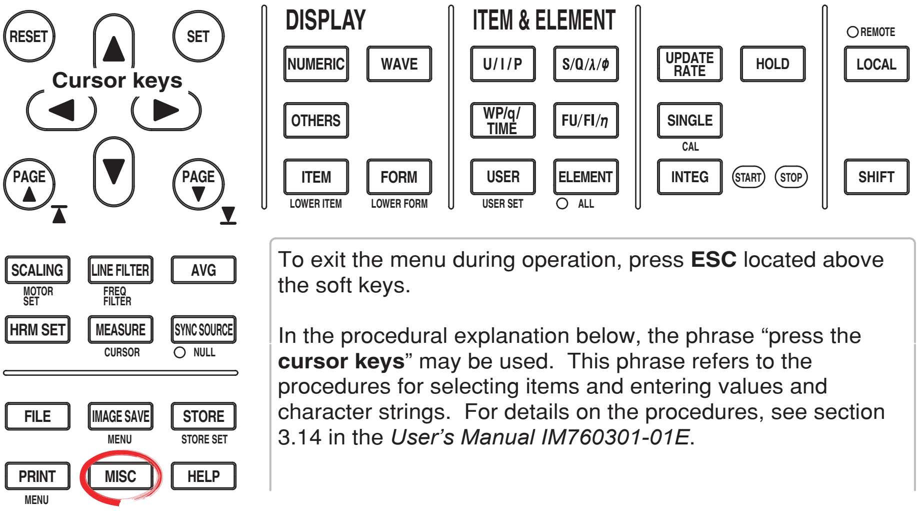
2.5 Combination of Data Formats

The RS-232 interface of the WT3000 performs communications using start-stop synchronization. In start-stop synchronization, characters are transmitted one at a time. Each character consists of a start bit, data bits, a parity bit, and a stop bit (see the following figure).



2.6 Setting the RS-232 Control

Procedure



To exit the menu during operation, press **ESC** located above the soft keys.

In the procedural explanation below, the phrase “press the **cursor keys**” may be used. This phrase refers to the procedures for selecting items and entering values and character strings. For details on the procedures, see section 3.14 in the *User’s Manual IM760301-01E*.

Selecting the RS-232 Control

1. Press **MISC** to display the Misc menu.
2. Press the **Remote Control** soft key to display the Remote Ctrl menu.
3. Press the **Device** soft key to select RS232.

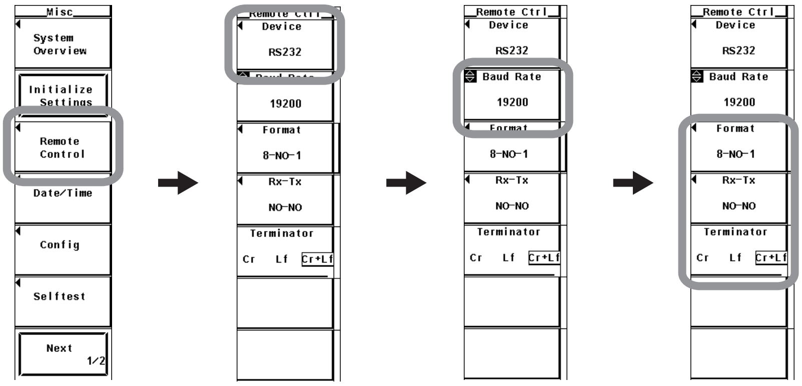
   Only the communication interface selected here is enabled. The WT3000 does not accept commands that are transmitted to other unselected communication interfaces.

Selecting the Baud Rate

4. Press the **cursor keys** to select Baud Rate.

Selecting the Data Format, Handshaking, and Terminator

4. Press the **Format** (data format), **Rx-Tx** (handshaking), or **Terminator** soft key and select the setting for the respective item.



Explanation

Enter the following settings when using a controller to set information that can be specified through key operation on the WT3000 or when outputting setup parameters or output waveform data to the controller.

Baud Rate

Select the baud rate from the following:
1200, 2400, 4800, 9600, 19200, or 38400

Data Format

Select the combination of data length, parity, and stop bit from the following:
8-NO-1, 7-EVEN-1, 7-ODD-1, or 7-NO-2

Handshaking Method

Select the transmit data control and receive data control from the following:
NO-NO, XON-XON, XON-RTS, or CTS-RTS

Terminator

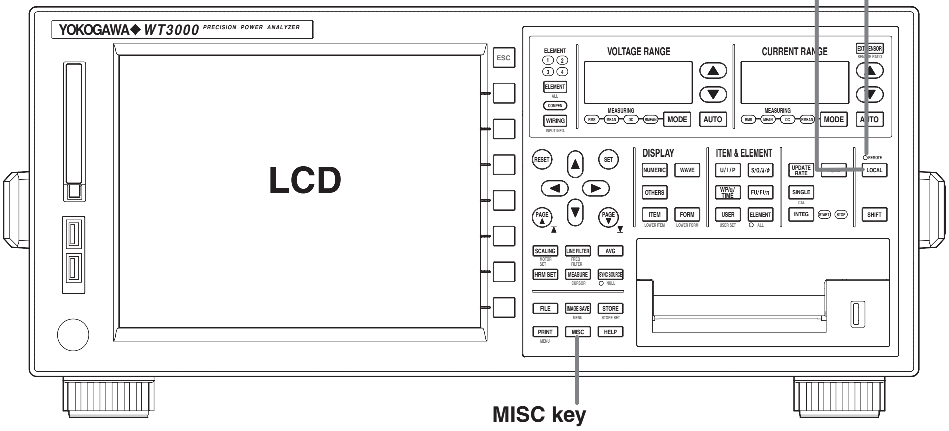
Select the terminator from below. On the WT3000 menu, select the terminator that is used when transmitting data from the WT3000. Use “Lf” or “Cr+Lf” for the terminator for receiving data on the WT3000.
Cr, Lf, or Cr+Lf

3.1 Names of Parts

Front Panel

REMOTE indicator
Turns ON when the WT3000 is in the remote mode (controlled via communications).

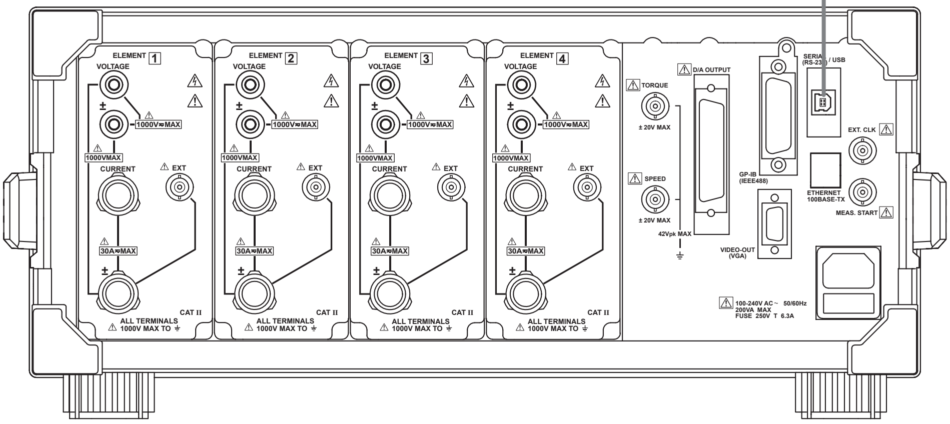
LOCAL key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.



MISC key
Press this key to configure communications.

Rear Panel

USB connector for connecting to a PC
A connector used to connect the WT3000 to the controller (such as a PC) using a USB cable. For the connection procedure, see page 3-3.



3.2 USB Interface Functions and Specifications

Reception Function

You can specify the same settings as those specified by front panel key operations.

Receives output requests for measured and computed data, setup data, and error codes.

Transmission Function

Outputs measured and computed data.

Outputs panel setup parameters and the status byte.

Outputs error codes that have occurred.

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

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

- The REMOTE indicator is turned ON.
- All keys except the LOCAL key are disabled.
- Settings entered in local mode are retained even when the WT3000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing the LOCAL key when the WT3000 is in the remote mode causes the instrument to switch to the local mode. However, this is not possible when the `:COMMunicate:LOCKout ON` command is received from the PC while Local Lockout mode is active. Local mode is activated when the `:COMMunicate:REMote OFF` command is received regardless of Local Lockout.

- The REMOTE indicator is turned OFF.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the WT3000 switches to local mode.

Note

The USB interface cannot be used simultaneously with other communication interfaces (GP-IB, RS-232, and Ethernet).

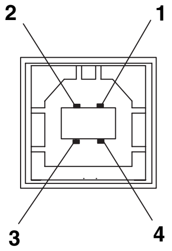
USB Interface Specifications

Electrical and mechanical specifications: Conforms to USB Rev.1.1

Connector:	Type B connector (receptacle)
Number of ports:	1
Power supply:	Self-powered
PC system supported:	A controller such as a PC running Windows 98 SE, Windows Me, Windows 2000, or Windows XP that is equipped with a USB port as standard (a separate device driver is required for the connection with a PC)

3.3 Connection via the USB Interface

Connector and Signal Names



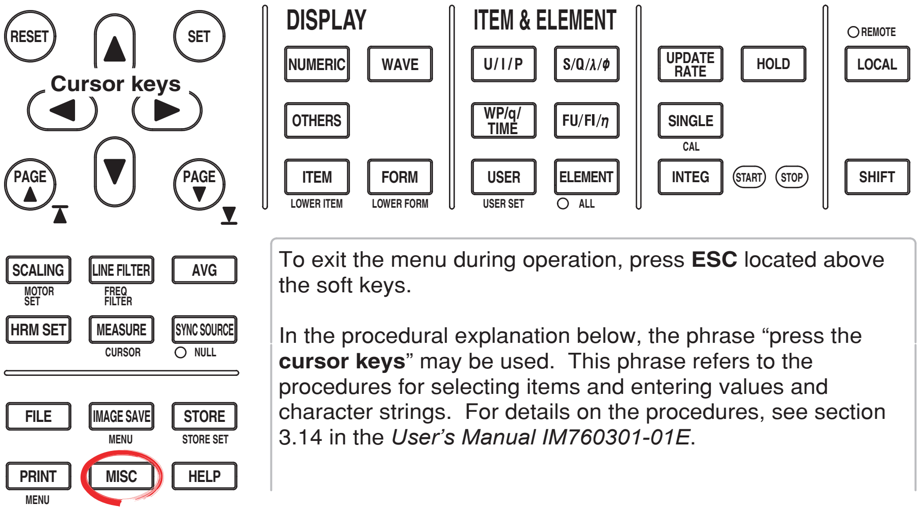
Pin No.	Signal Name	
1	V_{BUS}:	+5 V
2	D–:	–Data
3	D+:	+Data
4	GND:	Ground

Precautions to Be Taken When Making Connections

- Connect the USB cable by inserting the connector firmly into the USB connector.
- When connecting multiple devices using USB hubs, connect the WT3000 to the USB hub that is closest to the controller.
- Do not connect or disconnect the USB cable after the power is turned ON until the WT3000 boots up completely (until the WT3000 is ready for operation, approximately 20 to 30 s). If you do, the WT3000 may malfunction.

3.4 Setting the USB Control

Procedure



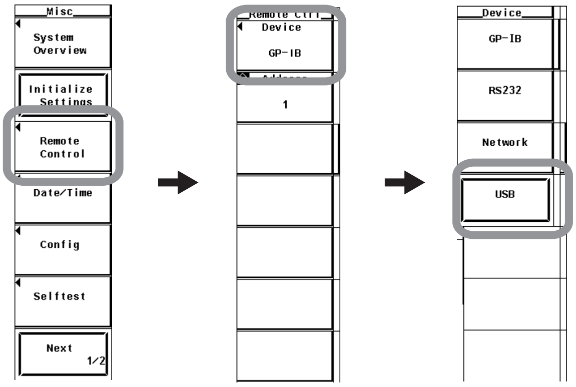
To exit the menu during operation, press **ESC** located above the soft keys.

In the procedural explanation below, the phrase “press the **cursor keys**” may be used. This phrase refers to the procedures for selecting items and entering values and character strings. For details on the procedures, see section 3.14 in the *User’s Manual IM760301-01E*.

Selecting USB Control

1. Press **MISC** to display the Misc menu.
2. Press the **Remote Control** soft key to display the Remote Ctrl menu.
3. Press the **Device** soft key and select USB.

   Only the communication interface selected here is enabled. The WT3000 does not accept commands that are transmitted to other unselected communication interfaces.
4. Press the **cursor keys** to set the ID value.



Explanation

You can control the WT3000 from a PC using the USB interface. YOKOGAWA's dedicated USB connection device driver and library software (TMCTL) must be installed on the PC in addition to entering the settings described above.

Setting the ID Value

Set the ID value of the WT3000 within the following range.

1 to 127

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

Note

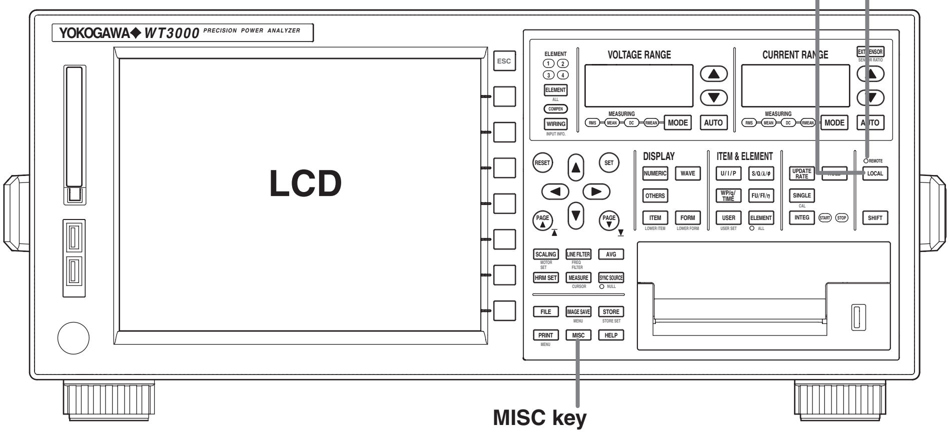
Do not change the ID value while communicating via the USB interface.

4.1 Names and Functions of Parts

Front Panel

REMOTE indicator
Turns ON when the WT3000 is in the remote mode (controlled via communications).

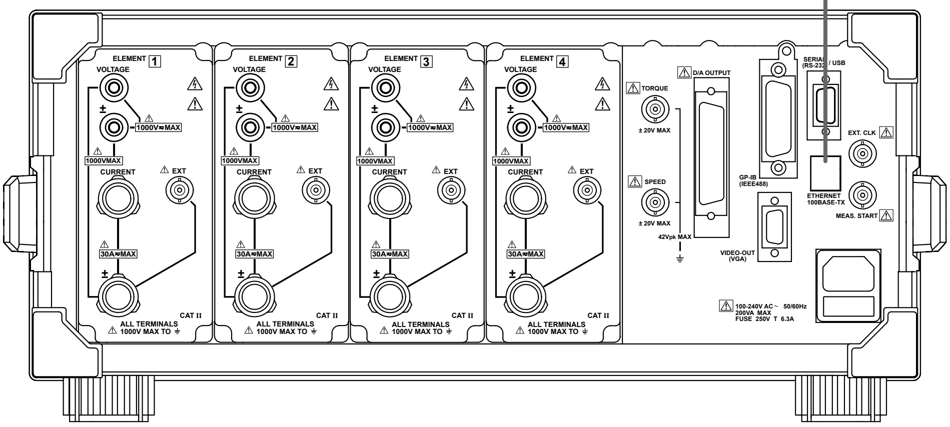
LOCAL key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.



MISC key
Press this key to configure communications.

Rear Panel

Ethernet port (100BASE-TX)
Port used to connect to the controller (PC) via the network or by a one-to-one connection. For the connection procedure, see section 5.1 in the *Expansion Function User's Manual IM760301-51E*.



4.2 Ethernet Interface Functions and Specifications

You can control the WT3000 from a PC using the Ethernet interface. Details about specific functions and how to enter settings are provided below.

Receiving Function

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

Sending Function

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

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

If the WT3000 receives a "`:COMMunicate:REMote ON`" command from the PC when it is in the local mode, it switches to the remote mode.

- The REMOTE indicator is turned ON.
- All keys except the LOCAL key are disabled.
- Settings entered in local mode are retained even when the WT3000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing LOCAL key in remote mode puts the instrument in local mode. However, this is void when the WT3000 has received a "`:COMMunicate:LOCKout ON`" command from the PC (local lockout condition). When the WT3000 receives a "`:COMMunicate:REMote OFF`" command from the PC, the WT3000 switches to the local mode regardless of the local lockout condition.

- The REMOTE indicator is turned OFF.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the WT3000 switches to local mode.

Note

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

Ethernet Interface Specifications

Electrical and mechanical specifications:	Conforms to IEEE 802.3.
Number of simultaneous connections:	1
Port number:	10001/tcp

For details on other specifications, see chapter 5 in the *Expansion Function User's Manual IM760301-51E*.

User Authentication Function

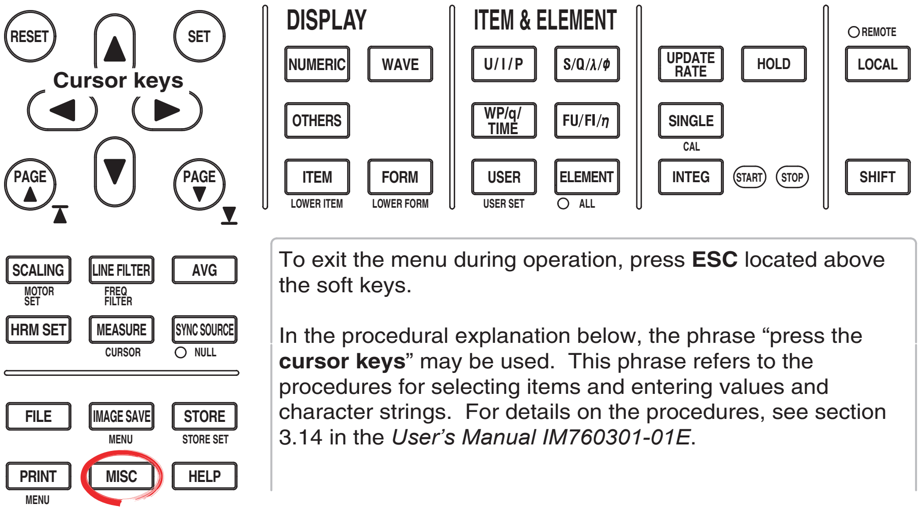
You must enter the user name and password to access the WT3000 from a PC using the Ethernet interface. The user name and password for accessing the WT3000 can be specified in the User Account screen under the Misc menu. For details, see “Setting the Ethernet Control” in the next section.

Connecting the WT3000 and the PC

For the procedure of connecting the WT3000 to a PC, see section 5.1 in the *Expansion Function User’s Manual IM760301-51E*.

4.3 Setting the Ethernet Control

Procedure



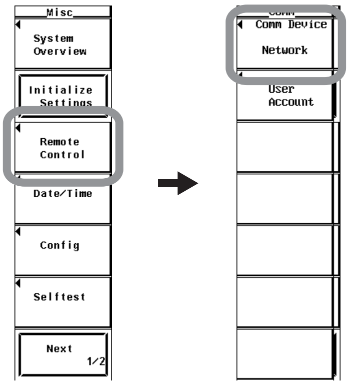
To exit the menu during operation, press **ESC** located above the soft keys.

In the procedural explanation below, the phrase “press the **cursor keys**” may be used. This phrase refers to the procedures for selecting items and entering values and character strings. For details on the procedures, see section 3.14 in the *User’s Manual IM760301-01E*.

Selecting the Ethernet Control

1. Press **MISC** to display the Misc menu.
2. Press the **Remote Control** soft key to display the Remote Ctrl menu.
3. Press the **Device** soft key and select Network.

   Only the communication interface selected here is enabled. The WT3000 does not accept commands that are transmitted to other unselected communication interfaces.

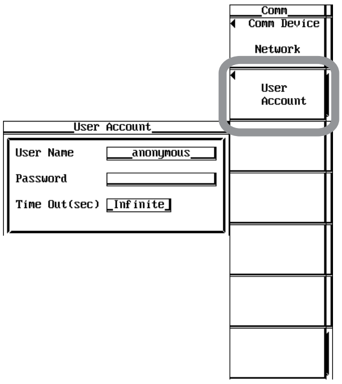


Setting the User account (user name and password)

5. Press the **User Account** soft key to display the User Account dialog box.
6. Press the **cursor keys** to select User Name.
7. Press **SET**. A keyboard appears.
8. Use the **keyboard** to enter the user name.
   For keyboard operations, see section 3.14, “Entering Values and Strings” in the *User’s Manual IM760301-01E*.
9. Press the **cursor keys** to select Password. The password setting is entered twice.
10. Press **SET**. A keyboard appears.
11. Use the **keyboard** to enter the password.
    - A password is not required if the login name is anonymous.
    - For keyboard operations, see section 3.14, “Entering Values and Strings” in the *User’s Manual IM760301-01E*.

Setting the Timeout Value

12. Press the **cursor keys** to select Time Out.
13. Press **SET**. The timeout value entry box appears.
14. Press the **cursor keys** to set the timeout value.
15. Press **SET** or **ESC** to close the box.



User Account

User Name anonymous

Password

Time Out(sec) Infinite

Entering TCP/IP Settings

You must enter TCP/IP settings to control the WT3000 from a PC using the Ethernet interface. For the setup procedure, see section 5.2 in the *Expansion Function User’s Manual IM760301-51E*.

Restarting the Instrument

You must restart the instrument after changing or entering a new setting for the user account, timeout value, or TCP/IP in order for the settings to take effect.

Explanation

You can control the WT3000 from a PC using the Ethernet interface. YOKOGAWA's dedicated library software (TMCTL) must be installed on the PC in addition to entering the settings described above.

Retail Software
WTViewer (760121) Version 4.01 or higher.
A trial version is available for download from the following URL.
http://www.yokogawa.com/tm/760122/

Setting the User Name

- Enter the user name to allow access to the WT3000.
- Enter up to 15 characters.
- The characters that can be used are 0-9, A-Z, %, _, ( ) (parentheses), - (minus sign).
- If you specify anonymous, the WT3000 can be accessed from the outside (PC) without a password.

Setting the Password

- Enter the password for the user name to allow access to the WT3000.
- Enter up to 15 characters.
- The characters that can be used are 0-9, A-Z, %, _, ( ) (parentheses), - (minus sign).
- If the user name is set to anonymous, the WT3000 can be accessed from the outside (PC) without a password.
- The password setting is entered twice.

Setting the Timeout Value

The WT3000 closes the connection to the network if there is no access for a certain period of time (timeout time).
The available settings are 0 to 3600 s, or Infinite. The default value is Infinite.

Note

To apply new settings, the WT3000 must be power cycled.

5.1 Messages

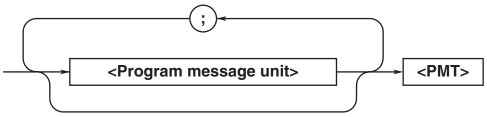
Messages and Queries

Messages are used to exchange information between the controller and the instrument. Messages sent from the controller to the WT3000 are called program messages. Program messages that request a response to be sent from the WT3000 to the controller are called queries. Messages that the controller receives from the WT3000 are called response messages.

If a query is included in a program message, the WT3000 sends a response message after receiving the program message. A single response message is always returned in response to a single program message.

Program Messages

The program message format is shown below.



<Program Message Unit>

A program message consists of one or more program message units; each unit corresponds to one command. The instrument executes the received commands in order.

Each program message unit is separated by a semicolon (;).

For details regarding the format of the program message unit, see the next section.

Example

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

Unit Unit

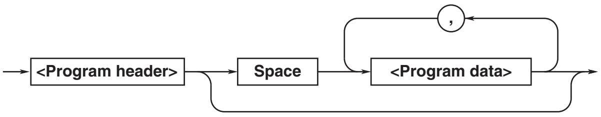
<PMT>

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

- **NL (New Line)**
  Same as LF (Line Feed). ASCII code “0AH.”
- **^END**
  The END message (EOI signal) as defined in the IEEE488.1. (The data byte that is sent with the END message is the last data of the program message.)
- **NL^END**
  NL with an END message attached. (NL is not included in the program message.)

Program Message Unit Format

The program message unit format is shown below.



<Program Header>

The program header indicates the command type. For details, see page 5-3.

<Program Data>

If certain conditions are required in executing a command, program data is added. A space (ASCII code “20H”) separates the program data from the header. If there are multiple sets of program data, they are separated by commas (,).

For details, see page 5-5.

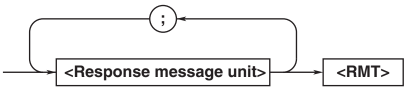
Example

```
:INPut:CFACtor 3<PMT>
```

Header Data

Response Message

The response message format is shown below.



<Response Message Unit>

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

Response message units are separated by a semicolon (;).

For details regarding the format of the response message unit, see the next section.

Example:

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

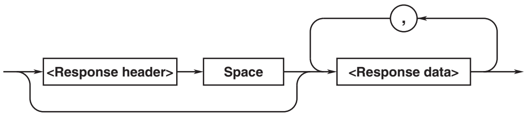
Unit Unit

<RMT>

A response message terminator. It is NL^END.

Response Message Unit Format

The response message unit format is shown below.



<Response Header>

A response header sometimes precedes the response data. A space separates the data from the header. For details, see page 5-4.

<Response Data>

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

Example

$\underbrace{\texttt{100.00E-03}}_{\text{Data}}\texttt{<RMT>} \quad \underbrace{\texttt{:DISPLAY:MODE}}_{\text{Header}}\ \underbrace{\texttt{WAVE}}_{\text{Data}}\texttt{<RMT>}$

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

Precautions to Be Taken when Transferring Messages

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

Deadlock

The instrument can store in its buffer program and response messages of length 1024 bytes or more (The number of available bytes varies depending on the operating conditions). When both the transmit and receive buffers become full at the same time, the instrument can no longer continue its communication operation. This state is called a deadlock. In this case, operation can be resumed by discarding the program message.

Deadlock will not occur if the program message (including the <PMT>) is kept below 1024 bytes. Furthermore, deadlock never occurs if a program message does not contain a query.

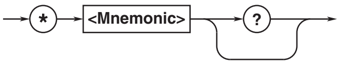
5.2 Commands

Commands

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

Common Command Header

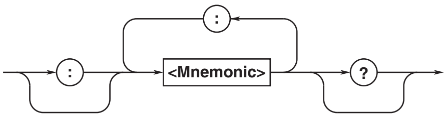
Commands that are defined in the IEEE 488.2-1992 are called common commands. The header format of a common command is shown below. An asterisk (*) is always placed in the beginning of a command.



Common command example `*CLS`

Compound Header

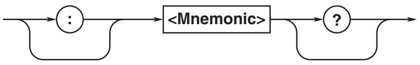
Dedicated commands used by the instrument are classified and arranged in a hierarchy according to their functions. The format of a compound header is shown below. A colon (:) must be used to specify a lower hierarchy.



Compound header example `:DISPlay:MODE`

Simple Header

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



Simple header example `:HOLD`

Note

A <mnemonic> is a character string made up of alphanumeric characters.

When Concatenating Commands

Command Group

A command group is a group of commands that have common compound headers arranged in a hierarchy. A command group may contain sub-groups.

Example Group of commands related to harmonic measurement

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

When Concatenating Commands of the Same Group

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

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

When Concatenating Commands of Different Groups

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

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

When Concatenating Simple Headers

If a simple header follows another command, a colon (:) is placed in front of the simple header.

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

When Concatenating Common Commands

Common commands that are defined in the IEEE 488.2-1992 are independent of hierarchy. Colons (:) are not needed before a common command.

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

When Separating Commands with <PMT>

If a terminator is used to separate two commands, each command is a separate message. Therefore, the common header must be specified for each command even when commands belonging to the same command group are being concatenated.

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

Upper-Level Query

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

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

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

Header Interpretation Rules

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

- Mnemonics are not case sensitive.
  Example "CURSor" can also be written as "cursor" or "CUrsor."
- The lower-case section of the header can be omitted.
  Example "CURSor" can also be written as "CURSO" or "CURS."
- The question mark (?) at the end of a header indicates that it is a query. The question mark (?) cannot be omitted.
  Example: The shortest abbreviation for CURSor? is CURS?.
- If the <x> (value) at the end of a mnemonic is omitted, it is interpreted as a 1.
  Example If "ELEMent<x>" is written as "ELEM," it means "ELEMent1."
- The section enclosed by braces ([]) can be omitted.
  Example "[:INPut]:SCALing[:STATe][:ALL] ON" can be written as "SCAL ON."
  However, the last section enclosed by braces ([]) cannot be omitted in an upper-level query.
  Example "SCALing?" and "SCALing:STATe?" are different queries.

5.3 Responses

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

- **Response Consisting of a Header and Data**
  If the response can be used as a program message without any change, it is returned with a command header attached.

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

- **Response Consisting of Data Only**
  If the response cannot be used as a program message unless changes are made to it (query-only command), only the data section is returned. However, there are query-only commands that return responses with the header attached.

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

When You Wish to Return a Response without a Header

Responses that return both header and data can be set so that only the data section is returned. The "COMMunicate:HEADer" command is used to do this.

Abbreviated Form

Normally, the lower-case section is removed from a response header before the response is returned to the controller. Naturally, the full form of the header can also be used. For this, the "COMMunicate:VERBose" command is used. The sections enclosed by braces ([ ]) are also omitted in the abbreviated form.

5.4 Data

Data

A data section comes after the header. A space must be included between the header and the data. The data contains conditions and values. Data is classified as below.

Data	Description
<Decimal>	A value expressed as a decimal number (Example: VT ratio setting -> `[:INPUt]:SCALing:VT:ELEMent1 100`)
<Voltage><Current> <Time><Frequency>	A physical value (Example: Voltage range setting ->`[:INPUt]:VOLTage:RANGE:ELEMent1 100V`)
<Register>	Register value expressed as binary, octal, decimal or hexadecimal. (Example: Extended event register value ->`:STATUS:EESE #HFE`)
<Character Data>	Predefined character string (mnemonic). Can be selected from { }. (Example: Trigger mode selection ->`:DISPlay:WAVE:TRIGger:MODE {AUTO|NORMal}`)
<Boolean>	Indicates ON and OFF. Set using ON, OFF or a value (Example: Data hold ON ->`:HOLD ON`)
<String data>	An arbitrary character string (Example: User-defined function ->`:MEASure:FUNCtion1:EXPRession "URMS(E1)"`)
<Filename>	Indicates a file name. (Example: Save file name ->`:FILE:SAVE:SETup[:EXECute] "CASE1"`)
<Block data>	Arbitrary 8-bit data (Example: Response to acquired waveform data ->`#40012ABCDEFGHIJKL`)

<Decimal>

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

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

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

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

<Voltage>, <Current>, <Time>, and <Frequency> indicate data that have physical dimensions. <Multiplier> or <Unit> can be attached to the <NRf> form that was described earlier. It is expressed in one of the following forms.

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

<Multiplier>

<Multipliers> which can be used are indicated below.

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

<Unit>

<Units> that can be used are indicated below.

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

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

<Register>

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

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

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

<Character Data>

<Character Data> is a specified string of character data (a mnemonic). It is mainly used to indicate options and is chosen from the character strings given in { }. For interpretation rules, refer to "Header Interpretation Rules" on page 5-4.

Form	Example
{AUTO|NORMal}	AUTO

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

<Boolean>

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

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

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

<String Data>

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

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

- If a character string contains a double quotation mark ("), the double quotation mark is replaced by two concatenated double quotation marks (""). This rule also applies to a single quotation mark within a character string.
- A response message is always enclosed in double quotation marks (").
- <String data> is an arbitrary character string. Therefore the instrument assumes that the remaining program message units are part of the character string if no single (') or double quotation mark (") is encountered. As a result, no error is detected if a quotation mark is omitted.

<Filename>

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

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

- <NRf> is rounded to an 8-digit integer and converted to ASCII code. The result is the file name (example: 1 becomes "00000001"). Negative values are not allowed.
- For <Character data> and <String data>, the first eight characters become the file name.
- Response messages are always returned in the <String data> form.

<Block Data>

<Block data> is arbitrary 8-bit data. It is only used in response messages on the WT3000. The syntax is as follows:

Form	Example
#N<N-digit decimal number><Data byte sequence>	#40012ABCDEFGHIJKL

- #N
  Indicates that the data is <Block data>. "N" indicates the number of succeeding data bytes (digits) in ASCII code characters.
- <N-digit decimal number>
  Indicates the number of bytes of data (example: 0012 = 12 bytes).
- <Data byte sequence>
  Expresses the actual data (example: ABCDEFGHIJKL).
- Data is comprised of 8-bit values (0 to 255). This means that the ASCII code "0AH," which stands for "NL," can also be a code used for data. Hence, care must be taken when programming the controller.

5.5 Synchronization with the Controller

Overlap Commands and Sequential Commands

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

For example, if the next program message is transmitted when specifying the voltage range and querying the result, the response always returns the most recent setting (100 V in this case).

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

This is because the next command is forced to wait until the processing of "`INPut:VOLTage:RANGe:ELEMent1`" itself is completed. This type of command is called a sequential command.

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

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

In this case, "`:INPut:VOLTage:RANGe:ELEMent1?`" is executed before the loading of the file is completed, and the voltage range that is returned is the value before the file is loaded.

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

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

Synchronizing with Overlap Commands

Using the *WAI Command

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

Example

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

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

Because "`*WAI`" is executed immediately before "`:INPut:VOLTage:RANGe:ELEMent1?,`" "`:INPut:VOLTage:RANGE:ELEMent1?`" is not executed until the loading of the file is completed.

Using the COMMunicate:OVERlap command

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

Example

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

"`COMMunicate:OVERlap #HFFBF`" enables overlap operation on commands other than media access.

Because the overlap operation of file loading is disabled, "`FILE:LOAD:SETup`" operates in the same way as a sequential command. Therefore, "`:INPut:VOLTage:RANGe:ELEMent1?`" is not executed until the loading of the file is completed.

Using the *OPC Command

The *OPC command sets the OPC bit, bit 0 of the standard event register (see page 7-5), to 1 when the overlap operation is completed.

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

"COMMunicate:OPSE" is a command used to select the "*OPC" target. Here, media access is specified. "*ESE 1" and "*SRE 32" indicate that a service request is generated only when the OPC bit is 1."*ESR?" clears the standard event register.In the example above, ":INPut:VOLTage:RANGe:ELEMent1?" is not executed until a service request is generated.

Using the *OPC? Query

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

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 used to select the "*OPC?" target. Here, media access is specified.Because "*OPC?" does not generate a response until the overlap operation is completed, the loading of the file will have been completed by the time the response to "*OPC?" is read.

Note

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

Achieving Synchronization without Using Overlap Commands

Even for sequential commands, synchronization is sometimes required to correctly query the measured data. If you wish to query the most recent numeric data on every time measured data is updated, for example, sending the ":NUMeric[:NORMal]:VALue?" command at an arbitrary timing can cause data that is the same as the previous data to be received. This is because the WT3000 returns the current measured data regardless of whether the measured data has been updated since the previous query.In this case, the following method must be used to synchronize with the end of the updating of the measured data.

Using the STATus:CONDition? query

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

Using the Extended Event Register

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

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

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

The ":STATus:EESE 1" command is used to reflect only bit 0 of the extended event register to the status byte.

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

The "*SRE 8" command is used to generate a service request solely on the cause of the extended event register.

The ":NUMeric[:NORMal]:VALue?" command is not executed until a service request is generated.

Using the COMMunicate:WAIT command

The "COMMunicate:WAIT" command halts communications until a specific event is generated.

Example
```
:STATus:FILTer1 FALL;:STATus:
EESR?<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)
```

For a description of "STATus:FILTer1 FALL" and "STATus:EESR?" see the previous section regarding the extended event register.

The "COMMunicate:WAIT 1" command indicates that the program will wait for bit 0 of the extended event register to be set to "1."

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

Chapter 6 Communication Commands

6.1 A List of Commands

Command	Function	Page
ACQuisition Group		
:ACQuisition?	Queries all settings related to the output of the waveform sampling data.	6-17
:ACQuisition:BYTeorder	Sets the output byte order of the waveform sampling data (FLOAT format) that is transmitted by ":ACQuisition:SEND?" or queries the current setting.	6-17
:ACQuisition:END	Sets the output end point of the waveform sampling data that is transmitted by ":ACQuisition:SEND?" or queries the current setting.	6-17
:ACQuisition:FORMat	Sets the format of the waveform sampling data that is transmitted by ":ACQuisition:SEND?" or queries the current setting.	6-17
:ACQuisition:HOLD	Sets whether to hold (ON) or release (OFF) all the waveform sampling data or queries the current setting.	6-17
:ACQuisition:LENGth?	Queries the total number of points of the waveform sampling data specified by ":ACQuisition:TRACe."	6-18
:ACQuisition:SEND?	Queries the waveform sampling data specified by ":ACQuisition:TRACe."	6-18
:ACQuisition:SRATe?	Queries the sampling rate of the retrieved data.	6-18
:ACQuisition:STARt	Sets the output start point of the waveform sampling data that is transmitted by ":ACQuisition:SEND?" or queries the current setting.	6-18
:ACQuisition:TRACe	Sets the target trace of ":ACQuisition:SEND?" or queries the current setting.	6-18
AOUTput Group		
:AOUTput?	Queries all settings related to the D/A output.	6-19
:AOUTput:NORMal?	Queries all settings related to the D/A output.	6-19
:AOUTput[:NORMal]:CHANnel<x>	Sets the D/A output items (function, element, and harmonic order) or queries the current setting.	6-19
:AOUTput[:NORMal]:IRTime	Sets the rated integration time for the D/A output of integrated values or queries the current setting.	6-19
:AOUTput[:NORMal]:MODE<x>	Sets the method of setting the rated value for the D/A output items or queries the current setting.	6-20
:AOUTput[:NORMal]:RATE<x>	Manually sets the rated maximum and minimum values for the D/A output items or queries the current setting.	6-20
CBCycle Group		
:CBCycle?	Queries all settings related to the Cycle by Cycle measurement function.	6-21
:CBCycle:COUNt	Sets the number of cycles for Cycle by Cycle measurement or queries the current setting.	6-21
:CBCycle:DISPlay?	Queries all settings related to the Cycle by Cycle display.	6-21
:CBCycle:DISPlay:CURSor	Sets the cursor position of the Cycle by Cycle display or queries the current setting.	6-21
:CBCycle:DISPlay:ITEM<x>	Sets the displayed items (function and element) of the Cycle by Cycle display or queries the current setting.	6-21
:CBCycle:DISPlay:PAGE	Sets the number of the displayed page of the Cycle by Cycle display or queries the current setting.	6-21
:CBCycle:FILTer?	Queries all settings related to the filter for Cycle by Cycle measurement.	6-22
:CBCycle:FILTer:LINE?	Queries all settings related to the line filter for Cycle by Cycle measurement.	6-22
:CBCycle:FILTer[:LINE][:ALL]	Collectively sets the line filters of all elements for Cycle by Cycle measurement.	6-22
:CBCycle:FILTer[:LINE]:ELEMent<x>	Sets the line filter of individual elements for Cycle by Cycle measurement or queries the current setting.	6-22
:CBCycle:FILTer[:LINE]:MOTor	Sets the motor input line filter for Cycle by Cycle measurement or queries the current setting.	6-22
:CBCycle:RESet	Resets Cycle by Cycle measurement.	6-22
:CBCycle:STARt	Starts Cycle by Cycle measurement.	6-22

Command	Function	Page
:CBCycle:STATe?	Queries the Cycle by Cycle measurement status.	6-22
:CBCycle:SYNChronize?	Queries all settings related to the synchronization source of Cycle by Cycle measurement.	6-22
:CBCycle:SYNChronize:SLOPe	Sets the slope of the synchronization source of Cycle by Cycle measurement or queries the current setting.	6-22
:CBCycle:SYNChronize:SOURce	Sets the synchronization source for Cycle by Cycle measurement or queries the current setting.	6-23
:CBCycle:TIMEout	Sets the timeout value for Cycle by Cycle measurement or queries the current setting.	6-23
:CBCycle:TRIGger?	Queries all settings related to triggers.	6-23
:CBCycle:TRIGger:LEVel	Sets the trigger level or queries the current setting.	6-23
:CBCycle:TRIGger:MODE	Sets the trigger mode or queries the current setting.	6-23
:CBCycle:TRIGger:SLOPe	Sets the trigger slope or queries the current setting.	6-23
:CBCycle:TRIGger:SOURce	Sets the trigger source or queries the current setting.	6-23
COMMunicate Group		
:COMMunicate?	Queries all settings related to communications.	6-24
:COMMunicate:HEADer	Sets whether to add a header to the response to a query or queries the current setting.	6-24
:COMMunicate:LOCKout	Sets or clears local lockout.	6-24
:COMMunicate:OPSE	Sets the overlap command that is to be used by the *OPC, *OPC?, and *WAI commands or queries the current setting.	6-24
:COMMunicate:OPSR?	Queries the operation pending status register.	6-24
:COMMunicate:OVERlap	Sets the commands that will operate as overlap commands or queries the current setting.	6-24
:COMMunicate:REMote	Sets remote or local.	6-24
:COMMunicate:STATus?	Queries line-specific status.	6-25
:COMMunicate:VERBose	Sets the response messages to full form or abbreviated form or queries the current setting.	6-25
:COMMunicate:WAIT	Waits for a specified extended event.	6-25
:COMMunicate:WAIT?	Creates the response that is returned when the specified event occurs.	6-25
CURSor Group		
:CURSor?	Queries all settings related to the cursor measurement.	6-26
:CURSor:BAR?	Queries all settings related to the cursor measurement of the bar graph display.	6-26
:CURSor:BAR:POSition<x>	Sets the cursor position (order) on the bar graph display or queries the current setting.	6-26
:CURSor:BAR[:STATe]	Turns ON/OFF the cursor display on the bar graph or queries the current setting.	6-26
:CURSor:BAR:{Y<x>|DY}?	Queries the cursor measurement value on the bar graph display.	6-26
:CURSor:FFT?	Queries all settings related to the cursor measurement on the FFT waveform display.	6-26
:CURSor:FFT:POSition<x>	Sets the cursor position on the FFT waveform display or queries the current setting.	6-26
:CURSor:FFT[:STATe]	Turns ON/OFF the cursor display on the FFT waveform display or queries the current setting.	6-27
:CURSor:FFT:TRACe<x>	Sets the cursor target on the FFT waveform display or queries the current setting.	6-27
:CURSor:FFT:{X<x>|DX|Y<x>|DY}?	Queries the cursor measurement value on the FFT waveform display.	6-27
:CURSor:TRENd?	Queries all settings related to the cursor measurement of the trend display.	6-27
:CURSor:TRENd:POSition<x>	Sets the cursor position on the trend or queries the current setting.	6-27
:CURSor:TRENd[:STATe]	Turns ON/OFF the cursor display on the trend display or queries the current setting.	6-27
:CURSor:TRENd:TRACe<x>	Sets the cursor target on the trend or queries the current setting.	6-27
:CURSor:TRENd:{X<x>|Y<x>|DY}?	Queries the cursor measurement value on the trend display.	6-27

Command	Function	Page
:CURSor:WAVE?	Queries all settings related to the cursor measurement on the waveform display.	6-28
:CURSor:WAVE:PATH	Sets the cursor path on the waveform display or queries the current setting.	6-28
:CURSor:WAVE:POSition<x>	Sets the cursor position on the waveform display or queries the current setting.	6-28
:CURSor:WAVE[:STATe]	Turns ON/OFF the cursor display on the waveform display or queries the current setting.	6-28
:CURSor:WAVE:TRACe<x>	Sets the cursor target on the waveform display or queries the current setting.	6-28
:CURSor:WAVE:{X<x>|DX|PERDt|Y<x>|DY}?	Queries the cursor measurement value on the waveform display.	6-28
DISPlay Group		
:DISPlay?	Queries all settings related to the screen display.	6-29
:DISPlay:BAR?	Queries all settings related to the bar graph.	6-29
:DISPlay:BAR:FORMat	Sets the display format of the bar graph or queries the current setting.	6-29
:DISPlay:BAR:ITEM<x>	Sets the bar graph item (function and element) or queries the current setting.	6-29
:DISPlay:BAR:ORDer	Sets the start and end orders of the bar graph or queries the current setting.	6-30
:DISPlay:CBCycle?	Queries all settings related to the Cycle by Cycle display.	6-30
:DISPlay:CBCycle:CURSor	Sets the cursor position of the Cycle by Cycle display or queries the current setting.	6-30
:DISPlay:CBCycle:ITEM<x>	Sets the displayed items (function and element) of the Cycle by Cycle display or queries the current setting.	6-30
:DISPlay:CBCycle:PAGE	Sets the number of the displayed page of the Cycle by Cycle display or queries the current setting.	6-30
:DISPlay:FFT?	Queries all settings related to the FFT waveform display.	6-30
:DISPlay:FFT:FFT<x>?	Queries all settings related to the FFT waveform.	6-31
:DISPlay:FFT:FFT<x>:LABel	Sets the label of the FFT waveform or queries the current setting.	6-31
:DISPlay:FFT:FFT<x>:OBJect	Sets the source waveform of the FFT computation or queries the current setting.	6-31
:DISPlay:FFT:FFT<x>[:STATe]	Turns ON/OFF the FFT waveform display or queries the current setting.	6-31
:DISPlay:FFT:FORMat	Sets the display format of the FFT waveform or queries the current setting.	6-31
:DISPlay:FFT:POINt	Sets the number of points of the FFT computation or queries the current setting.	6-31
:DISPlay:FFT:SCOPe	Sets the display range of the FFT waveform or queries the current setting.	6-31
:DISPlay:FFT:SPECtrum	Sets the display spectrum format of the FFT waveform or queries the current setting.	6-32
:DISPlay:FFT:VSCale	Sets the display scale of the vertical axis of the FFT waveform or queries the current setting.	6-32
:DISPlay:FFT:WINDow	Sets the window function of the FFT computation or queries the current setting.	6-32
:DISPlay:FLICker?	Queries all settings related to flicker measurement display.	6-32
:DISPlay:FLICker:ELEMent	Sets the element to be displayed for flicker measurement display or queries the current setting.	6-32
:DISPlay:FLICker:PAGE	Sets the number of the page to be displayed for flicker measurement display or queries the current setting.	6-32
:DISPlay:FLICker:PERiod	Sets the observation period number to be displayed for flicker measurement display or queries the current setting.	6-32
:DISPlay:INFOrmation?	Queries all settings related to the display of the setup parameter list.	6-32
:DISPlay:INFOrmation:PAGE	Sets the page number of the display of setup parameter list or queries the current setting.	6-32
:DISPlay:INFOrmation[:STATe]	Turns ON/OFF the display of the setup parameter list or queries the current setting.	6-32
:DISPlay:MATH?	Queries all settings related to the computed waveform display.	6-33
:DISPlay:MATH:CONStant<x>	Sets the constant to be used in the waveform computing equation or queries the current setting.	6-33
:DISPlay:MATH:MATH<x>?	Queries all settings related to the computed waveform.	6-33

Command	Function	Page
:DISPlay:MATH:MATH<x>:EXPRession	Sets the equation of the waveform computation or queries the current setting.	6-33
:DISPlay:MATH:MATH<x>:LABel	Sets the label of the computed waveform or queries the current setting.	6-33
:DISPlay:MATH:MATH<x>:SCALing?	Queries all settings related to the scaling of the computed waveform.	6-33
:DISPlay:MATH:MATH<x>:SCALing:CENTer	Sets the center value of the manual scaling of the computed waveform or queries the current setting.	6-34
:DISPlay:MATH:MATH<x>:SCALing:MODE	Sets the scaling mode of the computed waveform or queries the current setting.	6-34
:DISPlay:MATH:MATH<x>:SCALing:SDIV	Sets the scale/division value of the manual scaling of the computed waveform or queries the current setting.	6-34
:DISPlay:MATH:MATH<x>:UNIT	Sets the unit to be added to the result of the waveform computation or queries the current setting.	6-34
:DISPlay:MODE	Sets the display format or queries the current setting.	6-35
:DISPlay:NUMeric?	Queries all settings related to the numeric display. (The same as :DISPlay:NUMeric:NORMal?.)	6-35
:DISPlay:NUMeric:NORMal?	Queries all settings related to the numeric display.	6-35
:DISPlay:NUMeric[:NORMal]:ALL?	Queries all settings related to the numeric display (all display).	6-35
:DISPlay:NUMeric[:NORMal]:ALL:CURSor	Sets the cursor position on the numeric display (all display) or queries the current setting.	6-35
:DISPlay:NUMeric[:NORMal]:ALL:ORDer	Sets the displayed harmonic order on the harmonic measurement function display page of the numeric display (all display) or queries the current setting.	6-36
:DISPlay:NUMeric[:NORMal]:ALL:PAGE	Sets the page number on the numeric display (all display) or queries the current setting.	6-36
:DISPlay:NUMeric[:NORMal]:FORMat	Sets the numeric display format or queries the current setting.	6-36
:DISPlay:NUMeric[:NORMal]:LIST?	Queries all settings related to the numeric display (list display).	6-36
:DISPlay:NUMeric[:NORMal]:LIST:CURSor	Sets the cursor position on the numeric display (list display) or queries the current setting.	6-37
:DISPlay:NUMeric[:NORMal]:LIST:HEADer	Sets the cursor position in the header section on the numeric display (list display) or queries the current setting.	6-37
:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x>	Sets the displayed items (function and element) on the numeric display (list display) or queries the current setting.	6-37
:DISPlay:NUMeric[:NORMal]:LIST:ORDer	Sets the harmonic order cursor position of the data section on the numeric display (list display) or queries the current setting.	6-37
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}?	Queries all settings related to the numeric display ({4-value|8-value|16-value} display).	6-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:CURSor	Sets the cursor position on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.	6-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>	Sets the displayed items (function, element, and harmonic order) on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.	6-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PAGE	Sets the page number on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.	6-38

Command	Function	Page
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PRESet	Presets the display order pattern of displayed items on the numeric display ({4-value|8-value|16-value} display).	6-39
:DISPlay:TRENd?	Queries all settings related to the trend.	6-39
:DISPlay:TRENd:ALL	Collectively turns ON/OFF all trends.	6-39
:DISPlay:TRENd:CLEar	Clears the trend.	6-39
:DISPlay:TRENd:FORMat	Sets the display format of the trend or queries the current setting.	6-39
:DISPlay:TRENd:ITEM<x>?	Queries all settings related to the trend.	6-39
:DISPlay:TRENd:ITEM<x>[:FUNCtion]	Sets the trend item (function, element, and harmonic order) or queries the current setting.	6-39
:DISPlay:TRENd:ITEM<x>:SCALing?	Queries all settings related to the scaling of the trend.	6-39
:DISPlay:TRENd:ITEM<x>:SCALing:MODE	Sets the scaling mode of the trend or queries the current setting.	6-40
:DISPlay:TRENd:ITEM<x>:SCALing:VALue	Sets the upper and lower limits of the manual scaling of the trend or queries the current setting.	6-40
:DISPlay:TRENd:TDIV	Sets the horizontal axis (T/div) of the trend or queries the current setting.	6-40
:DISPlay:TRENd:T<x>	Turns ON/OFF the trend or queries the current setting.	6-40
:DISPlay:VECTor?	Queries all settings related to the vector display.	6-40
:DISPlay:VECTor:NUMeric	Turns ON/OFF the numeric data display for the vector display or queries the current setting.	6-40
:DISPlay:VECTor:OBJect	Sets the wiring unit to be displayed during vector display or queries the current setting.	6-40
:DISPlay:VECTor:{UMAG|IMAG}	Sets the zoom factor for the vector display or queries the current setting.	6-41
:DISPlay:WAVE?	Queries all settings related to the waveform display.	6-41
:DISPlay:WAVE:ALL	Collectively turns ON/OFF all waveform displays.	6-41
:DISPlay:WAVE:FORMat	Sets the display format of the waveform or queries the current setting.	6-41
:DISPlay:WAVE:GRATicule	Sets the graticule (grid) type or queries the current setting.	6-41
:DISPlay:WAVE:INTerpolate	Sets the interpolation method of the waveform or queries the current setting.	6-41
:DISPlay:WAVE:MAPPing?	Queries all settings related to the waveform mapping to the split screen.	6-41
:DISPlay:WAVE:MAPPing[:MODE]	Sets the waveform mapping method for the split screen or queries the current setting.	6-41
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|SPEed|TORQue}	Sets the waveform mapping to the split screen or queries the current setting.	6-42
:DISPlay:WAVE:POSition?	Queries all settings related to the vertical position (level of the center position) of the waveform.	6-42
:DISPlay:WAVE:POSition:{UALL|IALL}	Collectively sets the vertical position (level of the center position) of the waveform {voltage|current} of all elements.	6-42
:DISPlay:WAVE:POSition:{U<x>|I<x>}	Sets the vertical position (level of the center position) of the waveform {voltage|current} of the element or queries the current setting.	6-42
:DISPlay:WAVE:SVALue	Turns ON/OFF the scale value display or queries the current setting.	6-42
:DISPlay:WAVE:TDIV	Sets the Time/div value of the waveform or queries the current setting.	6-42
:DISPlay:WAVE:TLABel	Turns ON/OFF the waveform labels or queries the current setting.	6-42
:DISPlay:WAVE:TRIGger?	Queries all settings related to the trigger.	6-42
:DISPlay:WAVE:TRIGger:LEVel	Sets the trigger level or queries the current setting.	6-42
:DISPlay:WAVE:TRIGger:MODE	Sets the trigger mode or queries the current setting.	6-43
:DISPlay:WAVE:TRIGger:SLOPe	Sets the trigger slope or queries the current setting.	6-43
:DISPlay:WAVE:TRIGger:SOURce	Sets the trigger source or queries the current setting.	6-43
:DISPlay:WAVE:{U<x>|I<x>|SPEed|TORQue}	Turns ON/OFF the waveform display or queries the current setting.	6-43
:DISPlay:WAVE:VZoom?	Queries all settings related to the vertical zoom factor of the waveform.	6-43
:DISPlay:WAVE:VZoom:{UALL|IALL}	Collectively sets the vertical zoom factor of the waveform {voltage|current} of all elements.	6-43

Command	Function	Page
`:DISPlay:WAVE:VZoom:{U<x>|I<x>}`	Sets the vertical zoom factor of the waveform {voltage|current} of the element or queries the current setting.	6-43
FILE Group		
`:FILE?`	Queries all settings related to the file operation.	6-47
`:FILE:CDIRectory`	Changes the current directory.	6-47
`:FILE:DELete:IMAGe:{TIFF|BMP|PSCRipt|PNG|JPEG}`	Deletes the screen image data file.	6-47
`:FILE:DELete:NUMeric:{ASCii|FLOat}`	Deletes the numeric data file.	6-47
`:FILE:DELete:SETup`	Deletes the setup parameter file.	6-47
`:FILE:DELete:WAVE:{BINary|ASCii|FLOat}`	Deletes the waveform display data file.	6-47
`:FILE:DRIVe`	Sets the target drive.	6-47
`:FILE:FORMat:EXECute`	Formats the PC card.	6-47
`:FILE:FREE?`	Queries the free space on the target drive.	6-47
`:FILE:LOAD:ABORt`	Aborts the file loading.	6-47
`:FILE:LOAD:SETup`	Loads the setup parameter file.	6-47
`:FILE:MDIRectory`	Creates a directory	6-48
`:FILE:PATH?`	Queries the absolute path of the current directory.	6-48
`:FILE:SAVE?`	Queries all settings related to the saving of files.	6-48
`:FILE:SAVE:ABORt`	Aborts file saving.	6-48
`:FILE:SAVE:ACQuisition?`	Queries all settings related to the file saving of the waveform sampling data.	6-48
`:FILE:SAVE:ACQuisition[:EXECute]`	Saves the waveform sampling data to a file.	6-48
`:FILE:SAVE:ACQuisition:TRACe`	Sets the waveform sampling data to be saved to a file or queries the current setting.	6-48
`:FILE:SAVE:ACQuisition:TYPE`	Sets the format of the waveform sampling data to be saved or queries the current setting.	6-48
`:FILE:SAVE:ANAMing`	Sets whether to automatically name the files to be saved or queries the current setting.	6-48
`:FILE:SAVE:COMMent`	Sets the comment to be added to the file to be saved or queries the current setting.	6-48
`:FILE:SAVE:NUMeric?`	Queries all settings related to the saving of numeric data files.	6-49
`:FILE:SAVE:NUMeric:CBCycle?`	Queries all settings related to Cycle by Cycle measurement items saved to numeric data files or queries the current setting.	6-49
`:FILE:SAVE:NUMeric:CBCycle:ALL`	Collectively turns ON/OFF the output of all elements and functions when saving numeric data from Cycle by Cycle measurement to a file.	6-49
`:FILE:SAVE:NUMeric:CBCycle:{ELEMent<x>|SIGMA|SIGMB}`	Turns ON/OFF the output of {each element | ΣA|ΣB} when saving numeric data from Cycle by Cycle measurement to a file.	6-49
`:FILE:SAVE:NUMeric:CBCycle:<Function>`	Turns ON/OFF the output of each function when saving numeric data from Cycle by Cycle measurement to a file or queries the current setting.	6-49
`:FILE:SAVE:NUMeric[:EXECute]`	Saves the numeric data to a file.	6-49
`:FILE:SAVE:NUMeric:NORMal?`	Queries all settings related to the items saved to numeric data files.	6-50
`:FILE:SAVE:NUMeric[:NORMal]:ALL`	Collectively turns ON/OFF the output of all element functions when saving the numeric data to a file.	6-50
`:FILE:SAVE:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB}`	Turns ON/OFF the output of {each element|ΣA|ΣB} when saving the numeric data to a file.	6-50
`:FILE:SAVE:NUMeric[:NORMal]:PRESet<x>`	Presets the output ON/OFF pattern of the element function for saving the numeric data to a file.	6-50
`:FILE:SAVE:NUMeric[:NORMal]:<Function>`	Turns ON/OFF the output of the function when saving the numerical data to a file or queries the current setting.	6-50

Command	Function	Page
:FILE:SAVE:NUMeric:TYPE	Sets the format of the numeric data to be saved or queries the current setting.	6-50
:FILE:SAVE:SETup[:EXECute]	Executes the saving of the setup parameter file.	6-50
:FILE:SAVE:WAVE?	Queries all settings related to the saving of waveform display data files.	6-50
:FILE:SAVE:WAVE[:EXECute]	Saves the waveform display data to a file.	6-51
:FILE:SAVE:WAVE:TRACe	Sets the waveform to be saved to a file or queries the current setting.	6-51
:FILE:SAVE:WAVE:TYPE	Sets the format of the waveform display data to be saved or queries the current setting.	6-51
FLICker Group		
:FLICker?	Queries all settings related to flicker measurement.	6-52
:FLICker:COUNt	Sets the number of measurements for the short-term flicker value Pst or queries the current setting.	6-52
:FLICker:DC?	Queries all settings related to the relative steady-state voltage change dc.	6-52
:FLICker:DC:LIMit	Sets the limit of the relative steady-state voltage change dc or queries the current setting.	6-52
:FLICker:DC[:STATe]	Turns ON/OFF judgment of the relative steady-state voltage change dc or queries the current setting.	6-52
:FLICker:DISPlay?	Queries all settings related to flicker measurement display.	6-52
:FLICker:DISPlay:ELEMent	Sets the element to be displayed for flicker measurement display or queries the current setting.	6-52
:FLICker:DISPlay:PAGE	Sets the number of the page to be displayed for flicker measurement display or queries the current setting.	6-53
:FLICker:DISPlay:PERiod	Sets the observation period number to be displayed for flicker measurement display or queries the current setting.	6-53
:FLICker:DMAX?	Queries all settings related to the maximum relative voltage change dmax.	6-53
:FLICker:DMAX:LIMit	Sets the limit of the maximum relative voltage change dmax or queries the current setting.	6-53
:FLICker:DMAX[:STATe]	Turns ON/OFF judgment of the maximum relative voltage change dmax or queries the current setting.	6-53
:FLICker:DMIN?	Queries all settings related to the steady-state range dmin.	6-53
:FLICker:DMIN:LIMit	Sets the limit of steady-state range dmin or queries the current setting.	6-53
:FLICker:DT?	Queries all settings related to the relative voltage change time d(t).	6-53
:FLICker:DT:LIMit	Sets the limit of the relative voltage change time d(t) or queries the current setting.	6-53
:FLICker:DT[:STATe]	Turns ON/OFF judgment of the relative voltage change time d(t) or queries the current setting.	6-53
:FLICker:ELEMent<x>	Sets the target element of flicker measurement or queries the current setting.	6-54
:FLICker:FREQuency	Sets the target frequency of flicker measurement or queries the current setting.	6-54
:FLICker:INITialize	Initializes flicker measurement.	6-54
:FLICker:INTerval	Sets the time per measurement for the short-term flicker value Pst or queries the current setting.	6-54
:FLICker:JUDGe	Stops measurement of dmax caused by manual switching and executes judgment.	6-54
:FLICker:MEASurement	Sets the flicker measurement method or queries the current setting.	6-54
:FLICker:MOVe	Moves the observation period number for measurement of dmax caused by manual switching.	6-54
:FLICker:PLT?	Queries all settings related to the long-term flicker value Plt.	6-54
:FLICker:PLT:LIMit	Sets the limit of the long-term flicker value Plt or queries the current setting.	6-55
:FLICker:PLT:NVALue	Sets constant N for the equation used to compute the long-term flicker value Plt or queries the current setting.	6-55
:FLICker:PLT[:STATe]	Turns ON/OFF judgment of the long-term flicker value Plt or queries the current setting.	6-55
:FLICker:PST?	Queries all settings related to the short-term flicker value Pst.	6-55
:FLICker:PST:LIMit	Sets the limit for the short-term flicker value Pst or queries the current setting.	6-55

Command	Function	Page
`:FLICker:PST[:STATe]`	Turns ON/OFF judgment of the short-term flicker value Pst or queries the current setting.	6-55
`:FLICker:RESet`	Resets measured flicker data.	6-55
`:FLICker:STARt`	Starts flicker measurement.	6-55
`:FLICker:STATe?`	Queries the status of flicker measurement.	6-55
`:FLICker:UN?`	Queries all settings related to rated voltage Un.	6-55
`:FLICker:UN:MODE`	Sets the assignment method for rated voltage Un or queries the current setting.	6-55
`:FLICker:UN:VALue`	Sets the predefined rated voltage Un or queries the current setting.	6-56
HARMonics Group		
`:HARMonics?`	Queries all settings related to harmonic measurement.	6-57
`:HARMonics:FBANd`	Sets the frequency bandwidth of the harmonic measurement or queries the current setting.	6-57
`:HARMonics:IEC?`	Queries all settings related to IEC harmonic measurement.	6-57
`:HARMonics:IEC:OBJect`	Sets the IEC harmonic measurement target or queries the current setting.	6-57
`:HARMonics:IEC:{UGRouping|IGRouping}`	Sets the {voltage|current} grouping of the IEC harmonic measurement or queries the current setting.	6-57
`:HARMonics:ORDer`	Sets the maximum and minimum orders to be measured or queries the current setting.	6-57
`:HARMonics:PLLSource`	Sets the PLL source or queries the current setting.	6-58
`:HARMonics:PLLWarning?`	Queries all settings related to the warning messages of the PLL source.	6-58
`:HARMonics:PLLWarning[:STATe]`	Sets whether to generate a warning message when the PLL source is not applied or queries the current setting.	6-58
`:HARMonics:THD`	Sets the equation used to compute the THD (total harmonic distortion) or queries the current setting.	6-58
HCOPy Group		
`:HCOPy?`	Queries all settings related to printing.	6-59
`:HCOPy:ABORt`	Aborts printing or paper feeding.	6-59
`:HCOPy:AUTO?`	Queries all settings related to the auto print.	6-59
`:HCOPy:AUTO:INTerval`	Sets the auto print interval or queries the current setting.	6-59
`:HCOPy:AUTO:{STARt|END}`	Sets the {start|stop} reservation time of the auto print or queries the current setting.	6-59
`:HCOPy:AUTO[:STATe]`	Turns ON/OFF the auto print or queries the current setting.	6-59
`:HCOPy:AUTO:SYNChronize`	Sets the synchronization mode of the auto print or queries the current setting.	6-59
`:HCOPy:COMMent`	Sets the comment displayed at the bottom of the screen or queries the current setting.	6-60
`:HCOPy:DIRection`	Sets the printer or queries the current setting.	6-60
`:HCOPy:EXECute`	Executes the print operation.	6-60
`:HCOPy:NETPrint?`	Queries all settings related to printing using a network printer.	6-60
`:HCOPy:NETPrint:COLor`	Turns ON/OFF color printing on the network printer or queries the current setting.	6-60
`:HCOPy:NETPrint:FORMat`	Sets the printer description language for printing on a network printer or queries the current setting.	6-60
`:HCOPy:PRINter?`	Queries all settings related to printing when using the built-in printer.	6-60
`:HCOPy:PRINter:FEED`	Executes paper feeding of the built-in printer.	6-60
`:HCOPy:PRINter:FORMat`	Sets the contents to be printed on the built-in printer or queries the current setting.	6-60
`:HCOPy:PRINter:LIST?`	Queries all settings related to the printing of the numeric data list using the built-in printer.	6-61
`:HCOPy:PRINter:LIST:INFOrmation`	Sets whether to add setup parameters when printing the numeric data list using the built-in printer or queries the current setting.	6-61
`:HCOPy:PRINter:LIST:NORMal?`	Queries all settings related to the printed items of the numeric data list using the built-in printer.	6-61

Command	Function	Page
:HCOPy:PRINter:LIST[:NORMal]:ALL	Collectively turns ON/OFF the output of all element functions when printing the numeric data list on the built-in printer.	6-61
:HCOPy:PRINter:LIST[:NORMal]:{ELEMent<x>|SIGMA|SIGMB}	Turns ON/OFF the output of {each element|ΣA|ΣB} when printing the numeric data list on the built-in printer.	6-61
:HCOPy:PRINter:LIST[:NORMal]:PRESet<x>	Presets the output ON/OFF pattern of the element functions when printing the numeric data list on the built-in printer.	6-61
:HCOPy:PRINter:LIST[:NORMal]:<Function>	Turns ON/OFF the output of the function when printing the numerical data list using the built-in printer or queries the current setting.	6-62
HOLD Group		
:HOLD	Sets the output data (display, communications, etc.) hold or queries the current setting.	6-63
IMAGe Group		
:IMAGe?	Queries all settings related to the saving of the screen image data.	6-64
:IMAGe:ABORt	Aborts the saving of the screen image data.	6-64
:IMAGe:COLor	Sets the color tone of the screen image data to be saved or queries the current setting.	6-64
:IMAGe:COMMent	Sets the comment displayed at the bottom of the screen or queries the current setting.	6-64
:IMAGe:COMPression	Sets the data compression when the screen image data is in BMP format or queries the current setting.	6-64
:IMAGe:EXECute	Saves the screen image data.	6-64
:IMAGe:FORMat	Sets the output format of the screen image data to be saved or queries the current setting.	6-64
:IMAGe:SAVE?	Queries all settings related to the saving of the screen image data.	6-64
:IMAGe:SAVE:ANAMing	Sets whether to automatically name the screen image data files to be saved or queries the current setting.	6-64
:IMAGe:SAVE:CDIRectory	Changes the save destination directory for the screen image data.	6-64
:IMAGe:SAVE:DRIVe	Sets the save destination drive for the screen image data.	6-65
:IMAGe:SAVE:NAME	Sets the name of the file for saving the screen image data or queries the current setting.	6-65
:IMAGe:SEND?	Queries the screen image data.	6-65
INPut Group		
:INPut?	Queries all settings related to the input element.	6-66
[:INPut]:CFACtor	Sets the crest factor or queries the current setting.	6-66
[:INPut]:CURRent?	Queries all settings related to the current measurement.	6-66
[:INPut]:CURRent:AUTO?	Queries the current auto range ON/OFF setting of all elements.	6-66
[:INPut]:CURRent:AUTO[:ALL]	Collectively turns ON/OFF the current auto range of all elements.	6-66
[:INPut]:CURRent:AUTO:ELEMent<x>	Turns ON/OFF the current auto range of the element or queries the current setting.	6-67
[:INPut]:CURRent:AUTO:{SIGMA|SIGMB}	Collectively turns ON/OFF the current auto range of all elements belonging to wiring unit {ΣA|ΣB}.	6-67
[:INPut]:CURRent:MODE?	Queries the current mode of all elements.	6-67
[:INPut]:CURRent:MODE[:ALL]	Collectively sets the current mode of all elements.	6-67
[:INPut]:CURRent:MODE:ELEMent<x>	Sets the current mode of the element or queries the current setting.	6-67
[:INPut]:CURRent:MODE:{SIGMA|SIGMB}	Collectively sets the current mode of all elements belonging to wiring unit {ΣA|ΣB}.	6-67
[:INPut]:CURRent:RANGe?	Queries the current ranges of all elements.	6-67
[:INPut]:CURRent:RANGe[:ALL]	Collectively sets the current ranges of all elements.	6-67

Command	Function	Page
`[:INPut]:CURRent:RANGe:ELEMent<x>`	Sets the current range of the element or queries the current setting.	6-68
`[:INPut]:CURRent:RANGe:{SIGMA|SIGMB}`	Collectively sets the current range of all elements belonging to wiring unit {ΣA|ΣB}.	6-68
`[:INPut]:CURRent:SRATio?`	Queries the current sensor scaling constants of all elements.	6-69
`[:INPut]:CURRent:SRATio[:ALL]`	Collectively sets the current sensor scaling constants of all elements.	6-69
`[:INPut]:CURRent:SRATio:ELEMent<x>`	Sets the current sensor scaling constant of the element or queries the current setting.	6-69
`[:INPut]:FILTer?`	Queries all settings related to the input filter.	6-69
`[:INPut]:FILTer:FREQuency?`	Queries the frequency filter ON/OFF setting of all elements.	6-69
`[:INPut]:FILTer:FREQuency[:ALL]`	Collectively turns ON/OFF the frequency filter of all elements.	6-69
`[:INPut]:FILTer:FREQuency:ELEMent<x>`	Turns ON/OFF the frequency filter of the element or queries the current setting.	6-69
`[:INPut]:FILTer:LINE?`	Queries the line filter settings of all elements.	6-69
`[:INPut]:FILTer[:LINE][:ALL]`	Collectively sets the line filters of all elements.	6-69
`[:INPut]:FILTer[:LINE]:ELEMent<x>`	Sets the line filter of the element or queries the current setting.	6-69
`[:INPut]:INDependent`	Turns ON/OFF the independent setting of input elements or queries the current setting.	6-69
`[:INPut]:MODUle?`	Queries the input element type.	6-70
`[:INPut]:NULL`	Turns ON/OFF the NULL function or queries the current setting.	6-70
`[:INPut]:POVer?`	Queries the peak over information.	6-70
`[:INPut]:SCALing?`	Queries all settings related to scaling.	6-70
`[:INPut]:SCALing:{VT|CT|SFACtor}?`	Queries the scaling constant of all elements.	6-70
`[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL]`	Collectively sets the scaling constants of all elements.	6-70
`[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x>`	Sets the scaling constant of the element or queries the current setting.	6-70
`[:INPut]:SCALing:STATe?`	Queries the scaling ON/OFF states of all elements.	6-70
`[:INPut]:SCALing[:STATe][:ALL]`	Collectively turns ON/OFF the scaling of all elements.	6-70
`[:INPut]:SCALing[:STATe]:ELEMent<x>`	Turns ON/OFF the scaling of the element or queries the current setting.	6-71
`[:INPut]:SYNChronize?`	Queries the synchronization source of all elements.	6-71
`[:INPut]:SYNChronize[:ALL]`	Collectively sets the synchronization source of all elements.	6-71
`[:INPut]:SYNChronize:ELEMent<x>`	Sets the synchronization source of the element or queries the current setting.	6-71
`[:INPut]:SYNChronize:{SIGMA|SIGMB}`	Collectively sets the synchronization source of all elements belonging to wiring unit {ΣA|ΣB}.	6-71
`[:INPut]:VOLTage?`	Queries all settings related to the voltage measurement.	6-71
`[:INPut]:VOLTage:AUTO?`	Queries the voltage auto range setting (ON/OFF) of all elements.	6-71
`[:INPut]:VOLTage:AUTO[:ALL]`	Collectively turns ON/OFF the voltage auto range of all elements.	6-71
`[:INPut]:VOLTage:AUTO:ELEMent<x>`	Turns ON/OFF the voltage auto range of the element or queries the current setting.	6-71
`[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB}`	Collectively turns ON/OFF the voltage auto range of all elements belonging to wiring unit {ΣA|ΣB}.	6-71
`[:INPut]:VOLTage:MODE?`	Queries the voltage mode of all elements.	6-72
`[:INPut]:VOLTage:MODE[:ALL]`	Collectively sets the voltage mode of all elements.	6-72
`[:INPut]:VOLTage:MODE:ELEMent<x>`	Sets the voltage mode of the element or queries the current setting.	6-72

Command	Function	Page
`[:INPut]:VOLTage:MODE:{SIGMA|SIGMB}`	Collectively sets the voltage mode of all elements belonging to wiring unit {ΣA|ΣB}.	6-72
`[:INPut]:VOLTage:RANGe?`	Queries the voltage range of all elements.	6-72
`[:INPut]:VOLTage:RANGe[:ALL]`	Collectively sets the voltage range of all elements.	6-72
`[:INPut]:VOLTage:RANGe:ELEMent<x>`	Sets the voltage range of the element or queries the current setting.	6-72
`[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB}`	Collectively sets the voltage range of all elements belonging to wiring unit {ΣA|ΣB}.	6-72
`[:INPut]:WIRing`	Sets the wiring system or queries the current setting.	6-73

INTEGrate Group

Command	Function	Page
`:INTEGrate?`	Queries all settings related to the integration.	6-74
`:INTEGrate:ACAL`	Turns ON/OFF the auto calibration or queries the current setting.	6-74
`:INTEGrate:MODE`	Sets the integration mode or queries the current setting.	6-74
`:INTEGrate:RESet`	Resets the integrated value.	6-74
`:INTEGrate:RTIMe?`	Queries the integration start and stop times for real-time integration mode.	6-74
`:INTEGrate:RTIMe:{STARt|END}`	Sets the integration {start|stop} time for real-time integration mode or queries the current setting.	6-74
`:INTEGrate:STARt`	Starts the integration.	6-74
`:INTEGrate:STATe?`	Queries the integration condition.	6-74
`:INTEGrate:STOP`	Stops the integration.	6-74
`:INTEGrate:TIMer`	Sets the integration timer time or queries the current setting.	6-75

MEASure Group

Command	Function	Page
`:MEASure?`	Queries all settings related to the computation.	6-76
`:MEASure:AVERaging?`	Queries all settings related to averaging.	6-76
`:MEASure:AVERaging:COUNt`	Sets the averaging coefficient or queries the current setting.	6-76
`:MEASure:AVERaging[:STATe]`	Turns ON/OFF averaging or queries the current setting.	6-76
`:MEASure:AVERaging:TYPE`	Sets the averaging type or queries the current setting.	6-77
`:MEASure:COMPensation?`	Queries all settings related to the compensation computation.	6-77
`:MEASure:COMPensation:EFFiciency`	Turns ON/OFF the efficiency compensation or queries the current setting.	6-77
`:MEASure:COMPensation:V3A3`	Turns ON/OFF the compensation for the two-wattmeter method or queries the current setting.	6-77
`:MEASure:COMPensation:WIRing?`	Queries all settings related to the wiring computation.	6-77
`:MEASure:COMPensation:WIRing:ELEMent<x>`	Sets the wiring compensation of the element or queries the current setting.	6-77
`:MEASure:DMeasure?`	Queries all settings related to the delta computation.	6-77
`:MEASure:DMeasure[:SIGMA]`	Sets the delta computation mode for wiring unit ΣA or queries the current setting.	6-78
`:MEASure:DMeasure:SIGMB`	Sets the delta computation mode for wiring unit ΣB or queries the current setting.	6-78
`:MEASure:EFFiciency?`	Queries all settings related to the efficiency computation.	6-78
`:MEASure:EFFiciency:ETA<x>`	Sets the efficiency equation or queries the current setting.	6-78
`:MEASure:EFFiciency:UDEF<x>`	Sets the user-defined parameter used in the efficiency equation or queries the current setting.	6-78
`:MEASure:FREQuency?`	Queries all settings related to frequency measurement.	6-79
`:MEASure:FREQuency:ITEM<x>`	Sets the frequency measurement item or queries the current setting.	6-79
`:MEASure:FUNCtion<x>?`	Queries all settings related to user-defined functions.	6-79
`:MEASure:FUNCtion<x>:EXPRession`	Sets the equation of the user-defined function or queries the current setting.	6-79
`:MEASure:FUNCtion<x>[:STATe]`	Enables (ON) or Disables (OFF) the user-defined function or queries the current setting.	6-79
`:MEASure:FUNCtion<x>:UNIT`	Sets the unit to be added to the computation result of the user-defined function or queries the current setting.	6-79

Command	Function	Page
:MEASure:MHOLd	Enables (ON) or Disables (OFF) MAX HOLD function used in the user-defined function or queries the current setting.	6-79
:MEASure:PC?	Queries all settings related to the computation of Pc (Corrected Power).	6-79
:MEASure:PC:IEC	Sets the equation used to compute Pc (Corrected Power) or queries the current setting.	6-80
:MEASure:PC:P<x>	Sets the parameter used to compute Pc (Corrected Power) or queries the current setting.	6-80
:MEASure:PHASe	Sets the display format of the phase difference or queries the current setting.	6-80
:MEASure:SAMPling	Sets the sampling frequency or queries the current setting.	6-80
:MEASure:SQFormula	Sets the equation used to compute S (apparent power) and Q (reactive power) or queries the current setting.	6-80
:MEASure:SYNChronize	Sets the synchronized measurement mode or queries the current setting.	6-80
MOTor Group		
:MOTor?	Queries all settings related to the motor evaluation function.	6-81
:MOTor:FILTer?	Queries all settings related to the input filter.	6-81
:MOTor:FILTer[:LINE]	Sets the line filter or queries the current setting.	6-81
:MOTor:PM?	Queries all settings related to the motor output (Pm).	6-81
:MOTor:PM:SCALing	Sets the scaling factor used for motor output computation or queries the current setting.	6-81
:MOTor:PM:UNIT	Sets the unit to be added to the motor output computation result or queries the current setting.	6-81
:MOTor:POLE	Sets the motor's number of poles or queries the current setting.	6-81
:MOTor:SPEed?	Queries all settings related to the rotating speed.	6-81
:MOTor:SPEed:AUTO	Turns ON/OFF the voltage auto range of the rotating speed signal input (analog input format) or queries the current setting.	6-81
:MOTor:SPEed:PRANge	Sets the range of the rotating speed (pulse input format) or queries the current setting.	6-82
:MOTor:SPEed:PULSe	Sets the pulse count of the rotating speed signal input (pulse input) or queries the current setting.	6-82
:MOTor:SPEed:RANGe	Sets the voltage range of the rotating speed signal input (analog input format) or queries the current setting.	6-82
:MOTor:SPEed:SCALing	Sets the scaling factor for rotating speed computation or queries the current setting.	6-82
:MOTor:SPEed:TYPE	Sets the input type of the rotating speed signal input or queries the current setting.	6-82
:MOTor:SPEed:UNIT	Sets the unit to be added to the rotating speed computation result or queries the current setting.	6-82
:MOTor:SSPeed	Sets the frequency measurement source used to compute the synchronous speed (SyncSp) or queries the current setting.	6-82
:MOTor:SYNChronize	Sets the synchronization source used to compute the rotating speed and torque or queries the current setting.	6-82
:MOTor:TORQue?	Queries all settings related to torque.	6-82
:MOTor:TORQue:AUTO	Turns ON/OFF the voltage auto range of the torque signal input (analog input format) or queries the current setting.	6-83
:MOTor:TORQue:PRANge	Sets the range of the torque (pulse input format) or queries the current setting.	6-83
:MOTor:TORQue:RANGe	Sets the voltage range of the torque signal input (analog input format) or queries the current setting.	6-83
:MOTor:TORQue:RATE?	Queries all settings related to the rated value of the torque signal (pulse input format).	6-83
:MOTor:TORQue:RATE:{UPPer|LOWer}	Sets the rated value {upper limit|lower limit} of the torque signal (pulse input format) or queries the current setting.	6-83
:MOTor:TORQue:SCALing	Sets the scaling factor for torque computation or queries the current setting.	6-83
:MOTor:TORQue:TYPE	Sets the input type of the torque signal input or queries the current setting.	6-83

Command	Function	Page
:MOTor:TORQue:UNIT	Sets the unit to be added to the torque computation result or queries the current setting.	6-83
NUMeric Group		
:NUMeric?	Queries all settings related to the numeric data output.	6-84
:NUMeric:CBCycle?	Queries all settings related to output of numeric list data of Cycle by Cycle measurement.	6-84
:NUMeric:CBCycle:END	Sets the output end cycle of the numeric list data output by ":NUMeric:CBCycle:VALue?" or queries the current setting.	6-84
:NUMeric:CBCycle:ITEM	Sets the numeric list data output items (function and element) of Cycle by Cycle measurement or queries the current setting.	6-84
:NUMeric:CBCycle:STARt	Sets the output start cycle of the numeric list data output by ":NUMeric:CBCycle:VALue?" or queries the current setting.	6-84
:NUMeric:CBCycle:VALue?	Queries the numeric list data from Cycle by Cycle measurement.	6-85
:NUMeric:FLICker?	Queries all settings related to output of numeric data from flicker measurement.	6-85
:NUMeric:FLICker:COUNt?	Queries the number of the measurement within the specified observation period at which flicker measurement stops.	6-85
:NUMeric:FLICker:FUNCtion?	Queries all settings related to output of measured flicker data (variable format).	6-85
:NUMeric:FLICker:FUNCtion:CLEar	Clears the output items of measured flicker data (variable format).	6-86
:NUMeric:FLICker:FUNCtion:DELete	Deletes the output items of measured flicker data (variable format).	6-86
:NUMeric:FLICker:FUNCtion:ITEM<x>	Sets output items (function, element, and observation period) of measured flicker data (variable format) or queries the current setting.	6-86
:NUMeric:FLICker:FUNCtion:NUMber	Sets the number of measured flicker data output by ":NUMeric:FLICker:FUNCtion:VALue?" or queries the current setting.	6-87
:NUMeric:FLICker:FUNCtion:VALue?	Queries the measured flicker data (variable format).	6-87
:NUMeric:FLICker:INFOrmation?	Queries all settings related to output of flicker judgment results (variable format).	6-87
:NUMeric:FLICker:INFOrmation:CLEar	Clears the output items of flicker judgment results (variable format).	6-88
:NUMeric:FLICker:INFOrmation:DELete	Deletes the output items of flicker judgment results (variable format).	6-88
:NUMeric:FLICker:INFOrmation:ITEM<x>	Sets output items (function, element, and observation period) of flicker judgment results (variable format) or queries the current setting.	6-88
:NUMeric:FLICker:INFOrmation:NUMber	Sets the number of flicker judgment results output by ":NUMeric:FLICker:INFOrmation:VALue?" or queries the current setting.	6-89
:NUMeric:FLICker:INFOrmation:VALue?	Queries the flicker judgment results (variable format).	6-89
:NUMeric:FLICker:JUDGement?	Queries the flicker judgment results (fixed format).	6-90
:NUMeric:FLICker:PERiod?	Queries the observation period number currently being measured during flicker measurement.	6-90
:NUMeric:FLICker:VALue?	Queries the measured flicker data (fixed format).	6-91
:NUMeric:FORMat	Sets the format of the numeric data that is transmitted by ":NUMeric[:NORMal]:VALue?" or ":NUMeric:LIST:VALue?" or queries the current setting.	6-91
:NUMeric:HOLD	Sets whether to hold (ON) or release (OFF) all the numeric data or queries the current setting.	6-92
:NUMeric:LIST?	Queries all settings related to the numeric list data output of harmonic measurement.	6-92
:NUMeric:LIST:CLEar	Clears the output items of the numeric list data of harmonic measurement.	6-92

Command	Function	Page
:NUMeric:LIST:DELete	Deletes the output items of the numeric list data of harmonic measurement.	6-93
:NUMeric:LIST:ITEM<x>	Sets the output items (function and element) of the numeric list data of harmonic measurement or queries the current setting.	6-93
:NUMeric:LIST:NUMber	Sets the number of the numeric list data that is transmitted by ":NUMeric:LIST:VALue?" or queries the current setting.	6-93
:NUMeric:LIST:ORDer	Sets the maximum output order of the numeric list data of harmonic measurement or queries the current setting.	6-93
:NUMeric:LIST:PRESet	Presets the output item pattern of the numeric list data of harmonic measurement.	6-93
:NUMeric:LIST:SELect	Sets the output component of the numeric list data of harmonic measurement or queries the current setting.	6-94
:NUMeric:LIST:VALue?	Queries the numeric list data of harmonic measurement.	6-94
:NUMeric:NORMal?	Queries all settings related to the numeric data output.	6-95
:NUMeric[:NORMal]:CLEar	Clears the output items of numeric data.	6-95
:NUMeric[:NORMal]:DELete	Deletes the output items of numeric data.	6-95
:NUMeric[:NORMal]:ITEM<x>	Sets the output items (function, element, and harmonic order) of numeric data or queries the current setting.	6-95
:NUMeric[:NORMal]:NUMber	Sets the number of the numeric data that is transmitted by ":NUMeric[:NORMal]:VALue?" or queries the current setting.	6-95
:NUMeric[:NORMal]:PRESet	Presets the output item pattern of numeric data.	6-95
:NUMeric[:NORMal]:VALue?	Queries the numeric data.	6-96
RATE Group		
:RATE	Sets the data update rate or queries the current setting.	6-102
STATus Group		
:STATus?	Queries all settings related to the communication status function.	6-103
:STATus:CONDition?	Queries the contents of the condition register.	6-103
:STATus:EESE	Sets the extended event enable register or queries the current setting.	6-103
:STATus:EESR?	Queries the content of the extended event register and clears the register.	6-103
:STATus:ERRor?	Queries the error code and message information (top of the error queue).	6-103
:STATus:FILTer<x>	Sets the transition filter or queries the current setting.	6-103
:STATus:QENable	Sets whether to store messages other than errors to the error queue (ON/OFF) or queries the current setting.	6-103
:STATus:QMESsage	Sets whether to attach message information to the response to the "STATus:ERRor?" query (ON/OFF) or queries the current setting.	6-103
:STATus:SPOLl?	Executes serial polling.	6-104
STORe Group		
:STORe?	Queries all settings related to store and recall.	6-105
:STORe:COUNt	Sets the store count or queries the current setting.	6-105
:STORe:DIRection	Sets the store destination or queries the current setting.	6-105
:STORe:FILE?	Queries all settings related to the saving of the stored data.	6-105
:STORe:FILE:ANAMing	Sets whether to automatically name the files when saving the stored data or queries the current setting.	6-105
:STORe:FILE:COMMent	Sets the comment to be added to the file when saving the stored data or queries the current setting.	6-105
:STORe:FILE:NAME	Sets the name of the file when saving the stored data or queries the current setting.	6-105
:STORe:FILE:TYPE	Sets the data format when saving the stored data or queries the current setting.	6-105
:STORe:INTerval	Sets the store interval or queries the current setting.	6-106
:STORe:ITEM	Sets the stored item or queries the current setting.	6-106
:STORe:MEMory?	Queries all settings related to the storage memory.	6-106
:STORe:MEMory:ALERt	Sets whether to display a confirmation message when clearing the storage memory or queries the current setting.	6-106

Command	Function	Page
:STORe:MEMory:CONVert:ABORt	Abort converting the stored data from the memory to the file.	6-106
:STORe:MEMory:CONVert:EXECute	Executes the converting of the stored data from the memory to the file.	6-106
:STORe:MEMory:INITialize	Executes the initialization of the storage memory.	6-106
:STORe:MODE	Sets the data storage/recall or queries the current setting.	6-106
:STORe:NUMeric?	Queries all settings related to the storing of numeric data.	6-107
:STORe:NUMeric:NORMal?	Queries all settings related to the stored items of numeric data.	6-107
:STORe:NUMeric[:NORMal]:ALL	Collectively turns ON/OFF the output of all element functions when storing the numeric data.	6-107
:STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB}	Turns ON/OFF the output of {each element|ΣA|ΣB} when storing the numeric data.	6-107
:STORe:NUMeric[:NORMal]:PRESet<x>	Presets the output ON/OFF pattern of the element function for storing the numeric data.	6-107
:STORe:NUMeric[:NORMal]:<Function>	Turns ON/OFF the output of the element when storing the numeric data or queries the current setting.	6-107
:STORe:RECall	Sets the data number to be recalled or queries the current setting.	6-108
:STORe:RTIMe?	Queries the store reservation time for real-time store mode.	6-108
:STORe:RTIMe:{STARt|END}	Sets the store {start|stop} reservation time for real-time store mode or queries the current setting.	6-108
:STORe:SMODe	Sets the store mode or queries the current setting.	6-108
:STORe:STARt	Starts the data store operation.	6-108
:STORe:STOP	Stops the data store operation.	6-108
:STORe:WAVE?	Queries all settings related to the storing of waveform display data.	6-108
:STORe:WAVE:ALL	Collectively turns ON/OFF the output of all waveforms when storing the waveform display data.	6-108
:STORe:WAVE:{U<x>|I<x>|SPEed|TORQue}	Turns ON/OFF the output of the waveform when storing the waveform display data or queries the current setting.	6-108

SYSTem Group

Command	Function	Page
:SYSTem?	Queries all settings related to the system.	6-109
:SYSTem:CLOCk?	Queries all settings related to the date/time.	6-109
:SYSTem:CLOCk:DISPlay	Turns ON/OFF the date/time display or queries the current setting.	6-109
:SYSTem:CLOCk:SNTP?	Queries all settings related to the date/time acquired via SNTP.	6-109
:SYSTem:CLOCk:SNTP[:EXECute]	Sets the date/time via SNTP.	6-109
:SYSTem:CLOCk:SNTP:GMTTime	Sets the difference from Greenwich Mean Time or queries the current setting.	6-109
:SYSTem:CLOCk:TYPE	Sets the date/time setting method or queries the current setting.	6-109
:SYSTem:DATE	Sets the date or queries the current setting.	6-109
:SYSTem:ECLear	Clears the error message displayed on the screen.	6-109
:SYSTem:FONT	Sets the display font or queries the current setting.	6-109
:SYSTem:KLOCk	Turns ON/OFF the key lock or queries the current setting.	6-110
:SYSTem:LANGuage?	Queries all settings related to the display language.	6-110
:SYSTem:LANGuage:MENU	Sets the menu language or queries the current setting.	6-110
:SYSTem:LANGuage:MESSage	Sets the message language or queries the current setting.	6-110
:SYSTem:LCD?	Queries all settings related to the LCD monitor.	6-110
:SYSTem:LCD:BRIGhtness	Sets the brightness of the LCD monitor or queries the current setting.	6-110
:SYSTem:LCD:COLor?	Queries all settings related to the display colors of the LCD monitor.	6-110
:SYSTem:LCD:COLor:GRAPh?	Queries all settings related to the display colors of the graphic items.	6-110
:SYSTem:LCD:COLor:GRAPh:{BACKground|GRATicule|CURSor|U<x>|I<x>}	Sets the display color of the {background|graticule|cursor|voltage waveform|current waveform} or queries the current setting.	6-110
:SYSTem:LCD:COLor:GRAPh:MODE	Sets the display color mode of the graphic items or queries the current setting.	6-110

Command	Function	Page
:SYSTem:LCD:COLor:TEXT?	Queries all settings related to the display colors of the text items.	6-111
:SYSTem:LCD:COLor:TEXT:{LETTer|BACKground|BOX|SUB|SELected}	Sets the display color of the {text (Menu Fore)|menu background (Menu Back)|selected menu (Select Box)|pop-up menu (Sub Menu)| selected key (Selected Key)} or queries the current setting.	6-111
:SYSTem:LCD:COLor:TEXT:MODE	Sets the display color mode of the text items or queries the current setting.	6-111
:SYSTem:SLOCk	Sets whether to continue the SHIFT key ON state or queries the current setting.	6-111
:SYSTem:TIME	Sets the time or queries the current setting.	6-111
:SYSTem:USBKeyboard	Sets the USB keyboard type or queries the current setting.	6-111
WAVeform Group		
:WAVeform?	Queries all settings related to the waveform display data output.	6-112
:WAVeform:BYTeorder	Sets the output byte order of the waveform display data (FLOAT format) that is transmitted by ":WAVeform:SEND?" or queries the current setting.	6-112
:WAVeform:END	Sets the output end point of the waveform display data that is transmitted by ":WAVeform:SEND?" or queries the current setting.	6-112
:WAVeform:FORMat	Sets the format of the waveform display data that is transmitted by ":WAVeform:SEND?" or queries the current setting.	6-112
:WAVeform:HOLD	Sets whether to hold (ON) or release (OFF) all the waveform display data or queries the current setting.	6-112
:WAVeform:LENGth?	Queries the total number of points of the waveform specified by ":WAVeform:TRACe".	6-112
:WAVeform:SEND?	Queries the waveform display data specified by ":WAVeform:TRACe".	6-113
:WAVeform:SRATe?	Queries the sample rate of the retrieved waveform.	6-113
:WAVeform:STARt	Sets the output start point of the waveform display data that is transmitted by ":WAVeform:SEND?" or queries the current setting.	6-113
:WAVeform:TRACe	Sets the target waveform for the commands in the WAVeform group or queries the current setting.	6-113
:WAVeform:TRIGger?	Queries the trigger position of the retrieved waveform.	6-113
Common Command Group		
*CAL?	Executes zero calibration (zero-level compensation, same operation as pressing CAL (SHIFT+SINGLE)) and queries the result.	6-114
*CLS	Clears the standard event register, extended event register, and error queue.	6-114
*ESE	Sets the standard event enable register or queries the current setting.	6-114
*ESR?	Queries the standard event register.	6-114
*IDN?	Queries the instrument model.	6-114
*OPC	Sets a 1 to bit 0 (OPC bit) of the standard event register bit upon the completion of the specified overlap command.	6-115
*OPC?	ASCII code "1" is returned when the specified overlap command is completed.	6-115
*OPT?	Queries the installed options.	6-115
*PSC	Sets whether to clear the registers at power on or queries the current setting.	6-115
*RST	Executes the initialization of settings.	6-115
*SRE	Sets the service request enable register or queries the current setting.	6-115
*STB?	Queries the status byte register.	6-116
*TRG	Executes single measurement (the same operation as when SINGLE is pressed).	6-116
*TST?	Performs a self-test and queries the result.	6-116
*WAI	Holds the subsequent command until the completion of the specified overlap operation.	6-116

6.2 ACQuisition Group

The commands in this group deal with output of the waveform sampling data (acquisition data).
There are no front panel keys that correspond to the commands in this group.
The commands in this group are valid only when the advanced computation function (/G6 option) is installed.

:ACQuisition?

Function Queries all settings related to the output of the waveform sampling data.

```
Syntax      :ACQuisition?
Example     :ACQUISITION? -> :ACQUISITION:
            TRACE U1;FORMAT ASCII;START 0;
            END 199999;HOLD 0
```

:ACQuisition:BYTeorder

Function Sets the output byte order of the waveform sampling data (FLOAT format) that is transmitted by ":ACQuisition:SEND?" or queries the current setting.

```
Syntax      :ACQuisition:BYTeorder {LSBFirst|
            MSBFirst}
            :ACQuisition:BYTeorder?
Example     :ACQUISITION:BYTEORDER LSBFIRST
            :ACQUISITION:BYTEORDER? ->
            :ACQUISITION:BYTEORDER LSBFIRST
```

Description This value is valid when ":ACQuisition:FORMat" is set to FLOat.

:ACQuisition:END

Function Sets the output end point of the waveform display data that is transmitted by ":ACQuisition:SEND?" or queries the current setting.

```
Syntax      :ACQuisition:END {<NRf>}
            :ACQuisition:END?
            <NRf> = 0 to 3999999
Example     :ACQUISITION:END 199999
            :ACQUISITION:END? ->
            :ACQUISITION:END 199999
```

Description Set the point in the range up to (the total number of data points – 1). The total number of data points can be queried using ":ACQuisition:LENGth?."

:ACQuisition:FORMat

Function Sets the format of the waveform sampling data that is transmitted by ":ACQuisition:SEND?" or queries the current setting.

```
Syntax      :ACQuisition:FORMat {ASCii|FLOat}
            :ACQuisition:FORMat?
Example     :ACQUISITION:FORMAT FLOAT
            :ACQUISITION:FORMAT? ->
            :ACQUISITION:FORMAT FLOAT
```

Description For the differences in the waveform sampling data output due to the format setting, see the description for ":ACQuisition:SEND?."

:ACQuisition:HOLD

Function Sets whether to hold (ON) or release (OFF) all the waveform sampling data or queries the current setting.

```
Syntax      :ACQuisition:HOLD {<Boolean>}
            :ACQuisition:HOLD?
Example     :ACQUISITION:HOLD ON
            :ACQUISITION:HOLD? ->
            :ACQUISITION:HOLD 1
```

Description
- This command is valid when the measurement mode is set to MATH or FFT. Otherwise, an error occurs.
- When ":ACQuisition:HOLD" is turned ON, the WT3000 stops sampling the waveform sampling data and holds all of the waveform sampling data at that point internally. Be sure to set :ACQuisition:HOLD to ON before executing ":ACQuisition:SEND?."
- For example, if you wish to retrieve the waveform sampling data of U1 and I1 at the same point, do the following:
  ```
  :ACQuisition:HOLD ON
  :ACQuisition:TRACe U1
  :ACQuisition:SEND?
  ```
  (Receive the waveform sampling data of U1)
  ```
  :ACQuisition:TRACe I1
  :ACQuisition:SEND?
  ```
  (Receive the waveform sampling data of I1)
  ```
  :ACQuisition:HOLD OFF
  ```
- To retrieve new waveform sampling data, set :ACQuisition:HOLD to OFF to resume sampling, and then set :ACQuisition:HOLD to ON again.

:ACQuisition:LENGth?

Function Queries the total number of points of the waveform sampling specified by ":ACQuisition:TRACe."

Syntax `:ACQuisition:LENGth?`

Example `:ACQUISITION:LENGTH? -> 100000`

Description
- When the measurement mode is MATH, the number of data is determined by the data update interval (`:RATE`) setting.
  length = rate(sec) × 200000
- When the measurement mode is FFT, the number of data is determined by the setting for the number of FFT computation points (`:DISPlay:FFT:POINt`) as follows:
  (1) When "`:ACQuisition:TRACe`" is FFT<x>
  length = point/2+1 = 10001 or 100001
  (2) When "`:ACQuisition:TRACe`" is not FFT<x>
  length = point = 20000 or 200000
- In a mode other than MATH or FFT, an error occurs since there is no acquisition data, and 0 is returned.

:ACQuisition:SEND?

Function Queries the waveform sampling data specified by ":ACQuisition:TRACe."

Syntax `:ACQuisition:SEND?`

Example
- When "`:ACQuisition:FORMat`" is set to `{ASCii}`
  `:ACQUISITION:SEND? ->`
  `<NR3>,<NR3>,...`
- When "`:ACQuisition:FORMat`" is set to `{FLOat}`
  `:ACQUISITION:SEND? -> #8` (number of bytes, 8 digits)(data byte sequence)

Description
- The format of the waveform sampling data that is output varies depending on the "`:ACQuisition:FORMat`" setting as follows:
  (1) When "`ASCii`" is specified
  The physical value is output in the <NR3> format. The data of each point is delimited by a comma.
  (2) When "`FLOat`" is specified
  The physical value is output in IEEE single-precision floating point (4-byte) format.
  The output byte order of the data of each point follows the order that is set using the "`:ACQuisition:BYTeorder`" command.
- The WT3000 outputs the waveform sampling data in the range specified by "`:ACQuisition:{STARt|END}`." However, data exceeding the waveform sampling data range, 0 to (the total number of data points – 1), is not output.
- This command is valid when the measurement mode is set to MATH or FFT and the waveform sampling data is held inside the WT3000 (`:ACQuisition:HOLD ON`). Otherwise, an error occurs, because there is no waveform sampling data. The output is as follows:
  (1) When "`ASCii`" is specified
  Outputs NAN.
  (2) When "`FLOat`" is specified
  Outputs #800000000.

:ACQuisition:SRATe?

Function Queries the sampling rate of the retrieved data

Syntax `:ACQuisition:SRATe?`

Example `:ACQUISITION:SRATE? -> 195.312E+03`

Description This command is valid when the measurement mode is set to MATH or FFT. Otherwise, an error occurs (returns NAN), because there is no waveform sampling data.

:ACQuisition:STARt

Function Sets the output start point of the waveform display data that is transmitted by "`:ACQuisition:SEND?`" or queries the current setting.

Syntax `:ACQuisition:STARt {<NRf>}`
`:ACQuisition:STARt?`
<NRf> = 0 to 3999999

Example `:ACQUISITION:START 0`
`:ACQUISITION:START? ->`
`:ACQUISITION:START 0`

Description Set the point in the range up to (the total number of data points – 1). The total number of data points can be queried using "`:ACQuisition:LENGth?`."

:ACQuisition:TRACe

Function Sets the target trace of "`:ACQuisition:SEND?`" or queries the current setting.

Syntax `:ACQuisition:TRACe {U<x>|I<x>|SPEed|TORQue|MATH<x>|FFT<x>}`
`:ACQuisition:TRACe?`
<x> of U<x>, I<x> = 1 to 4 (element)
<x> of MATH<x> = 1 to 2 (MATH)
<x> of FFT<x> = 1 or 2 (FFT)

Example `:ACQUISITION:TRACE U1`
`:ACQUISITION:TRACE? ->`
`:ACQUISITION:TRACE U1`

Description `{SPEed|TORQue}` are valid only on the motor version (-MV).

6.3 AOUTput Group

The commands in this group deal with the D/A output.
You can make the same settings and inquiries as when the “D/A Output Items” menu of MISC on the front panel is used.
However, the commands in this group are valid only when the D/A output (/DA option) is installed.

:AOUTput?

Function Queries all settings related to the D/A output.
Syntax `:AOUTput?`
Example `:AOUTPUT?` -> Same as the response to “`:AOUTput:NORMal?`”

:AOUTput:NORMal?

Function Queries all settings related to the D/A output.
Syntax `:AOUTput:NORMal?`
Example

```
:AOUTPUT:NORMAL? -> :AOUTPUT:
NORMAL:CHANNEL1 U,1,TOTAL;
CHANNEL2 I,1,TOTAL;
CHANNEL3 P,1,TOTAL;
CHANNEL4 S,1,TOTAL;
CHANNEL5 Q,1,TOTAL;
CHANNEL6 LAMBDA,1,TOTAL;
CHANNEL7 PHI,1,TOTAL;CHANNEL8 FU,1;
CHANNEL9 FI,1;CHANNEL10 NONE;
CHANNEL11 NONE;CHANNEL12 NONE;
CHANNEL13 NONE;CHANNEL14 NONE;
CHANNEL15 NONE;CHANNEL16 NONE;
CHANNEL17 NONE;CHANNEL18 NONE;
CHANNEL19 NONE;CHANNEL20 NONE;
MODE1 FIXED;MODE2 FIXED;
MODE3 FIXED;MODE4 FIXED;
MODE5 FIXED;MODE6 FIXED;
MODE7 FIXED;MODE8 FIXED;
MODE9 FIXED;MODE10 FIXED;
MODE11 FIXED;MODE12 FIXED;
MODE13 FIXED;MODE14 FIXED;
MODE15 FIXED;MODE16 FIXED;
MODE17 FIXED;MODE18 FIXED;
MODE19 FIXED;MODE20 FIXED;
RATE1 100.0E+00,-100.0E+00;
RATE2 100.0E+00,-100.0E+00;
RATE3 100.0E+00,-100.0E+00;
RATE4 100.0E+00,-100.0E+00;
RATE5 100.0E+00,-100.0E+00;
RATE6 100.0E+00,-100.0E+00;
RATE7 100.0E+00,-100.0E+00;
RATE8 100.0E+00,-100.0E+00;
RATE9 100.0E+00,-100.0E+00;
RATE10 100.0E+00,-100.0E+00;
RATE11 100.0E+00,-100.0E+00;
RATE12 100.0E+00,-100.0E+00;
RATE13 100.0E+00,-100.0E+00;
RATE14 100.0E+00,-100.0E+00;
RATE15 100.0E+00,-100.0E+00;
RATE16 100.0E+00,-100.0E+00;
RATE17 100.0E+00,-100.0E+00;
RATE18 100.0E+00,-100.0E+00;
RATE19 100.0E+00,-100.0E+00;
RATE20 100.0E+00,-100.0E+00;
IRTIME 1,0,0
```

:AOUTput[:NORMal]:CHANnel<x>

Function Sets the D/A output items (function, element, and harmonic order) or queries the current setting.
Syntax `:AOUTput[:NORMal]:CHANnel<x> {NONE|<Function>,<Element>,<Order>}`
`:AOUTput[:NORMal]:CHANnel<x>?`
<x> = 1 to 20 (output channel)
NONE = No output item
<Function> = `{U|I|P|S|Q|...}`(See the function selection list (1) of “DISPlay group” on page 6-44.)
<Element> = `{<NRf>|SIGMA|SIGMB}` (<NRf> = 1 to 4)
<Order> = `{TOTal|DC|<NRf>}` (<NRf> = 1 to 100)
Example `:AOUTPUT:NORMAL:CHANNEL1 U,1,TOTAL`
`:AOUTPUT:NORMAL:CHANNEL1? ->`
`:AOUTPUT:NORMAL:CHANNEL1 U,1,TOTAL`
Description
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTal is set.
- <Element> or <Order> is omitted from response to functions that do not need them.

:AOUTput[:NORMal]:IRTime

Function Sets the rated integration time for the D/A output of integrated values or queries the current setting.
Syntax `:AOUTput[:NORMal]:IRTime {<NRf>,<NRf>,<NRf>}`
`:AOUTput[:NORMal]:IRTime?`
{<NRf>, <NRf>, <NRf>} = 0, 0, 0 to 10000, 0, 0
1st <NRf> = 0 to 10000 (hour)
2nd <NRf> = 0 to 59 (minute)
3rd <NRf> = 0 to 59 (second)
Example `:AOUTPUT:NORMAL:IRTIME 1,0,0`
`:AOUTPUT:NORMAL:IRTIME? ->`
`:AOUTPUT:NORMAL:IRTIME 1,0,0`

:AOUTput[:NORMal]:MODE<x>

Function Sets the method of setting the rated value for the D/A output items or queries the current setting.

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

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

:AOUTput[:NORMal]:RATE<x>

Function Manually sets the rated maximum and minimum values for the D/A output items or queries the current setting.

Syntax
```
:AOUTput[:NORMal]:MODE<x> {<NRf>,
<NRf>}
:AOUTput[:NORMal]:MODE<x>?
```
<x> = 1 to 20 (output channel)
<NRf> = –9.999E+30 to 9.999E+30 (rated value)

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

Description
- Set the maximum value and then the minimum value.
- This setting is valid when the method of setting the rated value (:AOUTput[:NORMal]:MODE<x>) is set to FIXed.

6.4 CBCycle Group

The CBCycle group contains commands related to Cycle by Cycle measurement functions.
These commands allow you to enter and query the same settings that are available under ITEM in the “CbyC Items” menu and under FORM in the “CbyC Form” menu on the front panel.
Note that the commands in this group are only available with the Cycle by Cycle function (/CC option).

:CBCycle?

Function Queries all settings related to the Cycle by Cycle measurement function.

Syntax
```
:CBCycle?
```

Example
```
:CBCYCLE? -> :CBCYCLE:SYNCHRONIZE:
SOURCE U1;SLOPE RISE;:CBCYCLE:
TRIGGER:MODE AUTO;SOURCE U1;
SLOPE RISE;LEVEL 0.0;:CBCYCLE:
COUNT 100;TIMEOUT 10;FILTER:LINE:
ELEMENT1 50.0E+03;
ELEMENT2 50.0E+03;
ELEMENT3 50.0E+03;ELEMENT4 50.0E+03
```

:CBCycle:COUNt

Function Sets the number of cycles for Cycle by Cycle measurement or queries the current setting.

Syntax
```
:CBCycle:COUNt {<NRf>}
:CBCycle:COUNt?
```
<NRf> = 1 to 3000 (number of measured cycles)

Example
```
:CBCYCLE:COUNT 100
:CBCYCLE:COUNT? ->
:CBCYCLE:COUNT 100
```

:CBCycle:DISPlay?

Function Queries all settings related to the Cycle by Cycle display.

Syntax
```
:CBCycle:DISPlay?
```

Example
```
:CBCYCLE:DISPLAY? ->
:CBCYCLE:DISPLAY:ITEM1 FREQ;
ITEM2 U,1;ITEM3 I,1;ITEM4 P,1;
ITEM5 S,1;CURSOR 1
```

:CBCycle:DISPlay:CURSor

Function Sets the cursor position of the Cycle by Cycle display or queries the current setting.

Syntax
```
:CBCycle:DISPlay:CURSor {<NRf>}
:CBCycle:DISPlay:CURSor?
```
<NRf> = 1 to 3000 (cursor position)

Example
```
:CBCYCLE:DISPLAY:CURSOR 1
:CBCYCLE:DISPLAY:CURSOR? ->
:CBCYCLE:DISPLAY:CURSOR 1
```

Description
- Specifies the cursor position by the cycle number.
- You can make the same setting or query with the “:DISPlay:CBCycle:CURSor” command.

:CBCycle:DISPlay:ITEM<x>

Function Sets the displayed items (function and element) of the Cycle by Cycle display or queries the current setting.

Syntax
```
:CBCycle:DISPlay:
ITEM<x> {<Function>,<Element>}
:CBCycle:DISPlay:ITEM<x>?
```
<x> = 1 to 5 (item number)
<Function> = {FREQ|U|I|P|S|Q|LAMBda|SPEed|TORQue|PM}
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)

Example
```
:CBCYCLE:DISPLAY:ITEM1 U,1
:CBCYCLE:DISPLAY:ITEM1? ->
:CBCYCLE:DISPLAY:ITEM1 U,1
```

Description
- When <Function> = {FREQ|SPEed|TORQue|PM}, <Element> need not be specified.
  <Element> is omitted from the response.
- When <Element> is omitted, Element 1 is set.
- {SPEed|TORQue|PM} is only available with the motor version (-MV).
- You can make the same setting or query with the “:DISPlay:CBCycle:ITEM<x>” command.

:CBCycle:DISPlay:PAGE

Function Sets the number of the displayed page of the Cycle by Cycle display or queries the current setting.

Syntax
```
:CBCycle:DISPlay:PAGE {<NRf>}
:CBCycle:DISPlay:PAGE?
```
<NRf> = 1 to 150 (page number)

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

Description
- When a page number is set, the cursor moves to the top of the specified page.
- You can make the same setting or query with the “:DISPlay:CBCycle:CURSor” command.

:CBCycle:FILTer?

Function Queries all settings related to the filter for Cycle by Cycle measurement.

```
Syntax   :CBCycle:FILTer?
Example  :CBCYCLE:FILTER? ->
         :CBCYCLE:FILTER:LINE:
         ELEMENT1 50.0E+03;
         ELEMENT2 50.0E+03;
         ELEMENT3 50.0E+03;ELEMENT4 50.0E+03
```

:CBCycle:FILTer:LINE?

Function Queries all settings related to the line filter for Cycle by Cycle measurement.

```
Syntax   :CBCycle:FILTer:LINE?
Example  :CBCYCLE:FILTER:LINE? ->
         :CBCYCLE:FILTER:LINE:
         ELEMENT1 50.0E+03;
         ELEMENT2 50.0E+03;
         ELEMENT3 50.0E+03;ELEMENT4 50.0E+03
```

:CBCycle:FILTer[:LINE][:ALL]

Function Collectively sets the line filters of all elements for Cycle by Cycle measurement

```
Syntax   :CBCycle:FILTer[:LINE][:ALL] {OFF|
         <frequency>}
```

OFF = Line filter OFF

<frequency> = 500 Hz, 5.5 kHz, 50 kHz (line filter ON, cutoff frequency)

```
Example  :CBCYCLE:FILTER:LINE:ALL 50KHZ
```

Description Does not set line filters for motor input.

:CBCycle:FILTer[:LINE]:ELEMent<x>

Function Sets the line filter of individual elements for Cycle by Cycle measurement or queries the current setting

```
Syntax   :CBCycle:FILTer[:LINE]:
         ELEMent<x> {OFF|<frequency>}
         :CBCycle:FILTer[:LINE]:ELEMent<x>?
```

<x> = 1 to 4 (element)

OFF = Line filter OFF

<frequency> = 500 Hz, 5.5 kHz, 50 kHz (line filter ON, cutoff frequency)

```
Example  :CBCYCLE:FILTER:LINE:ELEMENT1 50KHZ
         :CBCYCLE:FILTER:LINE:ELEMENT1? ->
         :CBCYCLE:FILTER:LINE:
         ELEMENT1 50.0E+03
```

:CBCycle:FILTer[:LINE]:MOTor

Function Sets the motor input line filters for Cycle by Cycle measurement or queries the current setting

```
Syntax   :CBCycle:FILTer[:LINE]:MOTor {OFF|
         <frequency>}
         :CBCycle:FILTer[:LINE]:MOTor?
```

OFF = Line filter OFF

<frequency> = 100 Hz, 50 kHz (line filter ON, cutoff frequency)

Description Available only with the motor version (-MV).

:CBCycle:RESet

Function Resets Cycle by Cycle measurement.

```
Syntax   :CBCycle:RESet
Example  :CBCYCLE:RESET
```

:CBCycle:STARt

Function Starts Cycle by Cycle measurement.

```
Syntax   :CBCycle:STARt
Example  :CBCYCLE:START
```

:CBCycle:STATe?

Function Queries the Cycle by Cycle measurement status.

```
Syntax   :CBCycle:STATe?
Example  :CBCYCLE:STATE? -> RESET
```

Description The contents of the response are as follows:

RESet = Reset status
STARt = Measuring
COMPlete = Measurement finished and measured results displayed
TIMEout = Timeout
ERRFreq = Measurement finished (frequency measurement error occurred)

:CBCycle:SYNChronize?

Function Queries all settings related to the synchronization source for Cycle by Cycle measurement.

```
Syntax   :CBCycle:SYNChronize?
Example  :CBCYCLE:SYNCHRONIZE? ->
         :CBCYCLE:SYNCHRONIZE:SOURCE U1;
         SLOPE RISE
```

:CBCycle:SYNChronize:SLOPe

Function Sets the slope of the synchronization source of Cycle by Cycle measurement or queries the current setting.

```
Syntax   :CBCycle:SYNChronize:SLOPe {RISE|
         FALL}
         :CBCycle:SYNChronize:SLOPe?
Example  :CBCYCLE:SYNCHRONIZE:SLOPE RISE
         :CBCYCLE:SYNCHRONIZE:SLOPE? ->
         :CBCYCLE:SYNCHRONIZE:SLOPE RISE
```

:CBCycle:SYNChronize:SOURce

Function Sets the synchronization source for Cycle by Cycle measurement or queries the current setting.

Syntax
```
:CBCycle:SYNChronize:SOURce {U<x>|I<x>|EXTernal}
:CBCycle:SYNChronize:SOURce?
```
<x> = 1 to 4 (element)
EXTernal = External clock input (Ext Clk)

Example
```
:CBCYCLE:SYNCHRONIZE:SOURCE U1
:CBCYCLE:SYNCHRONIZE:SOURCE? ->
:CBCYCLE:SYNCHRONIZE:SOURCE U1
```

:CBCycle:TIMEout

Function Sets the timeout value for Cycle by Cycle measurement or queries the current setting.

Syntax
```
:CBCycle:TIMEout {<NRf>}
:CBCycle:TIMEout?
```
<NRf> = 0 to 3600 (seconds)
(0 = No timeout)

Example
```
:CBCYCLE:TIMEOUT 10
:CBCYCLE:TIMEOUT? ->
:CBCYCLE:TIMEOUT 10
```

:CBCycle:TRIGger?

Function Queries all settings related to triggers or queries the current setting.

Syntax
```
:CBCycle:TRIGger?
```

Example
```
:CBCYCLE:TRIGGER? ->
:CBCYCLE:TRIGGER:MODE AUTO;
SOURCE U1;SLOPE RISE;LEVEL 0.0
```

Description This is the same query as with the ":DISPlay:WAVE:TRIGger?" command.

:CBCycle:TRIGger:LEVel

Function Sets the trigger level or queries the current setting.

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

Example
```
:CBCYCLE:TRIGGER:LEVEL 0
:CBCYCLE:TRIGGER:LEVEL? ->
:CBCYCLE:TRIGGER:LEVEL 0.0
```

Description This is the same setting or query as with the ":DISPlay:WAVE:TRIGger:LEVel?" command.

:CBCycle:TRIGger:MODE

Function Sets the trigger mode or queries the current setting.

Syntax
```
:CBCycle:TRIGger:MODE {AUTO|NORMal}
:CBCycle:TRIGger:MODE?
```

Example
```
:CBCYCLE:TRIGGER:MODE AUTO
:CBCYCLE:TRIGGER:MODE? ->
:CBCYCLE:TRIGGER:MODE AUTO
```

Description This is the same setting or query as with the ":DISPlay:WAVE:TRIGger:MODE" command.

:CBCycle:TRIGger:SLOPe

Function Sets the trigger slope or queries the current setting.

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

Example
```
:CBCYCLE:TRIGGER:SLOPE RISE
:CBCYCLE:TRIGGER:SLOPE? ->
:CBCYCLE:TRIGGER:SLOPE RISE
```

Description This is the same setting or query as with the ":DISPlay:WAVE:TRIGger:SLOPe" command.

:CBCycle:TRIGger:SOURce

Function Sets the trigger source or queries the current setting.

Syntax
```
:CBCycle:TRIGger:SOURce {U<x>|I<x>|EXTernal}
:CBCycle:TRIGger:SOURce?
```
<x> = 1 to 4 (element)
EXTernal = External trigger input (Ext Clk)

Example
```
:CBCYCLE:TRIGGER:SOURCE U1
:CBCYCLE:TRIGGER:SOURCE? ->
:CBCYCLE:TRIGGER:SOURCE U1
```

Description This is the same setting or query as with the ":DISPlay:WAVE:TRIGger:SOURce" command.

6.5 COMMunicate Group

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

:COMMunicate?

Function Queries all settings related to communications.

Syntax `:COMMunicate?`

Example
```
:COMMUNICATE? ->
:COMMUNICATE:HEADER 1;OPSE 96;
OVERLAP 96;VERBOSE 1
```

:COMMunicate:HEADer

Function Sets whether to add a header to the response to a query (example `DISPLAY:MODE NUMERIC`) or not add the header (example `NUMERIC`).

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

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

:COMMunicate:LOCKout

Function Sets or clears local lockout.

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

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

Description This command is dedicated to the optional RS-232, USB, or Ethernet interface. An interface message is available for the GP-IB interface.

:COMMunicate:OPSE (Operation Pending Status Enable register)

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

Syntax
```
:COMMunicate:OPSE <Register>
:COMMunicate:OPSE?
```
<Register> = 0 to 65535, see the command diagram for `:COMMunicate:WAIT?` on page 6-25.

Example
```
:COMMUNICATE:OPSE 65535
:COMMUNICATE:OPSE? ->
:COMMUNICATE:OPSE 96
```

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

:COMMunicate:OPSR? (Operation Pending Status Register)

Function Queries the value of the operation pending status register.

Syntax `:COMMunicate:OPSR?`

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

Description For details on the operation pending status register, see the figure for the `:COMMunicate:WAIT?` command (page 6-25).

:COMMunicate:OVERlap

Function Sets the commands that will operate as overlap commands or queries the current setting.

Syntax
```
:COMMunicate:OVERlap <Register>
:COMMunicate:OVERlap?
```
<Register> = 0 to 65535, see the command diagram for `:COMMunicate:WAIT?` on page 6-25.

Example
```
:COMMUNICATE:OVERLAP 65535
:COMMUNICATE:OVERLAP? ->
:COMMUNICATE:OVERLAP 96
```

Description
- In the above example, all bits are set to 1 to make all overlap commands applicable. However, bits fixed to 0 are not set to 1. Thus, the response to the query indicates 1 for bits 5 and 6 only.
- For the description regarding how to synchronize the program using `COMMunicate:OVERlap`, see page 5-7.
- In the above example, bits 5 and 6 are set to 1 to make all overlap commands applicable (see the figure for the `:COMMunicate:WAIT?` command (page 6-25)).

:COMMunicate:REMote

Function Sets remote or local. ON is remote mode.

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

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

Description This command is dedicated to the optional RS-232, USB, or Ethernet interface. An interface message is available for the GP-IB interface.

:COMMunicate:STATus?

Function Queries line-specific status.

Syntax `:COMMunicate:STATus?`

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

Description The meaning of each status bit is as follows:

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

The value 0 is always returned for the optional USB or Ethernet interface.

The status bit is set when the corresponding cause occurs and cleared when it is read.

:COMMunicate:VERBose

Function Sets whether to return the response to a query using full spelling (example `:INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03`) or using abbreviation (example `:VOLT:RANG:ELEM 1.000E+03`).

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

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

:COMMunicate:WAIT

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

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

Example `:COMMUNICATE:WAIT 1`

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

:COMMunicate:WAIT?

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

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

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

Operation pending status register/Overlap enable register

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

When bit 5 (PRN) = 1:
Built-in printer operation not complete

When bit 6 (ACS) = 1:
Access to the medium not complete.

6.6 CURSor Group

The commands in this group deal with cursor measurements. You can make the same settings and inquiries as when CURSOR (SHIFT+MEASURE) on the front panel is used.

:CURSor?

Function Queries all settings related to the cursor measurement.

Syntax `:CURSor?`

Example
```
:CURSOR? -> :CURSOR:WAVE:STATE 0;
TRACE1 U1;TRACE2 I1;PATH MAX;
POSITION1 10.0E-03;
POSITION2 40.0E-03;:CURSOR:BAR:
STATE 0;POSITION1 1;POSITION2 15;:
CURSOR:TREND:STATE 0;TRACE1 1;
TRACE2 2;POSITION1 100;
POSITION2 900
```

:CURSor:BAR?

Function Queries all settings related to the cursor measurement of the bar graph display.

Syntax `:CURSor:BAR?`

Example
```
:CURSOR:BAR? -> :CURSOR:BAR:
STATE 1;POSITION1 1;POSITION2 15
```

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:CURSor:BAR:POSition<x>

Function Sets the cursor position (order) on the bar graph display or queries the current setting.

Syntax `:CURSor:BAR:POSition<x> {<NRf>}`
`:CURSor:BAR:POSition<x>?`
<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 0 to 100

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:CURSor:BAR[:STATe]

Function Turns ON/OFF the cursor display on the bar graph display or queries the current setting.

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

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

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

Function Queries the cursor measurement value on the bar graph display.

Syntax `:CURSor:BAR:{Y<x>|DY}?`
Y<x> = Y-axis value at the cursor position (Y1 = Y1+, Y2+, Y3+ Y2 = Y1x, Y2x, Y3x)
DY = Y-axis value between cursors (ΔY1, ΔY2, and ΔY3)
<x> = 1, 2 (1 = C1 +, 2 = C2 x)

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- When multiple bar graphs are displayed, the cursor measurement values of each bar graph are returned in order.
- If the cursor display is not turned ON on the bar graph, “NAN (Not A Number)” is returned.

:CURSor:FFT?

Function Queries all settings related to the cursor measurement on the FFT waveform display.

Syntax `:CURSor:FFT?`

Example
```
:CURSOR:FFT? -> :CURSOR:FFT:
STATE 0;TRACE1 FFT1;TRACE2 FFT2;
POSITION1 100;POSITION2 900
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:CURSor:FFT:POSition<x>

Function Sets the cursor position on the FFT waveform display or queries the current setting.

Syntax `:CURSor:FFT:POSition<x> {<NRf>}`
`:CURSor:FFT:POSition<x>?`
<x> = 1, 2(1 = C1 +, 2 = C2 x)
<NRf> = 0 to 1001

Example
```
:CURSOR:FFT:POSITION1 20
:CURSOR:FFT:POSITION1? ->
:CURSOR:FFT:POSITION1 20
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:CURSor:FFT[:STATe]

Function Turns ON/OFF the cursor display on the FFT waveform display or queries the current setting.

```
Syntax       :CURSor:FFT[:STATe] {<Boolean>}
             :CURSor:FFT:STATe?
Example      :CURSOR:FFT:STATE OFF
             :CURSOR:FFT:STATE? ->
             :CURSOR:FFT:STATE 0
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:CURSor:FFT:TRACe<x>

Function Sets the cursor target on the FFT waveform display or queries the current setting.

```
Syntax       :CURSor:FFT:TRACe<x> {FFT<x>}
             :CURSor:FFT:TRACe<x>?
```

<x> of TRACe<x> = 1 or 2 (1 = C1 +, 2 = C2 x)
<x> of FFT<x> = 1 or 2 (FFT)

```
Example      :CURSOR:FFT:TRACE1 FFT1
             :CURSOR:FFT:TRACE1? ->
             :CURSOR:FFT:TRACE1 FFT1
```

Description This command is valid only on models with the advanced computation function (/G6 option).

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

Function Queries the cursor measurement value on the FFT waveform display.

```
Syntax       :CURSor:FFT:{X<x>|DX|Y<x>|DY}?
```

X<x> = X-axis value of the cursor position (X1 = X+, X2 = Xx)
DX = X-axis value between cursors (ΔX)
Y<x> = Y-axis value of the cursor position (Y1 = Y+, Y2 = Yx)
DY = Y-axis value between cursors (ΔY)
<x> = 1, 2(1 = C1 +, 2 = C2 x)

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

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- If the cursor display is not turned ON in the FFT waveform display, “NAN (Not A Number)” is returned.

:CURSor:TRENd?

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

```
Syntax       :CURSor:TRENd?
Example      :CURSOR:TREND? -> :CURSOR:TREND:
             STATE 1;TRACE1 1;TRACE2 2;
             POSITION1 100;POSITION2 900
```

:CURSor:TRENd:POSition<x>

Function Sets the cursor position on the trend display or queries the current setting.

```
Syntax       :CURSor:TRENd:POSition<x> {<NRf>}
             :CURSor:TRENd:POSition<x>?
```

<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 0 to 1001

```
Example      :CURSOR:TREND:POSITION1 10
             :CURSOR:TREND:POSITION1? ->
             :CURSOR:TREND:POSITION1 10
```

:CURSor:TRENd[:STATe]

Function Turns ON/OFF the cursor display on the trend display or queries the current setting.

```
Syntax       :CURSor:TRENd[:STATe] {<Boolean>}
             :CURSor:TRENd:STATe?
Example      :CURSOR:TREND:STATE ON
             :CURSOR:TREND:STATE? ->
             :CURSOR:TREND:STATE 1
```

:CURSor:TRENd:TRACe<x>

Function Sets the cursor target on the trend display or queries the current setting.

```
Syntax       :CURSor:TRENd:TRACe<x> {<NRf>}
             :CURSor:TRENd:TRACe<x>?
```

<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 1 to 16 (T1 to T16)

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

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

Function Queries the cursor measurement value on the trend display.

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

X<x> = Trend time string of the cursor position (X1 = D+, X2 = Dx)
Y<x> = Y-axis value of the cursor position (Y1 = Y+, Y2 = Yx)
DY = Y-axis value between cursors (ΔY)
<x> = 1, 2 (1 = C1 +, 2 = C2 x)

```
Example      :CURSOR:TREND:X1? ->
             "2005/01/01 12:34:56"
             :CURSOR:TREND:Y1? -> 78.628E+00
```

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

:CURSor:WAVE?

Function Queries all settings related to the cursor measurement on the waveform display.

Syntax
```
:CURSor:WAVE?
```

Example
```
:CURSOR:WAVE? -> :CURSOR:WAVE:
STATE 1;TRACE1 U1;TRACE2 I1;
PATH MAX;POSITION1 10.0E-03;
POSITION2 40.0E-03
```

:CURSor:WAVE:PATH

Function Sets the cursor path on the waveform display or queries the current setting.

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

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

:CURSor:WAVE:POSition<x>

Function Sets the cursor position on the waveform display or queries the current setting.

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

<Time> = 0 to 20.00 s

Example
```
:CURSOR:WAVE:POSITION1 10MS
:CURSOR:WAVE:POSITION1? ->
:CURSOR:WAVE:POSITION1 10.0E-03
```

Description The selectable range and resolution of <Time> is determined by the Time/div value of the waveform (`:DISPlay:WAVE:TDIV`).

:CURSor:WAVE[:STATe]

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

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

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

:CURSor:WAVE:TRACe<x>

Function Sets the cursor target on the waveform display or queries the current setting.

Syntax
```
:CURSor:WAVE:TRACe<x> {U<x>|I<x>|
SPEed|TORQue|MATH<x>}
:CURSor:WAVE:TRACe<x>?
```
<x> of TRACe<x> = 1 or 2 (1 = C1 +, 2 = C2 x)

<x> of U<x>, I<x> = 1 to 4

<x> of MATH<x> = 1 to 2 (MATH)

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

Description
- `{SPEed|TORQue}` are valid only on the motor version (-MV).
- MATH<x> is valid only on models with the advanced computation function (/G6 option).

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

Function Queries the cursor measurement value on the waveform display.

Syntax
```
:CURSor:WAVE:{X<x>|DX|PERDt|Y<x>|
DY}?
```
X<x> = X-axis value of the cursor position (X1 = X+, X2 = Xx)

DX = X-axis value between cursors (ΔX)

PERDt = 1/DT (1/ΔX) value between cursors

Y<x> = Y-axis value of the cursor position (Y1 = Y+, Y2 = Yx)

DY = Y-axis value between cursors (ΔY)

<x> = 1, 2 (1 = C1 +, 2 = C2 x)

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

Description If the cursor display is not turned ON in the waveform display, “NAN (Not A Number)” is returned.

6.7 DISPlay Group

The commands in this group deal with the screen display.
You can make the same settings and inquiries as when the keys in the DISPLAY area and the ITEM & ELEMENT area on the front panel are used.

:DISPlay?

Function Queries all settings related to the screen display.

Syntax `:DISPlay?`

Example
- Example when the display mode (`:DISPlay:MODE`) is “NUMeric (numeric display)”
  `:DISPLAY? -> :DISPLAY:MODE NUMERIC;` (Response to “`:DISPlay:NUMeric?`” with the first “`:DISPLAY:`” section removed)`;:DISPLAY:INFORMATION: STATE 0;`
  `PAGE 1`
- Example when the display mode (`:DISPlay:MODE`) is “WAVE (waveform display)”
  `:DISPLAY? -> :DISPLAY:MODE WAVE;` (Response to “`:DISPlay:WAVE?`” with the first “`:DISPLAY:`” section removed)`;:DISPLAY:INFORMATION: STATE 0;`
  `PAGE 1`
- Example when the display mode (`:DISPlay:MODE`) is “NWAVe”
  `:DISPLAY? -> :DISPLAY:MODE NWAVE;` (Response to “`:DISPlay:NUMeric?`” with the first “`:DISPLAY:`” section removed)`;`(same as the response to “`:DISPlay:WAVE?`”)`;:DISPLAY: INFORMATION:STATE 0;PAGE 1`

Description Returns all settings corresponding to the display mode (`:DISPlay:MODE`).

:DISPlay:BAR?

Function Queries all settings related to the bar graph.

Syntax `:DISPlay:BAR?`

Example `:DISPLAY:BAR? -> :DISPLAY:BAR: FORMAT SINGLE;ITEM1 U,1;ITEM2 I,1; ITEM3 P,1;ORDER 1,100`

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:BAR:FORMat

Function Sets the display format of the bar graph or queries the current setting.

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

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:BAR:ITEM<x>

Function Sets the bar graph item (function and element) or queries the current setting.

Syntax `:DISPlay:BAR:ITEM<x> {<Function>, <Element>}`
`:DISPlay:BAR:ITEM<x>?`
<x> = 1 to 3 (item number)
<Function> = `{U|I|P|S|Q|LAMBda|...}`
(See the function selection list (2) on page 6-46.)
<Element> = 1 to 4

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:BAR:ORDer

Function: Sets the start and end orders of the bar graph or queries the current setting.

Syntax
```
:DISPlay:BAR:ORDer {<NRf>,<NRf>}
:DISPlay:BAR:ORDer?
```
1st <NRf> = 0 to 90 (start order to be displayed)
2nd <NRf> = 10 to 100 (end order to be displayed)

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- Set the start order and then the end order.
- Set the end order so that it is greater than or equal to (start order + 10).

:DISPlay:CBCycle?

Function: Queries all settings related to the Cycle by Cycle display.

Syntax
```
:DISPlay:CBCycle?
```

Example
```
:DISPLAY:CBCYCLE? -> :DISPLAY:
CBCYCLE:ITEM1 FREQ;ITEM2 U,1;
ITEM3 I,1;ITEM4 P,1;ITEM5 S,1;
CURSOR 1
```

Description: Only available with the Cycle by Cycle function (/CC option).

:DISPlay:CBCycle:CURSor

Function: Sets the cursor position of the Cycle by Cycle display or queries the current setting.

Syntax
```
:DISPlay:CBCycle:CURSor {<NRf>}
:DISPlay:CBCycle:CURSor?
```
<NRf> = 1 to 3000 (cursor position)

Example
```
:DISPLAY:CBCYCLE:CURSOR 1
:DISPLAY:CBCYCLE:CURSOR? ->
:DISPLAY:CBCYCLE:CURSOR 1
```

Description
- Only available with the Cycle by Cycle function (/CC option).
- Specifies the cursor position by the cycle number.

:DISPlay:CBCycle:ITEM<x>

Function: Sets the displayed items (function and element) of the Cycle by Cycle display or queries the current setting.

Syntax
```
:DISPlay:CBCycle:
ITEM<x> {<Function>,<Element>}
:DISPlay:CBCycle:ITEM<x>?
```
<x> = 1 to 5 (item number)
<Function> = {FREQ|U|I|P|S|Q|LAMBda|SPEed|TORQue|PM}
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)

Example
```
:DISPLAY:CBCYCLE:ITEM1 U,1
:DISPLAY:CBCYCLE:ITEM1? ->
:DISPLAY:CBCYCLE:ITEM1 U,1
```

Description
- Only available with the Cycle by Cycle function (/CC option).
- When <Function> = {FREQ|SPEed|TORQue|PM}, <Element> need not be specified. <Element> is omitted from the response.
- When <Element> is omitted, Element 1 is set.
- {SPEed|TORQue|PM} is only available with the motor version (-MV).

:DISPlay:CBCycle:PAGE

Function: Sets the number of the displayed page of the Cycle by Cycle display or queries the current setting.

Syntax
```
:DISPlay:CBCycle:PAGE {<NRf>}
:DISPlay:CBCycle:PAGE?
```
<NRf> = 1 to 150 (page number)

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

Description
- Only available with the Cycle by Cycle function (/CC option).
- When a page number is set, the cursor moves to the top of the specified page.

:DISPlay:FFT?

Function: Queries all settings related to the FFT waveform display.

Syntax
```
:DISPlay:FFT?
```

Example
```
:DISPLAY:FFT? -> :DISPLAY:FFT:
FORMAT SINGLE;POINT 20000;
WINDOW RECTANGLE;SCOPE 0,10000;
VSCALE LOG;SPECTRUM LINE;FFT1:
STATE 1;OBJECT U1;LABEL "FFT1";:
DISPLAY:FFT:FFT2:STATE 1;OBJECT I1;
LABEL "FFT2"
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:FFT<x>?

Function Queries all settings related to the FFT waveform.

Syntax `:DISPlay:FFT:FFT<x>?`
<x> = 1, 2 (FFT)

Example `:DISPLAY:FFT:FFT1? -> :DISPLAY:FFT:FFT1:STATE 1;OBJECT U1;LABEL "FFT1"`

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:FFT<x>:LABel

Function Sets the label of the FFT waveform or queries the current setting.

Syntax `:DISPlay:FFT:FFT<x>:LABel {<String>}`
`:DISPlay:FFT:FFT<x>:LABel?`
<x> = 1, 2 (FFT)
<String> = Up to 8 characters

Example
```
:DISPLAY:FFT:FFT1:LABEL "FFT1"
:DISPLAY:FFT:FFT1:LABEL? ->
:DISPLAY:FFT:FFT1:LABEL "FFT1"
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:FFT<x>:OBJect

Function Sets the source waveform of the FFT computation or queries the current setting.

Syntax `:DISPlay:FFT:FFT<x>:OBJect {U<x>|I<x>|P<x>|PA|PB|Q<x>|QA|QB|SPEed|TORQue}`
`:DISPlay:FFT:FFT<x>:OBJect?`
<x> of FFT<x> = 1, 2 (FFT)
<x> of U<x>, I<x>, P<x>, Q<x> = 1 to 4 (element)
PA, QA = PΣA, QΣA (only on models with 2 to 4 elements)
PB, QB = PΣB, QΣB (only on models with 4 elements)

Example
```
:DISPLAY:FFT:FFT1:OBJECT U1
:DISPLAY:FFT:FFT1:OBJECT? ->
:DISPLAY:FFT:FFT1:OBJECT U1
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- `{SPEed|TORQue}` are valid only on the motor version (-MV).

:DISPlay:FFT:FFT<x>[:STATe]

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

Syntax `:DISPlay:FFT:FFT<x>[:STATe] {<Boolean>}`
`:DISPlay:FFT:FFT<x>:STATe?`
<x> = 1, 2 (FFT)

Example
```
:DISPLAY:FFT:FFT1:STATE ON
:DISPLAY:FFT:FFT1:STATE? ->
:DISPLAY:FFT:FFT1:STATE 1
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:FORMat

Function Sets the display format of the FFT waveform or queries the current setting.

Syntax `:DISPlay:FFT:FORMat {SINGle|DUAL}`
`:DISPlay:FFT:FORMat?`

Example
```
:DISPLAY:FFT:FORMAT SINGLE
:DISPLAY:FFT:FORMAT? ->
:DISPLAY:FFT:FORMAT SINGLE
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:POINt

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

Syntax `:DISPlay:FFT:POINt {<NRf>}`
`:DISPlay:FFT:POINt?`
<NRf> = 20000, 200000

Example
```
:DISPLAY:FFT:POINT 20000
:DISPLAY:FFT:POINT? ->
:DISPLAY:FFT:POINT 20000
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:SCOPe

Function Sets the display range of the FFT waveform or queries the current setting.

Syntax `:DISPlay:FFT:SCOPe {<NRf>,<NRf>}`
`:DISPlay:FFT:SCOPe?`
1st <NRf> = 0 to 99990 (display start point)
2nd <NRf> = 10 to 100000 (display end point)

Example
```
:DISPLAY:FFT:SCOPE 0,10000
:DISPLAY:FFT:SCOPE? ->
:DISPLAY:FFT:SCOPE 0,10000
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- Set the start point and then the end point.
- Set the end point so that it is greater than or equal to (start point + 10).

:DISPlay:FFT:SPECtrum

Function Sets the display spectrum format of the FFT waveform or queries the current setting.

Syntax
```
:DISPlay:FFT:SPECtrum {LINE|BAR}
:DISPlay:FFT:SPECtrum?
```

Example
```
:DISPLAY:FFT:SPECTRUM LINE
:DISPLAY:FFT:SPECTRUM? ->
:DISPLAY:FFT:SPECTRUM LINE
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:VSCale

Function Sets the display scale of the vertical axis of the FFT waveform or queries the current setting.

Syntax
```
:DISPlay:FFT:VSCale {LINear|LOG}
:DISPlay:FFT:VSCale?
```

Example
```
:DISPLAY:FFT:VSCALE LOG
:DISPLAY:FFT:VSCALE? ->
:DISPLAY:FFT:VSCALE LOG
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FFT:WINDow

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

Syntax
```
:DISPlay:FFT:WINDow {RECTangle|
HANNing|FLATtop}
:DISPlay:FFT:WINDow?
```

Example
```
:DISPLAY:FFT:WINDOW RECTANGLE
:DISPLAY:FFT:WINDOW? ->
:DISPLAY:FFT:WINDOW RECTANGLE
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:FLICker?

Function Queries all settings related to flicker measurement display.

Syntax
```
:DISPlay:FLICker?
```

Example
```
:DISPLAY:FLICKER? ->
:DISPLAY:FLICKER:ELEMENT 1;PERIOD 1
```

Description Only available with the flicker measurement function (/FL option).

:DISPlay:FLICker:ELEMent

Function Sets the element to be displayed for flicker measurement display or queries the current setting.

Syntax
```
:DISPlay:FLICker:ELEMent {<NRf>}
:DISPlay:FLICker:ELEMent?
```
<NRf> = 1 to 4 (element)

Example
```
:DISPLAY:FLICKER:ELEMENT 1
:DISPLAY:FLICKER:ELEMENT? ->
:DISPLAY:FLICKER:ELEMENT 1
```

Description Only available with the flicker measurement function (/FL option).

:DISPlay:FLICker:PAGE

Function Sets the page numbers to be displayed for flicker measurement display or queries the current setting.

Syntax
```
:DISPlay:FLICker:PAGE {<NRf>}
:DISPlay:FLICker:PAGE?
```
<NRf> = 1 to 9 (page number)

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

Description Only available with the flicker measurement function (/FL option).

:DISPlay:FLICker:PERiod

Function Sets the display observation period number for flicker measurement display or queries the current setting.

Syntax
```
:DISPlay:FLICker:PERiod {<NRf>}
:DISPlay:FLICker:PERiod?
```
<NRf> = 1 to 99 (observation period number)

Example
```
:DISPLAY:FLICKER:PERIOD 1
:DISPLAY:FLICKER:PERIOD? ->
:DISPLAY:FLICKER:PERIOD 1
```

Description Only available with the flicker measurement function (/FL option).

:DISPlay:INFOrmation?

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

Syntax
```
:DISPlay:INFOrmation?
```

Example
```
:DISPLAY:INFORMATION? ->
:DISPLAY:INFORMATION:STATE 0;PAGE 1
```

:DISPlay:INFOrmation:PAGE

Function Sets the page number of the display of setup parameter list or queries the current setting.

Syntax
```
:DISPlay:INFOrmation {<NRf>}
:DISPlay:INFOrmation?
```
<NRf> = 1 to 4 (page number)

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

:DISPlay:INFOrmation[:STATe]

Function Turns ON/OFF the display of the setup parameter list or queries the current setting.

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

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

:DISPlay:MATH?

Function: Queries all settings related to the computed waveform display.

Syntax
```
:DISPlay:MATH?
```

Example
```
:DISPLAY:MATH? -> :DISPLAY:MATH:
MATH1:EXPRESSION "U1*I1";SCALING:
MODE AUTO;CENTER 0.0000E+00;SDIV
25.000E+00;:DISPLAY:MATH:MATH1:
UNIT "W":LABEL "Math1";:DISPLAY:
MATH:MATH2:EXPRESSION "ABS(U1)";
SCALING:MODE AUTO;
CENTER 0.0000E+00;SDIV 25.000E+00;:
DISPLAY:MATH:MATH2:UNIT "V":
LABEL "Math2";:DISPLAY:MATH:
CONSTANT1 1.0000E+00;
CONSTANT2 2.0000E+00;
CONSTANT3 3.0000E+00;
CONSTANT4 4.0000E+00;
CONSTANT5 5.0000E+00;
CONSTANT6 6.0000E+00;
CONSTANT7 7.0000E+00;
CONSTANT8 8.0000E+00
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:CONStant<x>

Function: Sets the constant to be used in the waveform computing equation or queries the current setting.

Syntax
```
:DISPlay:MATH:CONStant<x> {<NRf>}
:DISPlay:MATH:CONStant<x>?
```
<x> = 1 to 8 (K1 to K8)
<NRf> = –9.9999E+30 to 9.9999E+30

Example
```
:DISPLAY:MATH:CONSTANT1 1.0000E+00
:DISPLAY:MATH:CONSTANT1? ->
:DISPLAY:MATH:CONSTANT1 1.0000E+00
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>?

Function: Queries all settings related to the computed waveform.

Syntax
```
:DISPlay:MATH:MATH<x>?
```
<x> = 1, 2 (MATH)

Example
```
:DISPLAY:MATH:MATH1? -> :DISPLAY:
MATH:MATH1:EXPRESSION "U1*I1";
SCALING:MODE AUTO;
CENTER 0.0000E+00;SDIV 25.000E+00;:
DISPLAY:MATH:MATH1:UNIT "W":
LABEL "Math1"
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>:EXPRession

Function: Sets the equation of the waveform computation or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:
EXPRession {<String>}
:DISPlay:MATH:MATH<x>:EXPRession?
```
<x> = 1, 2 (MATH)
<String> = Up to 50 characters

Example
```
:DISPLAY:MATH:MATH1:
EXPRESSION "U1*I1"
:DISPLAY:MATH:MATH1:EXPRESSION? ->
:DISPLAY:MATH:MATH1:
EXPRESSION "U1*I1"
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>:LABel

Function: Sets the label of the computed waveform or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:
LABel {<String>}
:DISPlay:MATH:MATH<x>:LABel?
```
<x> = 1, 2 (MATH)
<String> = Up to 8 characters

Example
```
:DISPLAY:MATH:MATH1:LABEL "Math1"
:DISPLAY:MATH:MATH1:LABEL? ->
:DISPLAY:MATH:MATH1:LABEL "Math1"
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>:SCALing?

Function: Queries all settings related to the scaling of the computed waveform.

Syntax
```
:DISPlay:MATH:MATH<x>:SCALing?
```
<x> = 1, 2 (MATH)

Example
```
:DISPLAY:MATH:MATH1? ->
:DISPLAY:MATH:MATH1:SCALING:
MODE AUTO;CENTER 0.0000E+00;
SDIV 25.000E+00
```

Description: This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>:SCALing:CENTer

Function Sets the center value of the manual scaling of the computed waveform or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:SCALing:
CENTer {<NRf>}
:DISPlay:MATH:MATH<x>:SCALing:
CENTer?
```
<x> = 1, 2 (MATH)
<NRf> = –9.9999E+30 to 9.9999E+30

Example
```
:DISPLAY:MATH:MATH1:SCALING:
CENTER 0.0000E+00
:DISPLAY:MATH:MATH1:SCALING:CENTER?
-> :DISPLAY:MATH:MATH1:SCALING:
CENTER 0.0000E+00
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- This command is valid when the scaling mode of the computed waveform (`:DISPlay:MATH:MATH<x>:SCALing:MODE`) is set to “MANual.”

:DISPlay:MATH:MATH<x>:SCALing:MODE

Function Sets the scaling mode of the computed waveform or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:SCALing:
MODE {AUTO|MANual}
:DISPlay:MATH:MATH<x>:SCALing:MODE?
```
<x> = 1, 2 (MATH)

Example
```
:DISPLAY:MATH:MATH1:SCALING:
MODE AUTO
:DISPLAY:MATH:MATH1:SCALING:MODE?
-> :DISPLAY:MATH:MATH1:SCALING:
MODE AUTO
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MATH:MATH<x>:SCALing:SDIV

Function Sets the scale/division value of the manual scaling of the computed waveform or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:SCALing:
SDIV {<NRf>}
:DISPlay:MATH:MATH<x>:SCALing:SDIV?
```
<x> = 1, 2 (MATH)
<NRf> = –9.9999E+30 to 9.9999E+30

Example
```
:DISPLAY:MATH:MATH1:SCALING:
SDIV 2.5000E+01
:DISPLAY:MATH:MATH1:SCALING:SDIV?
-> :DISPLAY:MATH:MATH1:SCALING:
SDIV 25.000E+00
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- This command is valid when the scaling mode of the computed waveform (`:DISPlay:MATH:MATH<x>:SCALing:MODE`) is set to “MANual.”

:DISPlay:MATH:MATH<x>:UNIT

Function Sets the unit to be added to the result of the waveform computation or queries the current setting.

Syntax
```
:DISPlay:MATH:MATH<x>:
UNIT {<String>}
:DISPlay:MATH:MATH<x>:UNIT?
```
<x> = 1, 2 (MATH)
<String> = Up to 8 characters

Example
```
:DISPLAY:MATH:MATH1:UNIT "W"
:DISPLAY:MATH:MATH1:UNIT? ->
:DISPLAY:MATH:MATH1:UNIT "W"
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:DISPlay:MODE

Function Sets the display mode or queries the current setting.

Syntax
```
:DISPlay:MODE {NUMeric|WAVE|BAR|
VECTor|TRENd|NWAVe|NBAR|NTRend|
WBAR|WTRend|BTRend|MATH|NMATh|FFT|
NFFT|WFFT|IECHarm|CBCycle|FLICker}
:DISPlay:MODE?
```
NUMeric = Displays only the numeric values.
WAVE = Displays only the waveforms.
BAR = Bar graph
VECTor = Vector display
TRENd = Trend
NWAVe = Displays both the numeric values and the waveforms.
NBAR = Displays both the numeric values and the bar graph.
NTRend = Displays both the numeric values and the trends.
WBAR = Displays both the waveforms and the bar graph.
WTRend = Displays both the waveforms and the trends.
BTRend = Displays both the bar graphs and the trends.
MATH = Displays only the waveforms (including the computed waveforms).
NMATh = Displays both numeric values and waveforms (including the computed waveforms).
FFT = Displays only the FFT waveforms.
NFFT = Displays both the numeric values and the FFT waveforms.
WFFT = Displays both the waveforms and the FFT waveforms.
IECHarm = IEC harmonic measurement mode display (numeric values).
CBCycle = Cycle by Cycle mode display (value).
FLICker = Flicker measurement mode display (value).

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

Description
- {BAR|VECTor|NBAR|WBAR|BTRend} are selectable only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- {MATH|NMATh|FFT|NFFT|WFFT|IECHarm} are selectable only on models with the advanced computation function (/G6 option).
- {CBCycle} can only be selected with the Cycle by Cycle function (/CC option).
- {FLICker} can only be selected with the flicker measurement function (/FL option).

:DISPlay:NUMeric?

Function Queries all settings related to the numeric display.

Syntax `:DISPlay:NUMeric?`

Example `:DISPLAY:NUMERIC? ->` (same as the response to "`:DISPlay:NUMeric:NORMal?`")

:DISPlay:NUMeric:NORMal?

Function Queries all settings related to the numeric display.

Syntax `:DISPlay:NUMeric:NORMal?`

Example
- Example in which the numeric display format (`:DISPlay:NUMeric[:NORMal]:FORMat`) is set to "VAL4 (4-value display)"
```
:DISPLAY:NUMERIC:NORMAL? ->
:DISPLAY:NUMERIC:NORMAL:
FORMAT VAL4;VAL4:ITEM1 U,1,TOTAL;
ITEM2 I,1,TOTAL;ITEM3 P,1,TOTAL;
...(omitted)...;ITEM35 ETA3;
ITEM36 ETA4;CURSOR 1
```
- Example in which the numeric display format (`:DISPlay:NUMeric[:NORMal]:FORMat`) is set to "ALL (all display)"
```
:DISPLAY:NUMERIC:NORMAL? ->
:DISPLAY:NUMERIC:NORMAL:
FORMAT ALL;ALL:CURSOR U
```

Description Returns all settings corresponding to the numeric display format (`:DISPlay:NUMeric[:NORMal]:FORMat`).

:DISPlay:NUMeric[:NORMal]:ALL?

Function Queries all settings related to the numeric display (all display).

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

Example
```
:DISPLAY:NUMERIC:NORMAL:ALL? ->
:DISPLAY:NUMERIC:NORMAL:ALL:
CURSOR U
```

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

Function Sets the cursor position on the numeric display (all display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:ALL:
CURSor {<Function>}
:DISPlay:NUMeric[:NORMal]:ALL:
CURSor?
```
<Function> = {U|I|P|S|Q|...} (See the function selection list (1) on page 6-44.)

Example
```
:DISPLAY:NUMERIC:NORMAL:ALL:
CURSOR U
:DISPLAY:NUMERIC:NORMAL:ALL:CURSOR?
-> :DISPLAY:NUMERIC:NORMAL:ALL:
CURSOR U
```

Description Specify the cursor position in terms of the function name.

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

Function Sets the displayed harmonic order on the harmonic measurement function display page of the numeric display (all display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:ALL:
ORDer {<Order>}
:DISPlay:NUMeric[:NORMal]:ALL:
ORDer?
```
<Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 100)

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- This command is valid when the displayed page number (:DISPlay:NUMeric[:NORMal]:ALL:PAGE) on the numeric display (all display) is 6 or 7.

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

Function Sets the page number on the numeric display (all display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:ALL:
PAGE {<NRf>}
:DISPlay:NUMeric[:NORMal]:ALL:PAGE?
```
<NRf> = 1 to 5 (page number)
<NRf> = 1 to 9 (when the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option) is installed)

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

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

:DISPlay:NUMeric[:NORMal]:FORMat

Function Sets the numeric display format or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:
FORMat {VAL4|VAL8|VAL16|ALL|SINGle|
DUAL}
:DISPlay:NUMeric[:NORMal]:FORMat?
```

Example
```
:DISPLAY:NUMERIC:NORMAL:FORMAT VAL4
:DISPLAY:NUMERIC:NORMAL:FORMAT? ->
:DISPLAY:NUMERIC:NORMAL:FORMAT VAL4
```

Description
- The contents of the displayed numeric data are as follows:
  {VAL4|VAL8|VAL16}: Numeric display items are displayed in order by the item number. (The number expresses the number of items that is displayed on a single screen (page).)
  ALL = All functions are displayed in order by element.
  SINGle = One list display item is listed by separating the data into even and odd orders.
  DUAL = Two list display items are listed in order by harmonic order.
- {SINGle|DUAL} are selectable only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:NUMeric[:NORMal]:LIST?

Function Queries all settings related to the numeric display (list display).

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

Example
```
:DISPLAY:NUMERIC:NORMAL:LIST? ->
:DISPLAY:NUMERIC:NORMAL:LIST:
ITEM1 U,1;ITEM2 I,1;CURSOR ORDER;
HEADER 1;ORDER 1
```

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

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

Function Sets the cursor position on the numeric display (list display) or queries the current setting.

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

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

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

Function Sets the cursor position in the header section on the numeric display (list display) or queries the current setting.

Syntax `:DISPlay:NUMeric[:NORMal]:LIST:HEADer {<NRf>}`
`:DISPlay:NUMeric[:NORMal]:LIST:HEADer?`
<NRf> = 1 to 98

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- This command is valid when the cursor position (`:DISPlay:NUMeric[:NORMal]:LIST:CURSor`) on the numeric display (list display) is “HEADer.”

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

Function Sets the displayed items (function and element) on the numeric display (list display) or queries the current setting.

Syntax `:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x> {<Function>,<Element>}`
`:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x>?`
<x> = 1 or 2 (item number)
<Function> = `{U|I|P|S|Q|LAMBda|...}`
(See the function selection list (2) on page 6-46.)
<Element> = `{<NRf>|SIGMA|SIGMB}`
(<NRf> = 1 to 4)

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

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

Function Sets the harmonic order cursor position in the data section on the numeric display (list display) or queries the current setting.

Syntax `:DISPlay:NUMeric[:NORMal]:LIST:ORDer {<NRf>}`
`:DISPlay:NUMeric[:NORMal]:LIST:ORDer?`
<NRf> = –1 to 100 (order)

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- This command is valid when the cursor position (`:DISPlay:NUMeric[:NORMal]:LIST:CURSor`) on the numeric display (list display) is “ORDer.”

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

Function Queries all settings related to the numeric display ({4-value|8-value|16-value} display).

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

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4? ->
:DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 U,1,TOTAL;ITEM2 I,1,TOTAL;
ITEM3 P,1,TOTAL;
ITEM4 LAMBDA,1,TOTAL;
ITEM5 U,2,TOTAL;ITEM6 I,2,TOTAL;
ITEM7 P,2,TOTAL;
ITEM8 LAMBDA,2,TOTAL;
ITEM9 U,3,TOTAL;ITEM10 I,3,TOTAL;
ITEM11 P,3,TOTAL;
ITEM12 LAMBDA,3,TOTAL;
ITEM13 U,4,TOTAL;ITEM14 I,4,TOTAL;
ITEM15 P,4,TOTAL;
ITEM16 LAMBDA,4,TOTAL;
ITEM17 U,SIGMA,TOTAL;
ITEM18 I,SIGMA,TOTAL;
ITEM19 P,SIGMA,TOTAL;
ITEM20 LAMBDA,SIGMA,TOTAL;
ITEM21 U,SIGMB,TOTAL;
ITEM22 I,SIGMB,TOTAL;
ITEM23 P,SIGMB,TOTAL;
ITEM24 LAMBDA,SIGMB,TOTAL;
ITEM25 WH,1;ITEM26 WH,2;
ITEM27 WH,3;ITEM28 WH,4;
ITEM29 WH,SIGMA;ITEM30 WH,SIGMB;
ITEM31 F1;ITEM32 F2;ITEM33 ETA1;
ITEM34 ETA2;ITEM35 ETA3;
ITEM36 ETA4;CURSOR 1
```

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

Function Sets the cursor position on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|
VAL8|VAL16}:CURSor {<NRf>}
:DISPlay:NUMeric[:NORMal]:{VAL4|
VAL8|VAL16}:CURSor?
```
<NRf> = 1 to 36 (item number, for VAL4)
<NRf> = 1 to 72 (item number, for VAL8)
<NRf> = 1 to 144 (item number, for VAL16)

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

Description Specify the cursor position in terms of the item number.

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

Function Sets the displayed items (function, element, and harmonic order) on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric:[:NORMal]:{VAL4|
VAL8|VAL16}:ITEM<x> {NONE|
<Function>,<Element>[,<Order>]}
:DISPlay:NUMeric:[:NORMal]:{VAL4|
VAL8|VAL16}:ITEM<x>?
```
<x> = 1 to 36 (item number, for VAL4)
<x> = 1 to 72 (item number, for VAL8)
<x> = 1 to 144 (item number, for VAL16)
NONE = No display item
<Function> = {U|I|P|S|Q|...} (See the function selection list (1) on page 6-44.)
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)
<Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 100)

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 U,1,TOTAL
:DISPLAY:NUMERIC:NORMAL:VAL4:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 U,1,TOTAL
```

Description
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTal is set.
- <Element> or <Order> is omitted from response to functions that do not need them.

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

Function Sets the page number on the numeric display ({4-value|8-value|16-value} display) or queries the current setting.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|
VAL8|VAL16}:PAGE {<NRf>}
:DISPlay:NUMeric[:NORMal]:{VAL4|
VAL8|VAL16}:PAGE?
```
<NRf> = 1 to 9 (page number)

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

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

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

Function Sets the displayed items on the numeric display ({4-value|8-value|16-value} display) to a preset pattern.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|
VAL8|VAL16}:PRESet {<NRf>}
```
<NRf> = 1 to 4

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

Description Regardless of what value (1 to 4) is specified for <NRf>, the display pattern (order) of the numeric display items will be the same as the display order when Reset Items Exec of the ITEM setting menu, which is displayed on the WT3000 screen, is executed. For details on the order of displayed items when reset is executed, see the *User's Manual IM760301-01E*.

:DISPlay:TRENd?

Function Queries all settings related to the trend.

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

Example
```
:DISPLAY:TREND? -> :DISPLAY:TREND:
FORMAT SINGLE;T1 1;T2 1;T3 1;T4 1;
T5 1;T6 1;T7 1;T8 1;T9 0;T10 0;
T11 0;T12 0;T13 0;T14 0;T15 0;
T16 0;TDIV 0,0,3;ITEM1:
FUNCTION U,1,TOTAL;SCALING:
MODE AUTO;
VALUE 100.0E+00,-100.0E+00;:
DISPLAY:TREND:ITEM2:
FUNCTION I,1,TOTAL;SCALING:
MODE AUTO;
VALUE 100.0E+00,-100.0E+00;...
(omitted)...;:DISPLAY:TREND:NORMAL:
ITEM16:FUNCTION AH,1;SCALING:
MODE AUTO;
VALUE 100.0E+00,-100.0E+00
```

:DISPlay:TRENd:ALL

Function Collectively turns ON/OFF all trends.

Syntax
```
:DISPlay:TRENd:ALL {<Boolean>}
```

Example
```
:DISPLAY:TREND:ALL ON
```

:DISPlay:TRENd:CLEar

Function Clears the trend.

Syntax
```
:DISPlay:TRENd:CLEar
```

Example
```
:DISPLAY:TREND:CLEAR
```

:DISPlay:TRENd:FORMat

Function Sets the display format of the trend or queries the current setting.

Syntax
```
:DISPlay:TRENd:FORMat {SINGle|DUAL|
TRIad|QUAD}
:DISPlay:TRENd:FORMat?
```

Example
```
:DISPLAY:TREND:FORMAT SINGLE
:DISPLAY:TREND:FORMAT? ->
:DISPLAY:TREND:FORMAT SINGLE
```

:DISPlay:TRENd:ITEM<x>?

Function Queries all settings related to the trend.

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

Example
```
:DISPLAY:TREND:ITEM1? -> :DISPLAY:
TREND:ITEM1:FUNCTION U,1,TOTAL;
SCALING:MODE AUTO;
VALUE 100.0E+00,-100.0E+00
```

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

Function Sets the trend item (function, element, and harmonic order) or queries the current setting.

Syntax
```
:DISPlay:TRENd:ITEM<x>
[:FUNCtion] {NONE|<Function>,
<Element>[,<Order>]}
:DISPlay:TRENd:ITEM<x>:FUNCtion?
```
<x> = 1 to 16 (item number)
NONE = No display item
<Function> = {U|I|P|S|Q|...} (See the function selection list (1) on page 6-44.)
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)
<Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 100)

Example
```
:DISPLAY:TREND:ITEM1:
FUNCTION U,1,TOTAL
:DISPLAY:TREND:ITEM1:FUNCTION? ->
:DISPLAY:TREND:ITEM1:
FUNCTION U,1,TOTAL
```

Description
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTal is set.
- <Element> or <Order> is omitted from response to functions that do not need them.

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

Function Queries all settings related to the scaling of the trend.

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

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

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

Function Sets the scaling mode of the trend or queries the current setting.

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

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

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

Function Sets the upper and lower limits of manual scaling of the trend or queries the current setting.

Syntax
```
:DISPlay:TRENd:ITEM<x>:SCALing:
VALue {<NRf>,<NRf>}
:DISPlay:TRENd:ITEM<x>:SCALing:
VALue?
```
<x> = 1 to 16 (item number)
<NRf> = –9.999E+30 to 9.999E+30

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

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

:DISPlay:TRENd:TDIV

Function Sets the horizontal axis (T/div) of the trend or queries the current setting.

Syntax
```
:DISPlay:TRENd:TDIV {<NRf>,<NRf>,
<NRf>}
:DISPlay:TRENd:TDIV?
```
{<NRf>, <NRf>, <NRf>} = 0, 0, 3 to 24, 0, 0
1st <NRf> = 1, 3, 6, 12, 24 (hour)
2nd <NRf> = 1, 3, 6, 10, 30 (minute)
3rd <NRf> = 3, 6, 10, or 30 (second)

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

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

:DISPlay:TRENd:T<x>

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

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

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

:DISPlay:VECTor?

Function Queries all settings related to the vector display.

Syntax
```
:DISPlay:VECTor?
```

Example
```
:DISPLAY:VECTOR? -> :DISPLAY:
VECTOR:OBJECT SIGMA;NUMERIC 1;
UMAG 1.000;IMAG 1.000
```

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:VECTor:NUMeric

Function Turns ON/OFF the numeric data display for the vector display or queries the current setting.

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

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:VECTor:OBJect

Function Sets the wiring unit to be displayed during vector display or queries the current setting.

Syntax
```
:DISPlay:VECTor:OBJect {SIGMA|
SIGMB}
:DISPlay:VECTor:OBJect?
```

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

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- SIGMB is selectable only on the 4-element model.

:DISPlay:VECTor:{UMAG|IMAG}

Function Sets the zoom factor of the {voltage|current} display during vector display or queries the current setting.

Syntax `:DISPlay:VECTor:{UMAG|IMAG} {<NRf>}`
`:DISPlay:VECTor:{UMAG|IMAG}?`
<NRf> = 0.100 to 100.000

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

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:DISPlay:WAVE?

Function Queries all settings related to the waveform display.

Syntax `:DISPlay:WAVE?`

Example
```
:DISPLAY:WAVE? -> :DISPLAY:WAVE:
FORMAT SINGLE;U1 1;U2 1;U3 1;U4 1;
I1 1;I2 1;I3 1;I4 1;TDIV 5.0E-03;
TRIGGER:MODE AUTO;SOURCE U1;
SLOPE RISE;LEVEL 0.0;:DISPLAY:WAVE:
INTERPOLATE LINE;GRATICULE GRID;
SVALUE 1;TLABEL 0;MAPPING:
MODE AUTO;:DISPLAY:WAVE:VZOOM:
U1 1.00;U2 1.00;U3 1.00;U4 1.00;
I1 1.00;I2 1.00;I3 1.00;I4 1.00;:
DISPLAY:WAVE:POSITION:U1 0.000;
U2 0.000;U3 0.000;U4 0.000;
I1 0.000;I2 0.000;I3 0.000;I4 0.000
```

:DISPlay:WAVE:ALL

Function Collectively turns ON/OFF all waveform displays.

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

Example `:DISPLAY:WAVE:ALL ON`

:DISPlay:WAVE:FORMat

Function Sets the display format of the waveform or queries the current setting.

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

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

:DISPlay:WAVE:GRATicule

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

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

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

:DISPlay:WAVE:INTerpolate

Function Sets the interpolation method of the waveform or queries the current setting.

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

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

:DISPlay:WAVE:MAPPing?

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

Syntax `:DISPlay:WAVE:MAPPing?`

Example
```
:DISPLAY:WAVE:MAPPING? ->
:DISPLAY:WAVE:MAPPING:MODE USER;
U1 0;U2 1;U3 2;U4 3;I1 0;I2 1;I3 2;
I4 3
```

:DISPlay:WAVE:MAPPing[:MODE]

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

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

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

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

Function Sets the mapping of the {voltage|current|rotating speed|torque|waveform computation} waveform to the split screen or queries the current setting.

Syntax
```
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|
SPEed|TORQue|MATH<x>} {<NRf>}
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|
SPEed|TORQue|MATH<x>}?
```
<x> of U<x>, I<x> = 1 to 4 (element)
<x> of MATH<x> = 1 to 2 (MATH)
<NRf> = 0 to 3

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

Description
- This command is valid when the waveform mapping method (:DISPlay:WAVE:MAPPing[:MODE]) is set to “USER.”
- {SPEed|TORQue} are valid only on the motor version (-MV).
- MATH<x> is valid only on models with the advanced computation function (/G6 option).

:DISPlay:WAVE:POSition?

Function Queries all settings related to the vertical position (level of the center position) of the waveform.

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

Example
```
:DISPLAY:WAVE:POSITION? ->
:DISPLAY:WAVE:POSITION:U1 0.000;
U2 0.000;U3 0.000;U4 0.000;
I1 0.000;I2 0.000;I3 0.000;I4 0.000
```

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

Function Collectively sets the vertical position (level of the center position) of the waveform {voltage|current} of all elements.

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

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

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

Function Sets the vertical position (level of the center position) of the waveform {voltage|current} of the element or queries the current setting.

Syntax
```
:DISPlay:WAVE:POSition:{U<x>|
I<x>} {<NRf>}
:DISPlay:WAVE:POSition:{U<x>|I<x>}?
```
<x> = 1 to 4 (element)
<NRf> = –130.000 to 130.000(%)

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

:DISPlay:WAVE:SVALue (Scale VALue)

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

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

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

:DISPlay:WAVE:TDIV

Function Sets the Time/div value of the waveform or queries the current setting.

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

Example
```
:DISPLAY:WAVE:TDIV 5MS
:DISPLAY:WAVE:TDIV? ->
:DISPLAY:WAVE:TDIV 5.0E-03
```

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

:DISPlay:WAVE:TLABel (Trace LABel)

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

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

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

:DISPlay:WAVE:TRIGger?

Function Queries all settings related to the trigger.

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

Example
```
:DISPLAY:WAVE:TRIGGER? ->
:DISPLAY:WAVE:TRIGGER:MODE AUTO;
SOURCE U1;SLOPE RISE;LEVEL 0.0
```

:DISPlay:WAVE:TRIGger:LEVel

Function Sets the trigger level or queries the current setting.

Syntax
```
:DISPlay:WAVE:TRIGger:LEVel {<NRf>}
:DISPlay:WAVE:TRIGger:LEVel?
```
<NRf> = –100.0 to 100.0 (%) (The resolution is 0.1(%))

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

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

:DISPlay:WAVE:TRIGger:MODE

Function Sets the trigger mode or queries the current setting.

```
Syntax   :DISPlay:WAVE:TRIGger:MODE {AUTO|
         NORMal}
         :DISPlay:WAVE:TRIGger:MODE?
Example  :DISPLAY:WAVE:TRIGGER:MODE AUTO
         :DISPLAY:WAVE:TRIGGER:MODE? ->
         :DISPLAY:WAVE:TRIGGER:MODE AUTO
```

:DISPlay:WAVE:TRIGger:SLOPe

Function Sets the trigger slope or queries the current setting.

```
Syntax   :DISPlay:WAVE:TRIGger:SLOPe {RISE|
         FALL|BOTH}
         :DISPlay:WAVE:TRIGger:SLOPe?
Example  :DISPLAY:WAVE:TRIGGER:SLOPE RISE
         :DISPLAY:WAVE:TRIGGER:SLOPE? ->
         :DISPLAY:WAVE:TRIGGER:SLOPE RISE
```

:DISPlay:WAVE:TRIGger:SOURce

Function Sets the trigger source or queries the current setting.

```
Syntax   :DISPlay:WAVE:TRIGger:SOURce {U<x>|
         I<x>|EXTernal}
         :DISPlay:WAVE:TRIGger:SOURce?
```

<x> = 1 to 4 (element)

EXTernal = External trigger input (Ext Clk)

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

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

Function Turns ON/OFF the {voltage|current|rotating speed|torque|waveform computation} waveform or queries the current setting.

```
Syntax   :DISPlay:WAVE:{U<x>|I<x>|SPEed|
         TORQue|MATH<x>} {<Boolean>}
         :DISPlay:WAVE:{U<x>|I<x>|SPEed|
         TORQue|MATH<x>}?
```

<x> of U<x>, I<x> = 1 to 4 (element)

<x> of MATH<x> = 1 to 2 (MATH)

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

Description
- {SPEed|TORQue} are valid only on the motor version (-MV).
- MATH<x> is valid only on models with the advanced computation function (/G6 option).

:DISPlay:WAVE:VZoom?

Function Queries all settings related to the vertical zoom factor of the waveform.

```
Syntax   :DISPlay:WAVE:VZoom?
Example  :DISPLAY:WAVE:VZOOM? ->
         :DISPLAY:WAVE:VZOOM:U1 1.00;
         U2 1.00;U3 1.00;U4 1.00;I1 1.00;
         I2 1.00;I3 1.00;I4 1.00
```

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

Function Collectively sets the vertical zoom factor of the waveform {voltage|current} of all elements.

```
Syntax   :DISPlay:WAVE:VZoom:{UALL|IALL}
         {<NRf>}
```

<NRf> = 0.1 to 100 (see the *User's Manual IM760301-01E*)

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

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

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

```
Syntax   :DISPlay:WAVE:VZoom:{U<x>|I<x>}
         {<NRf>}
         :DISPlay:WAVE:VZoom:{U<x>|I<x>}?
```

<x> = 1 to 4 (element)

<NRf> = 0.1 to 100 (see the *User's Manual IM760301-01E*)

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

*Function Selection (<Function>) List

(1) Function of numeric data

Applicable commands

```
:AOUTput[:NORMal]:CHANnel<x>
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>
:DISPlay:TRENd:ITEM<x>[:FUNCtion]
:NUMeric[:NORMal]:ITEM<x>
```

Function name used in commands	**Function name used on the menu (Numeric display header name)**	**Elements**	**Order**
U	U (Urms/Umn/Udc/Urmn)	Required	Required
I	I (Irms/Imn/Idc/Irmn)	Required	Required
P	P	Required	Required
S	S	Required	Required
Q	Q	Required	Required
LAMBda	λ	Required	Required
PHI	φ	Required	Required
FU	FreqU (fU)	Required	Not required
FI	FreqI (fI)	Required	Not required
UPPeak	U+peak (U+pk)	Required	Not required
UMPeak	U-peak (U-pk)	Required	Not required
IPPeak	I+peak (I+pk)	Required	Not required
IMPeak	I-peak (I-pk)	Required	Not required
CFU	CfU	Required	Not required
CFI	CfI	Required	Not required
PC	Pc	Required	Not required
TIME	Time	Required	Not required
WH	WP	Required	Not required
WHP	WP+	Required	Not required
WHM	WP-	Required	Not required
AH	q	Required	Not required
AHP	q+	Required	Not required
AHM	q-	Required	Not required
WS	WS	Required	Not required
WQ	WQ	Required	Not required
ETA1	η1	Not required	Not required
ETA2	η2	Not required	Not required
ETA3	η3	Not required	Not required
ETA4	η4	Not required	Not required
F1	F1	Not required	Not required
F2	F2	Not required	Not required
F3	F3	Not required	Not required
F4	F4	Not required	Not required
F5	F5	Not required	Not required
F6	F6	Not required	Not required
F7	F7	Not required	Not required
F8	F8	Not required	Not required
F9	F9	Not required	Not required
F10	F10	Not required	Not required
F11	F11	Not required	Not required
F12	F12	Not required	Not required
F13	F13	Not required	Not required
F14	F14	Not required	Not required
F15	F15	Not required	Not required
F16	F16	Not required	Not required
F17	F17	Not required	Not required
F18	F18	Not required	Not required
F19	F19	Not required	Not required
F20	F20	Not required	Not required

Functions that require the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option)			
PHIU	ϕU	Required	Required
PHII	ϕI	Required	Required
Z	Z	Required	Required
RS	Rs	Required	Required
XS	Xs	Required	Required
RP	Rp	Required	Required
XP	Xp	Required	Required
UHDF	Uhdf	Required	Required
IHDF	Ihdf	Required	Required
PHDF	Phdf	Required	Required
UTHD	Uthd	Required	Not required
ITHD	Ithd	Required	Not required
PTHD	Pthd	Required	Not required
UTHF	Uthf	Required	Not required
ITHF	Ithf	Required	Not required
UTIF	Utif	Required	Not required
ITIF	Itif	Required	Not required
HVF	hvf	Required	Not required
HCF	hcf	Required	Not required
PHI_U1U2	ϕUi-Uj	Required	Not required
PHI_U1U3	ϕUi-Uk	Required	Not required
PHI_U1I1	ϕUi-Ii	Required	Not required
PHI_U1I2	ϕUi-Ij	Required	Not required
PHI_U1I3	ϕUi-Ik	Required	Not required
Functions that require the delta computation function (/DT option)			
DELTA1	ΔF1	Not required	Not required
DELTA2	ΔF2	Not required	Not required
DELTA3	ΔF3	Not required	Not required
DELTA4	ΔF4	Not required	Not required
Functions that require the motor version (-MV)			
SPEed	Speed	Not required	Not required
TORQue	Torque	Not required	Not required
SYNCsp	SyncSp	Not required	Not required
SLIP	Slip	Not required	Not required
PM	Pm	Not required	Not required

In addition, the function listed below can be used for the following command.

```
:DISPlay:NUMeric[:NORMal]:ALL:CURSor
:FILE:SAVE:NUMeric[:NORMal]:<Function>
:HCOPy:PRINter:LIST[:NORMal]:<Function>
:STORe:NUMeric[:NORMal]:<Function>
```

Functions that require the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option)

UK	U(k)
IK	I(k)
PK	P(k)
SK	S(k)
QK	Q(k)
LAMBDAK	λ(k)
PHIK	ϕ(k)
PHIUk	ϕU(k)
PHIIk	ϕI(k)
Zk	Z(k)
RSk	Rs(k)
XSk	Xs(k)
RPk	Rp(k)
XPk	Xp(k)

Note

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

(2) Functions of the numeric list data (The advanced computation function (/G6 option) or the harmonic measurement function (/G5 option) is required.)

Applicable commands

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

Function name used in commands	Function name used on the menu (Numeric display header name)
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

In addition, the function listed below can be used for the following command.

```
:NUMeric:LIST:ITEM<x>
```

UHDF	Uhdf
IHDF	Ihdf
PHDF	Phdf

6.8 FILE Group

The commands in this group deal with file operations.
You can make the same settings and inquiries as when FILE on the front panel is used.

:FILE?

Function Queries all settings related to the file operation.
Syntax `:FILE?`
Example `:FILE?` -> (same as the response to "`:FILE:SAVE?`")

:FILE:CDIRectory

Function Changes the current directory.
Syntax `:FILE:CDIRectory {<Filename>}`
<Filename> = Directory name
Example `:FILE:CDIRECTORY "IMAGE"`
Description Specify "`..`" to move up to the parent directory.

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

Function Deletes the screen image data file.
Syntax `:FILE:DELete:IMAGe:{TIFF|BMP|PSCRipt|PNG|JPEG} {<Filename>}`
Example `:FILE:DELETE:IMAGE:TIFF "IMAG1"`
Description Specify the file name without the extension.

:FILE:DELete:NUMeric:{ASCii|FLOat}

Function Deletes the numeric data file.
Syntax `:FILE:DELete:NUMeric:{ASCii|FLOat} {<Filename>}`
Example `:FILE:DELETE:NUMERIC:ASCII "NUM1"`
Description Specify the file name without the extension.

:FILE:DELete:SETup

Function Deletes the setup parameter file.
Syntax `:FILE:DELete:SETup {<Filename>}`
Example `:FILE:DELETE:SETUP "SETUP1"`
Description Specify the file name without the extension.

:FILE:DELete:WAVE:{BINary|ASCii|FLOat}

Function Deletes the waveform display data file.
Syntax `:FILE:DELete:WAVE:{BINary|ASCii|FLOat} {<Filename>}`
Example `:FILE:DELETE:WAVE:BINARY "WAVE1"`
Description Specify the file name without the extension.

:FILE:DRIVe

Function Sets the target drive.
Syntax `:FILE:DRIVe {PCCard[,<NRf>]|NETWork|USB,<NRf>[,<NRf>][,<NRf>]}`
PCCard = PC card drive
<NRf> = Partition (0 to 3)
NETWork = Network drive
USB = USB memory drive
1st <NRf> = ID number (address)
2nd <NRf> = Partition (0 to 3) or LUN (logical unit number: 0 to 3)
3rd <NRf> = Partition (0 to 3) when LUN is specified
Example `:FILE:DRIVE PCCARD`
Description
- If the drive does not contain partitions, omit the <NRf> corresponding to partitions.
- "NETWork" can be used when the Ethernet interface (/C7 option) is installed.
- "USB" can be used when the USB port (peripheral device) (/C5 option) is installed.
- The second or third <NRf> when USB is selected can be omitted if the drive is not partitioned or divided by LUN.

:FILE:FORMat:EXECute

Function Formats the PC card.
Syntax `:FILE:FORMat:EXECute`
Example `:FILE:FORMAT:EXECUTE`

:FILE:FREE?

Function Queries the free disk space (bytes) on the drive.
Syntax `:FILE:FREE?`
Example `:FILE:FREE? -> 163840`

:FILE:LOAD:ABORt

Function Aborts file loading.
Syntax `:FILE:LOAD:ABORt`
Example `:FILE:LOAD:ABORT`

:FILE:LOAD:SETup

Function Loads the setup parameter file.
Syntax `:FILE:LOAD:SETup {<Filename>}`
Example `:FILE:LOAD:SETUP "SETUP1"`
Description
- Specify the file name without the extension.
- This command is an overlap command.

:FILE:MDIRectory

Function Creates a directory.

Syntax `:FILE:MDIRectory {<Filename>}`
<Filename> = Directory name

Example `:FILE:MDIRECTORY "TEST"`

:FILE:PATH?

Function Queries the absolute path of the current directory.

Syntax `:FILE:PATH?`

Example `:FILE:PATH? -> "PC_Card\IMAGE"`

:FILE:SAVE?

Function Queries all settings related to the saving of files.

Syntax `:FILE:SAVE?`

Example
```
:FILE:SAVE? -> :FILE:SAVE:
ANAMING 1;COMMENT "";WAVE:
TYPE BINARY;:FILE:SAVE:NUMERIC:
TYPE ASCII;NORMAL:ELEMENT1 1;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;
SIGMA 0;SIGMB 0;U 1;I 1;P 1;S 1;
Q 1;LAMBDA 1;PHI 1;FU 1;FI 1;
UPPEAK 0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0
```

:FILE:SAVE:ABORt

Function Aborts file saving.

Syntax `:FILE:SAVE:ABORt`

Example `:FILE:SAVE:ABORT`

:FILE:SAVE:ACQuisition?

Function Queries all settings related to the file saving of the waveform sampling data.

Syntax `:FILE:SAVE:ACQuisition?`

Example
```
:FILE:SAVE:ACQUISITION? ->
:FILE:SAVE:ACQUISITION:TYPE FLOAT;
TRACE U1
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:FILE:SAVE:ACQuisition[:EXECute]

Function Saves the waveform sampling data to a file.

Syntax `:FILE:SAVE:ACQuisition [:EXECute] {<Filename>}`

Example `:FILE:SAVE:ACQUISITION: EXECUTE "ACQ1"`

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- Specify the file name without the extension.
- This command is an overlap command.

:FILE:SAVE:ACQuisition:TRACe

Function Sets the waveform sampling data to be saved to a file or queries the current setting.

Syntax
```
:FILE:SAVE:ACQuisition:TRACe {U<x>|
I<x>|SPEed|TORQue|MATH<x>|FFT<x>}
:FILE:SAVE:ACQuisition:TRACe?
```
<x> of U<x>, I<x> = 1 to 4 (element)
<x> of MATH<x> = 1 to 2 (MATH)
<x> of FFT<x> = 1 or 2 (FFT)

Example
```
:FILE:SAVE:ACQUISITION:TRACE U1
:FILE:SAVE:ACQUISITION:TRACE? ->
:FILE:SAVE:ACQUISITION:TRACE U1
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- This command is valid when the format of the waveform sampling data to be saved (`:FILE:SAVE:ACQuisition:TYPE`) is “FLOat.” When it is `{BINary|ASCii}`, all waveform sampling data of which the waveform display is turned ON are saved.
- `{SPEed|TORQue}` are valid only on the motor version (-MV).

:FILE:SAVE:ACQuisition:TYPE

Function Sets the format of the waveform sampling data to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:ACQuisition:
TYPE {BINary|ASCii|FLOat}
:FILE:SAVE:ACQuisition:TYPE?
```

Example
```
:FILE:SAVE:ACQUISITION:TYPE FLOAT
:FILE:SAVE:ACQUISITION:TYPE? ->
:FILE:SAVE:ACQUISITION:TYPE FLOAT
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:FILE:SAVE:ANAMing

Function Sets whether to automatically name the files to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:ANAMing {<Boolean>}
:FILE:SAVE:ANAMing?
```

Example
```
:FILE:SAVE:ANAMING ON
:FILE:SAVE:ANAMING? ->
:FILE:SAVE:ANAMING 1
```

:FILE:SAVE:COMMent

Function Sets the comment to be added to the file to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:COMMent {<String>}
:FILE:SAVE:COMMent?
```
<String> = Up to 25 characters

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

:FILE:SAVE:NUMeric?

Function Queries all settings related to the saving of numeric data files.

Syntax `:FILE:SAVE:NUMeric?`

Example
```
:FILE:SAVE:NUMERIC? ->
:FILE:SAVE:NUMERIC:TYPE ASCII;
NORMAL:ELEMENT1 1;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;SIGMA 0;
SIGMB 0;U 1;I 1;P 1;S 1;Q 1;
LAMBDA 1;PHI 1;FU 1;FI 1;UPPEAK 0;
UMPEAK 0;IPPEAK 0;IMPEAK 0;CFU 0;
CFI 0;PC 0;TIME 0;WH 0;WHP 0;WHM 0;
AH 0;AHP 0;AHM 0;WS 0;WQ 0;ETA1 0;
ETA2 0;ETA3 0;ETA4 0;F1 0;F2 0;
F3 0;F4 0;F5 0;F6 0;F7 0;F8 0;F9 0;
F10 0;F11 0;F12 0;F13 0;F14 0;
F15 0;F16 0;F17 0;F18 0;F19 0;F20 0
```

:FILE:SAVE:NUMeric:CBCycle?

Function Queries all settings related to Cycle by Cycle measurement items saved to numeric data files or queries the current setting

Syntax `:FILE:SAVE:NUMeric:CBCycle?`

Example
```
:FILE:SAVE:NUMERIC:CBCYCLE? ->
:FILE:SAVE:NUMERIC:CBCYCLE
:ELEMENT1 1;ELEMENT2 0;ELEMENT3 0;
ELEMENT4 0;SIGMA 0;SIGMB 0;FREQ 1;
U 1;I 1;P 1;S 1;Q 1;LAMBDA 1
```

Description Only available with the Cycle by Cycle measurement function (/CC option).

:FILE:SAVE:NUMeric:CBCycle:ALL

Function Collectively turns ON/OFF the output of all elements and functions when saving numeric data from Cycle by Cycle measurement to a file.

Syntax
```
:FILE:SAVE:NUMeric:CBCycle:
ALL {<Boolean>}
```

Example `:FILE:SAVE:NUMERIC:CBCYCLE:ALL ON`

Description Only available with the Cycle by Cycle measurement function (/CC option).

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

Function Turns ON/OFF the output of {each element | ΣA | ΣB} when saving numeric data from Cycle by Cycle measurement to a file.

Syntax
```
:FILE:SAVE:NUMeric:
CBCycle:{ELEMent<x>|SIGMA|SIGMB}
{<Boolean>}
:FILE:SAVE:NUMeric:
CBCycle:{ELEMent<x>|SIGMA|SIGMB}?
```
<x> = 1 to 4

Example
```
:FILE:SAVE:NUMERIC:CBCYCLE:
ELEMENT1 ON
:FILE:SAVE:NUMERIC:CBCYCLE:
ELEMENT1? -> :FILE:SAVE:NUMERIC:
CBCYCLE:ELEMENT1 1
```

Description
- Only available with the Cycle by Cycle measurement function (/CC option).
- ":FILE:SAVE:NUMeric:CBCycle:SIGMA" is available for models with 2 elements or more. Also, to turn output ON, wiring unit ΣA must exist per the wiring system setting command ([:INPut]WIRing).
- ":FILE:SAVE:NUMeric:CBCycle:SIGMB" is valid for models with 4 elements. Also, to turn output ON, wiring unit ΣB must exist per the wiring system setting command ([:INPut]WIRing).

:FILE:SAVE:NUMeric:CBCycle:<Function>

Function Turns ON/OFF the output of each function when saving numeric data from Cycle by Cycle measurement to a file or queries the current setting.

Syntax
```
:FILE:SAVE:NUMeric:
CBCycle:<Function> {<Boolean>}
:FILE:SAVE:NUMeric:
CBCycle:<Function>?
```
<Function> = {FREQ|U|I|P|S|Q|LAMBda|SPEed|TORQue|PM}

Example
```
:FILE:SAVE:NUMERIC:CBCYCLE:U ON
:FILE:SAVE:NUMERIC:CBCYCLE:U? ->
:FILE:SAVE:NUMERIC:CBCYCLE:U 1
```

Description
- Only available with the Cycle by Cycle function (/CC option).
- {SPEed|TORQue|PM} is only available with the motor version (-MV).

:FILE:SAVE:NUMeric[:EXECute]

Function Saves the numeric data to a file.

Syntax
```
:FILE:SAVE:NUMeric[:EXECute]
{<Filename>}
```

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

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

:FILE:SAVE:NUMeric:NORMal?

Function Queries all settings related to the items saved to numeric data files.

Syntax `:FILE:SAVE:NUMeric:NORMal?`

Example
```
:FILE:SAVE:NUMERIC:NORMAL? ->
:FILE:SAVE:NUMERIC:NORMAL:
ELEMENT1 1;ELEMENT2 0;ELEMENT3 0;
ELEMENT4 0;SIGMA 0;SIGMB 0;U 1;I 1;
P 1;S 1;Q 1;LAMBDA 1;PHI 1;FU 1;
FI 1;UPPEAK 0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0
```

:FILE:SAVE:NUMeric[:NORMal]:ALL

Function Collectively turns ON/OFF the output of all element functions when saving the numerical data file.

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

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

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

Function Turns ON/OFF the output of {each element|ΣA|ΣB} when saving the numeric data to 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
```

Description
- `:FILE:SAVE:NUMeric[:NORMal]:SIGMA` is valid on models with two or more elements. To turn the output ON, wiring unit ΣA must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.
- `:FILE:SAVE:NUMeric[:NORMal]:SIGMB` is valid on models with four elements. To turn the output ON, wiring unit ΣB must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.

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

Function Presets the output ON/OFF pattern of the element function for saving the numeric data to file.

Syntax
```
:FILE:SAVE:NUMeric[:NORMal]:
PRESet<x>
```
<x> = 1 to 2 (preset pattern number)

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

Description For details on the output pattern when preset is executed, see the *User's Manual IM760301-01E*.

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

Function Turns ON/OFF the output of the function when saving the numerical data file or queries the current setting.

Syntax
```
:FILE:SAVE:NUMeric[:NORMal]:
<Function> {<Boolean>}
:FILE:SAVE:NUMeric[:NORMal]:
<Function>?
```
<Function> = `{U|I|P|S|Q|...}`(See the function selection list (1) of “DISPlay group” on page 6-44.)

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

:FILE:SAVE:NUMeric:TYPE

Function Sets the format of the numeric data to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:NUMeric:TYPE {ASCii|
FLOat}
:FILE:SAVE:NUMeric:TYPE?
```

Example
```
:FILE:SAVE:NUMERIC:TYPE ASCII
:FILE:SAVE:NUMERIC:TYPE? ->
:FILE:SAVE:NUMERIC:TYPE ASCII
```

:FILE:SAVE:SETup[:EXECute]

Function Saves of the setup parameter file.

Syntax
```
:FILE:SAVE:SETup[:EXECute]
{<Filename>}
```

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

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

:FILE:SAVE:WAVE?

Function Queries all settings related to the saving of waveform display data files.

Syntax `:FILE:SAVE:WAVE?`

Example
```
:FILE:SAVE:WAVE? ->
:FILE:SAVE:WAVE:TYPE BINARY
```

:FILE:SAVE:WAVE[:EXECute]

Function Executes the saving of the waveform display data file.

Syntax `:FILE:SAVE:WAVE[:EXECute] {<Filename>}`

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

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

:FILE:SAVE:WAVE:TRACe

Function Sets the waveform to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:WAVE:TRACe {U<x>|I<x>|
SPEed|TORQue|MATH<x>}
:FILE:SAVE:WAVE:TRACe?
```
<x> of U<x>, I<x> = 1 to 4 (element)
<x> of MATH<x> = 1 to 2 (MATH)

Example
```
:FILE:SAVE:WAVE:TRACE U1
:FILE:SAVE:WAVE:TRACE? ->
:FILE:SAVE:WAVE:TRACE U1
```

Description
- This command is valid when the format of the waveform display data to be saved (`:FILE:SAVE:WAVE:TYPE`) is “FLOat.” When it is `{BINary|ASCii}`, all waveforms of which the display is turned ON are saved.
- `{SPEed|TORQue}` are valid only on the motor version (-MV).
- MATH<x> is valid only on models with the advanced computation function (/G6 option).

:FILE:SAVE:WAVE:TYPE

Function Sets the format of the waveform display data to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:WAVE:TYPE {BINary|ASCii|
FLOat}
:FILE:SAVE:WAVE:TYPE?
```

Example
```
:FILE:SAVE:WAVE:TYPE BINARY
:FILE:SAVE:WAVE:TYPE? ->
:FILE:SAVE:WAVE:TYPE BINARY
```

6.9 FLICker Group

The FLICker group contains commands related to flicker measurement.
These commands allow you to enter and query the same settings that are available under ITEM in the Flicker Items menu and under FORM in the Flicker Form menu on the front panel.
Note that the commands in this group are only available with the flicker measurement function (/FL option).

:FLICker?

Function Queries all settings related to flicker measurement.

Syntax
```
:FLICker?
```

Example
```
:FLICKER? -> :FLICKER:
MEASUREMENT FLICKER;ELEMENT1 1;
ELEMENT2 0;ELEMENT3 0;
INTERVAL 10,0;COUNT 12;
FREQUENCY 50;UN:MODE AUTO;
VALUE 230.00;:FLICKER:DC:STATE 1;
LIMIT 3.30;:FLICKER:DMAX:STATE 1;
LIMIT 4.00;:FLICKER:DT:STATE 1;
LIMIT 500,3.30;:FLICKER:PST:
STATE 1;LIMIT 1.00;:FLICKER:PLT:
STATE 1;LIMIT 0.65;NVALUE 12;:
FLICKER:DMIN:LIMIT 0.10
```

:FLICker:COUNt

Function Sets the number of measurements for the short-term flicker value Pst or queries the current setting.

Syntax
```
:FLICker:COUNt {<NRf>}
:FLICker:COUNt?
```
<NRf> = 1 to 99 (no. of measurements)

Example
```
:FLICKER:COUNT 12
:FLICKER:COUNT? ->
:FLICKER:COUNT 12
```

Description This setting is available with normal flicker measurement (`:FLICker:MEASurement FLICker`).
For measurement of dmax caused by manual switching (`:FLICker:MEASurement DMAX`), the number of measurements is fixed at 24.

:FLICker:DC?

Function Sets all settings related to the relative steady-state voltage change dc or queries the current setting.

Syntax
```
:FLICker:DC?
```

Example
```
:FLICKER:DC? ->
:FLICKER:DC:STATE 1;LIMIT 3.30
```

:FLICker:DC:LIMit

Function Sets the limit of the relative steady-state voltage change dc or queries the current setting.

Syntax
```
:FLICker:DC:LIMit {<NRf>}
:FLICker:DC:LIMit?
```
<NRf> = 1.00 to 99.99 (limit[%])

Example
```
:FLICKER:DC:LIMIT 3.30
:FLICKER:DC:LIMIT? ->
:FLICKER:DC:LIMIT 3.30
```

:FLICker:DC[:STATe]

Function Turns ON/OFF judgment of the relative steady-state voltage change dc or queries the current setting.

Syntax
```
:FLICker:DC[:STATe] {<Boolean>}
:FLICker:DC:STATe?
```

Example
```
:FLICKER:DC:STATE ON
:FLICKER:DC:STATE? ->
:FLICKER:DC:STATE 1
```

:FLICker:DISPlay?

Function Queries all settings related to flicker measurement display.

Syntax
```
:FLICker:DISPlay?
```

Example
```
:FLICKER:DISPLAY? ->
:FLICKER:DISPLAY:ELEMENT 1;PERIOD 1
```

:FLICker:DISPlay:ELEMent

Function Sets the element to be displayed for flicker measurement display or queries the current setting.

Syntax
```
:FLICker:DISPlay:ELEMent {<NRf>}
:FLICker:DISPlay:ELEMent?
```
<NRf> = 1 to 4 (element)

Example
```
:FLICKER:DISPLAY:ELEMENT 1
:FLICKER:DISPLAY:ELEMENT? ->
:FLICKER:DISPLAY:ELEMENT 1
```

Description You can make the same setting or query with the "`:DISPlay:FLICker:ELEMent`" command.

:FLICker:DISPlay:PAGE

Function Sets the page numbers to be displayed for flicker measurement display or queries the current setting.

Syntax `:FLICker:DISPlay:PAGE {<NRf>}`
`:FLICker:DISPlay:PAGE?`
<NRf> = 1 to 9 (page number)

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

Description You can make the same setting or query with the "`:DISPlay:FLICker:PAGE`" command.

:FLICker:DISPlay:PERiod

Function Sets the display observation period number for flicker measurement display or queries the current setting.

Syntax `:FLICker:DISPlay:PERiod {<NRf>}`
`:FLICker:DISPlay:PERiod?`
<NRf> = 1 to 99 (observation period number)

Example `:FLICKER:DISPLAY:PERIOD 1`
`:FLICKER:DISPLAY:PERIOD? ->`
`:FLICKER:DISPLAY:PERIOD 1`

Description You can make the same setting or query with the "`:DISPlay:FLICker:PERiod`" command.

:FLICker:DMAX?

Function Sets all settings related to the maximum relative voltage change dmax or queries the current setting.

Syntax `:FLICker:DMAX?`

Example `:FLICKER:DMAX? ->`
`:FLICKER:DMAX:STATE 1;LIMIT 4.00`

:FLICker:DMAX:LIMit

Function Sets the limit of the maximum relative voltage change dmax or queries the current setting.

Syntax `:FLICker:DMAX:LIMit {<NRf>}`
`:FLICker:DMAX:LIMit?`
<NRf> = 1.00 to 99.99 (limit[%])

Example `:FLICKER:DMAX:LIMIT 4.00`
`:FLICKER:DMAX:LIMIT? ->`
`:FLICKER:DMAX:LIMIT 4.00`

:FLICker:DMAX[:STATe]

Function Turns ON/OFF judgment of the maximum relative voltage change dmax or queries the current setting.

Syntax `:FLICker:DMAX[:STATe] {<Boolean>}`
`:FLICker:DMAX:STATe?`

Example `:FLICKER:DMAX:STATE ON`
`:FLICKER:DMAX:STATE? ->`
`:FLICKER:DMAX:STATE 1`

:FLICker:DMIN?

Function Sets all settings related to the steady-state range dmin or queries the current setting.

Syntax `:FLICker:DMIN?`

Example `:FLICKER:DMIN? ->`
`:FLICKER:DMIN:LIMIT 0.10`

:FLICker:DMIN:LIMit

Function Sets the limit of the steady-state range dmin or queries the current setting.

Syntax `:FLICker:DMIN:LIMit {<NRf>}`
`:FLICker:DMIN:LIMit?`
<NRf> = 0.10.00 to 9.99 (limit[%])

Example `:FLICKER:DMIN:LIMIT 0.10`
`:FLICKER:DMIN:LIMIT? ->`
`:FLICKER:DMIN:LIMIT 0.10`

:FLICker:DT?

Function Sets all settings related to the relative voltage change time d(t) or queries the current setting.

Syntax `:FLICker:DT?`

Example `:FLICKER:DT? ->`
`:FLICKER:DT:STATE 1;LIMIT 500,3.30`

:FLICker:DT:LIMit

Function Sets the limit of the relative voltage change time d(t) or queries the current setting.

Syntax `:FLICker:DT:LIMit {<NRf>[,<NRf>]}`
`:FLICker:DT:LIMit?`
1st <NRf> = 1.00 to 99999 (limit[ms])
2nd <NRf> = 1.00 to 99.99 (threshold level[%])

Example `:FLICKER:DT:LIMIT 500,3.30`
`:FLICKER:DT:LIMIT? ->`
`:FLICKER:DT:LIMIT 500,3.30`

Description If the second parameter (threshold level) is not to be set, it can be omitted.

:FLICker:DT[:STATe]

Function Turns ON/OFF judgment of the relative voltage change time d(t) or queries the current setting.

Syntax `:FLICker:DT[:STATe] {<Boolean>}`
`:FLICker:DT:STATe?`

Example `:FLICKER:DT:STATE ON`
`:FLICKER:DT:STATE? ->`
`:FLICKER:DT:STATE 1`

:FLICker:ELEMent<x>

Function Sets the target element flicker measurement or queries the current setting.

Syntax `:FLICker:ELEMent<x> {<Boolean>}`
`:FLICker:ELEMent<x>?`
<x> = 1 to 4 (element)

Example `:FLICKER:ELEMENT1 ON`
`:FLICKER:ELEMENT1? ->`
`:FLICKER:ELEMENT1 1`

Description When turned ON (1), that element is targeted for flicker measurement.

:FLICker:FREQuency

Function Sets the target frequency for flicker measurement or queries the current setting.

Syntax `:FLICker:FREQuency {<NRf>}`
`:FLICker:FREQuency?`
<NRf> = 50, 60 (target frequency [Hz])

Example `:FLICKER:FREQUENCY 50`
`:FLICKER:FREQUENCY? ->`
`:FLICKER:FREQUENCY 50`

:FLICker:INITialize

Function Initializes flicker measurement.

Syntax `:FLICker:INITialize`

Example `:FLICKER:INITIALIZE`

:FLICker:INTerval

Function Sets the time per measurement of the short-term flicker value Pst or queries the current setting.

Syntax `:FLICker:INTerval {<NRf>,<NRf>}`
`:FLICker:INTerval?`
<NRf>,<NRf> = 0,30 to 15,00 (measurement time: minutes, seconds)

Example `:FLICKER:INTERVAL 10,00`
`:FLICKER:INTERVAL? ->`
`:FLICKER:INTERVAL 10,00`

Description
- This setting is available with normal flicker measurement (`:FLICker:MEASurement FLICker`). The time per measurement of dmax caused by manual switching (`:FLICker:MEASurement DMAX`) is fixed at 1 (min) 00 (sec).
- The setting resolution for the measurement time is 2 seconds. When an odd number of seconds is set, it is rounded up to the next second.

:FLICker:JUDGe

Function Finishes measurement of dmax caused by manual switching and performs judgment.

Syntax `:FLICker:JUDGe`

Example `:FLICKER:JUDGE`

Description This command can be executed with measurement of dmax caused by manual switching (`:FLICker:MEASurement DMAX`). An error occurs if used during normal flicker measurement (`:FLICker:MEASurement FLICker`).

:FLICker:MEASurement

Function Sets the flicker measurement method or queries the current setting.

Syntax `:FLICker:MEASurement {FLICker|DMAX}`
`:FLICker:MEASurement?`
FLICker = Normal flicker measurement
DMAX = measurement of dmax caused by manual switching

Example `:FLICKER:MEASUREMENT FLICKER`
`:FLICKER:MEASUREMENT? ->`
`:FLICKER:MEASUREMENT FLICKER`

:FLICker:MOVe

Function Moves the observation period number for measurement of dmax caused by manual switching.

Syntax `:FLICker:MOVe {<NRf>}`
<NRf> = 1 to 24 (observation period number of destination)

Example `:FLICKER:MOVE 1`

Description
- The command re-executes measurement if dmax measurement of certain observation periods is not made correctly.
- This command can be executed with measurement of dmax caused by manual switching (`:FLICker:MEASurement DMAX`). An error occurs if used during normal flicker measurement (`:FLICker:MEASurement FLICker`).

:FLICker:PLT?

Function Queries all settings related to the long-term flicker value Plt.

Syntax `:FLICker:PLT?`

Example `:FLICKER:PLT? -> :FLICKER:PLT:`
`STATE 1;LIMIT 0.65;NVALUE 12`

:FLICker:PLT:LIMit

Function Sets the limit of the long-term flicker value Plt or queries the current setting.

Syntax `:FLICker:PLT:LIMit {<NRf>}`
`:FLICker:PLT:LIMit?`
<NRf> = 0.10 to 99.99 (limit)

Example `:FLICKER:PLT:LIMIT 0.65`
`:FLICKER:PLT:LIMIT? ->`
`:FLICKER:PLT:LIMIT 0.65`

:FLICker:PLT:NVALue

Function Sets constant N for the equation used to compute the long-term flicker value Plt or queries the current setting.

Syntax `:FLICker:PLT:NVALue {<NRf>}`
`:FLICker:PLT:NVALue?`
<NRf> = 1 to 99 (constant N)

Example `:FLICKER:PLT:NVALUE 12`
`:FLICKER:PLT:NVALUE? ->`
`:FLICKER:PLT:NVALUE 12`

:FLICker:PLT[:STATe]

Function Turns ON/OFF judgment of the long-term flicker value Plt or queries the current setting.

Syntax `:FLICker:PLT[:STATe] {<Boolean>}`
`:FLICker:PLT:STATe?`

Example `:FLICKER:PLT:STATE ON`
`:FLICKER:PLT:STATE? ->`
`:FLICKER:PLT:STATE 1`

:FLICker:PST?

Function Queries all settings related to the short-term flicker value Pst.

Syntax `:FLICker:PST?`

Example `:FLICKER:PST? ->`
`:FLICKER:PST:STATE 1;LIMIT 1.00`

:FLICker:PST:LIMit

Function Sets the limit for the short-term flicker value or queries the current setting.

Syntax `:FLICker:PST:LIMit {<NRf>}`
`:FLICker:PST:LIMit?`
<NRf> = 0.10 to 99.99 (limit)

Example `:FLICKER:PST:LIMIT 1.00`
`:FLICKER:PST:LIMIT? ->`
`:FLICKER:PST:LIMIT 1.00`

:FLICker:PST[:STATe]

Function Turns ON/OFF judgment of the short-term flicker value Pst or queries the current setting.

Syntax `:FLICker:PST[:STATe] {<Boolean>}`
`:FLICker:PST:STATe?`

Example `:FLICKER:PST:STATE ON`
`:FLICKER:PST:STATE? ->`
`:FLICKER:PST:STATE 1`

:FLICker:RESet

Function Resets measured flicker data.

Syntax `:FLICker:RESet`

Example `:FLICKER:RESET`

:FLICker:STARt

Function Starts flicker measurement.

Syntax `:FLICker:STARt`

Example `:FLICKER:START`

:FLICker:STATe?

Function Queries the status of flicker measurement.

Syntax `:FLICker:STATe?`

Example `:FLICKER:STATE? -> RESET`

Description The contents of the response are as follows:
RESet = Reset status
INITialize = Initializing
READy = Measurement start wait state
STARt = Measuring
COMPlete = Measurement stopped, judgment results displayed

:FLICker:UN?

Function Queries all settings related to rated voltage Un.

Syntax `:FLICker:UN?`

Example `:FLICKER:UN? -> :FLICKER:UN:`
`MODE AUTO;VALUE 230.00`

:FLICker:UN:MODE

Function Sets the assignment method for rated voltage Un or queries the current setting.

Syntax `:FLICker:UN:MODE {AUTO|SET}`
`:FLICker:UN:MODE?`
AUTO = Use the voltage value measured upon start of measurement.
SET = Use the predefined value (`:FLICker:UN:VALue`).

Example `:FLICKER:UN:MODE AUTO`
`:FLICKER:UN:MODE? ->`
`:FLICKER:UN:MODE AUTO`

:FLICker:UN:VALue

Function Sets the predefined value of rated voltage Un or queries the current setting.

Syntax
```
:FLICker:UN:VALue {<NRf>}
:FLICker:UN:VALue?
```
<NRf> = 0.01 to 999.99 (predefined value[V])

Example
```
:FLICKER:UN:VALUE 230.00
:FLICKER:UN:VALUE? ->
:FLICKER:UN:VALUE 230.00
```

6.10 HARMonics Group

The commands in this group deal with harmonic measurement.
You can make the same settings and inquiries as when the HRM SET on the front panel is used.
However, the commands in this group are valid only when the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option) is installed.

:HARMonics?

Function Queries all settings related to harmonic measurement.

```
Syntax    :HARMonics?
Example   HARMONICS? -> :HARMONICS:
          FBAND NORMAL;PLLSOURCE U1;
          ORDER 1,100;THD TOTAL;IEC:
          OBJECT ELEMENT1;UGROUPING OFF;
          IGROUPING OFF;:HARMONICS:
          PLLWARNING:STATE 1
```

:HARMonics:FBANd

Function Sets the frequency bandwidth of the harmonic measurement or queries the current setting.

```
Syntax    :HARMonics:FBANd {NORMal|WIDE}
          :HARMonics:FBANd?
Example   :HARMONICS:FBAND NORMAL
          :HARMONICS:FBAND? ->
          :HARMONICS:FBAND NORMAL
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- For details on the frequency bandwidth corresponding to {NORMal|WIDE}, see the *Expansion Function User's Manual IM760301-51E.*

:HARMonics:IEC?

Function Queries all settings related to IEC harmonic measurement.

```
Syntax    :HARMonics:IEC?
Example   :HARMONICS:IEC? -> :HARMONICS:IEC:
          OBJECT ELEMENT1;UGROUPING OFF;
          IGROUPING OFF
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:HARMonics:IEC:OBJect

Function Sets the IEC harmonic measurement target or queries the current setting.

```
Syntax    :HARMonics:IEC:OBJect {ELEMent<x>|
          SIGMA|SIGMB}
          :HARMonics:IEC:OBJect?
```

<x> = 1 to 4 (element)

```
Example   :HARMONICS:IEC:OBJECT ELEMENT1
          :HARMONICS:IEC:OBJECT? ->
          :HARMONICS:IEC:OBJECT ELEMENT1
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:HARMonics:IEC:{UGRouping|IGRouping}

Function Sets the {voltage|current} grouping of the IEC harmonic measurement or queries the current setting.

```
Syntax    :HARMonics:IEC:{UGRouping|
          IGRouping} {OFF|TYPE1|TYPE2}
          :HARMonics:IEC:{UGRouping|
          IGRouping}?
Example   :HARMONICS:IEC:UGROUPING OFF
          :HARMONICS:IEC:UGROUPING? ->
          :HARMONICS:IEC:UGROUPING OFF
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- For details on the grouping corresponding to {OFF|TYPE1|TYPE2}, see the *Expansion Function User's Manual IM760301-51E.*

:HARMonics:ORDer

Function Sets the maximum and minimum orders to be measured or queries the current setting.

```
Syntax    :HARMonics:ORDer {<NRf>,<NRf>}
          :HARMonics:ORDer?
```

1st <NRf> = 0 or 1 (minimum order to be measured)
2nd <NRf> = 1 to 100 (maximum order to be measured)

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

:HARMonics:PLLSource

Function Sets the PLL source or queries the current setting.

Syntax
```
:HARMonics:PLLSource {U<x>|I<x>|
EXTernal|SAMPle}
:HARMonics:PLLSource?
```
<x> = 1 to 4 (element)
EXTernal = External clock input (Ext Clk)
SAMPle = Sampling clock input (Smp Clk)

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

Description
- “SAMPle” is selectable only on models with the advanced computation function (/G6 option).
- If SAMPle is selected, it is used in wide bandwidth harmonic measurement mode. In other measurement modes, EXTernal is used. “EXTernal” is also returned in response to a query.

:HARMonics:PLLWarning?

Function Queries all settings related to the warning messages of the PLL source.

Syntax
```
:HARMonics:PLLWarning?
```

Example
```
:HARMONICS:PLLWARNING? ->
:HARMONICS:PLLWARNING:STATE 1
```

Description This command is valid only on models with the advanced computation function (/G6 option).

:HARMonics:PLLWarning[:STATe]

Function Sets whether to generate a warning message when the PLL source is not applied or queries the current setting.

Syntax
```
:HARMonics:PLLWarning
[:STATe] {<Boolean>}
:HARMonics:PLLWarning:STATe?
```

Example
```
:HARMONICS:PLLWARNING:STATE ON
:HARMONICS:PLLWARNING:STATE? ->
:HARMONICS:PLLWARNING:STATE 1
```

Description
- This command is valid only on models with the advanced computation function (/G6 option).
- This setting is valid only in wide bandwidth harmonic measurement mode.

:HARMonics:THD

Function Sets the equation used to calculate the THD (total harmonic distortion) or queries the current setting.

Syntax
```
:HARMonics:THD {TOTal|FUNDamental}
:HARMonics:THD?
```

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

6.11 HCOPy Group

The commands in this group deal with printing to the built-in printer or network printer.
You can make the same settings and inquiries as when PRINT and MENU (SHIFT+PRINT) on the front panel is used.
However, the commands in this group are valid only when the built-in printer (/B5 option) or Ethernet interface (/C7 option) is installed.

:HCOPy? (Hard COPY)

Function Queries all settings related to the printing.

Syntax `:HCOPy?`

Example
```
:HCOPY? -> :HCOPY:
DIRECTION PRINTER;PRINTER:
FORMAT HCOPY;:HCOPY:AUTO:STATE 0;:
HCOPY:COMMENT "THIS IS TEST."
```

:HCOPy:ABORt

Function Aborts printing or paper feeding.

Syntax `:HCOPy:ABORt`

Example `:HCOPY:ABORT`

:HCOPy:AUTO?

Function Queries all settings related to the auto print.

Syntax `:HCOPy:AUTO?`

Example
```
:HCOPY:AUTO? -> :HCOPY:AUTO:
STATE 1;SYNCHRONIZE TIMER;
START 2005,1,1,0,0,0;
END 2005,1,1,1,0,0;INTERVAL 0,0,10
```

:HCOPy:AUTO:INTerval

Function Sets the auto print interval or queries the current setting.

Syntax
```
:HCOPy:AUTO:INTerval
{<NRf>,<NRf>,<NRf>}
:HCOPy:AUTO:INTerval?
```
{<NRf>, <NRf>, <NRf>} = 0, 0, 10 to 99, 59, 59
1st <NRf> = 0 to 99 (hour)
2nd <NRf> = 0 to 59 (minute)
3rd <NRf> = 0 to 59 (second)

Example
```
:HCOPY:AUTO:INTERVAL 0,0,10
:HCOPY:AUTO:INTERVAL? ->
:HCOPY:AUTO:INTERVAL 0,0,10
```

:HCOPy:AUTO:{STARt|END}

Function Sets the {start|stop} reservation time of the auto print or queries the current setting.

Syntax
```
:HCOPy:AUTO:{STARt|END}
{<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,
<NRf>}
:HCOPy:AUTO:{STARt|END}?
```
{<NRf>, <NRf>, <NRf>, <NRf>, <NRf>, <NRf>} = 2001, 1, 1, 0, 0, 0 to 2099, 12, 31, 23, 59, 59

1st <NRf> = 2001 to 2099	(year)
2nd <NRf> = 1 to 12	(month)
3rd <NRf> = 1 to 31	(day)
4th <NRf> = 0 to 23	(hour)
5th <NRf> = 0 to 59	(minute)
6th <NRf> = 0 to 59	(second)

Example
```
:HCOPY:AUTO:START 2005,1,1,0,0,0
:HCOPY:AUTO:START? ->
:HCOPY:AUTO:START 2005,1,1,0,0,0
```

Description This setting is valid when the synchronization mode (`:HCOPy:AUTO:SYNChronize`) is set to TIMer (timer synchronized printing).

:HCOPy:AUTO[:STATe]

Function Turns ON/OFF the auto print or queries the current setting.

Syntax
```
:HCOPy:AUTO[:STATe] {<Boolean>}
:HCOPy:AUTO:STATe?
```

Example
```
:HCOPY:AUTO:STATE ON
:HCOPY:AUTO:STATE? ->
:HCOPY:AUTO:STATE 1
```

:HCOPy:AUTO:SYNChronize

Function Sets the synchronization mode of the auto print or queries the current setting.

Syntax
```
:HCOPy:AUTO:SYNChronize {TIMer|
INTEGrate}
:HCOPy:AUTO:SYNChronize?
```
TIMer = Timer synchronized printing
INTEGrate = Integration synchronized printing

Example
```
:HCOPY:AUTO:SYNCHRONIZE TIMER
:HCOPY:AUTO:SYNCHRONIZE? ->
:HCOPY:AUTO:SYNCHRONIZE TIMER
```

:HCOPy:COMMent

Function Sets the comment displayed at the bottom of the screen or queries the current setting.

Syntax `:HCOPy:COMMent {<String>}`
`:HCOPy:COMMent?`
<String > = 25 characters or less (However, only the first 20 characters are displayed.)

Example `:HCOPY:COMMENT "THIS IS TEST."`
`:HCOPY:COMMENT? -> :HCOPY:COMMENT "THIS IS TEST."`

:HCOPy:DIRection

Function Sets the printer or queries the current setting.

Syntax `:HCOPy:DIRection {PRINter|NETPrint}`
`:HCOPy:DIRection?`
PRINter = Built-in printer
NETPrint = Network printer

Example `:HCOPY:DIRECTION PRINTER`
`:HCOPY:DIRECTION? -> :HCOPY:DIRECTION PRINTER`

Description
- PRINTer is valid only when the built-in printer (/B5 option) is installed.
- NETPrint is valid only when the Ethernet interface (/C7 option) is installed.

:HCOPy:EXECute

Function Executes printing.

Syntax `:HCOPy:EXECute`

Example `:HCOPY:EXECUTE`

Description This command is an overlap command.

:HCOPy:NETPrint?

Function Queries all settings related to the printing on the network printer.

Syntax `:HCOPy:NETPrint?`

Example `:HCOPY:NETPRINT? -> :HCOPY:NETPRINT:FORMAT BJ,180;COLOR 0`

Description This command is valid only on models with the Ethernet interface (/C7 option).

:HCOPy:NETPrint:COLor

Function Turns ON/OFF color printing on the network printer or queries the current setting.

Syntax `:HCOPy:NETPrint:COLor {<Boolean>}`
`:HCOPy:NETPrint:COLor?`

Example `:HCOPY:NETPRINT:COLOR OFF`
`:HCOPY:NETPRINT:COLOR? -> :HCOPY:NETPRINT:COLOR 0`

Description This command is valid only on models with the Ethernet interface (/C7 option).

:HCOPy:NETPrint:FORMat

Function Sets the printer description language for printing on a network printer or queries the current setting.

Syntax `:HCOPy:NETPrint:FORMat {PCL5|LIPS3|BJ,<NRf>}`
`:HCOPy:NETPrint:FORMat?`
<NRf> = 180, 300, or 360 (dpi, resolution)

Example `:HCOPY:NETPRINT:FORMAT BJ,180`
`:HCOPY:NETPRINT:FORMAT? -> :HCOPY:NETPRINT:FORMAT BJ,180`

Description
- Set <NRf> only when BJ is selected.
- This command is valid only on models with the Ethernet interface (/C7 option).

:HCOPy:PRINter?

Function Queries all settings related to printing on the built-in printer.

Syntax `:HCOPy:PRINter?`

Example `:HCOPY:PRINTER? -> :HCOPY:PRINTER:FORMAT HCOPY`

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:FEED

Function Executes paper feeding of the built-in printer.

Syntax `:HCOPy:PRINter:FEED`

Example `:HCOPY:PRINTER FEED`

Description
- This command is valid only when the built-in printer (/B5 option) is installed.
- This command is an overlap command.

:HCOPy:PRINter:FORMat

Function Sets the contents to be printed on the built-in printer or queries the current setting.

Syntax `:HCOPy:PRINter:FORMat {HCOPy|LIST}`
`:HCOPy:PRINter:FORMat?`
HCOPy = Screen image data
LIST = Numeric data list

Example `:HCOPY:PRINTER:FORMAT HCOPY`
`:HCOPY:PRINTER:FORMAT? -> :HCOPY:PRINTER:FORMAT HCOPY`

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:LIST?

Function Queries all settings related to the printing of the numeric data list on the built-in printer.

Syntax
```
:HCOPy:PRINter:LIST?
```

Example
```
:HCOPY:PRINTER:LIST? -> :HCOPY:
PRINTER:LIST:INFORMATION 1;NORMAL:
ELEMENT1 1;ELEMENT2 0;ELEMENT3 0;
ELEMENT4 0;SIGMA 0;SIGMB 0;U 1;I 1;
P 1;S 1;Q 1;LAMBDA 1;PHI 1;FU 1;
FI 1;UPPEAK 0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0
```

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:LIST:INFOrmation

Function Sets whether to add setup parameters when printing the numeric data list on the built-in printer or queries the current setting.

Syntax
```
:HCOPy:PRINter:LIST:INFOrmation
{<Boolean>}
:HCOPy:PRINter:LIST:INFOrmation?
```

Example
```
:HCOPY:PRINTER:LIST:INFORMATION ON
:HCOPY:PRINTER:LIST:INFORMATION? ->
:HCOPY:PRINTER:LIST:INFORMATION 1
```

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:LIST:NORMal?

Function Queries all settings related to the printed items of the numeric data list using the built-in printer.

Syntax
```
:HCOPy:PRINter:LIST:NORMal?
```

Example
```
:HCOPY:PRINTER:LIST:NORMAL? ->
:HCOPY:PRINTER:LIST:NORMAL:
ELEMENT1 1;ELEMENT2 0;ELEMENT3 0;
ELEMENT4 0;SIGMA 0;SIGMB 0;U 1;I 1;
P 1;S 1;Q 1;LAMBDA 1;PHI 1;FU 1;
FI 1;UPPEAK0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0
```

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:LIST[:NORMal]:ALL

Function Collectively turns ON/OFF the output of all element functions when printing the numeric data list on the built-in printer.

Syntax
```
:HCOPy:PRINter:LIST[:NORMal]:
ALL {<Boolean>}
```

Example
```
:HCOPY:PRINTER:LIST:NORMAL:ALL ON
```

Description This command is valid only when the built-in printer (/B5 option) is installed.

:HCOPy:PRINter:LIST[:NORMal]:{ELEMent<x>|SIGMA|SIGMB}

Function Turns ON/OFF the output of {each element|ΣA|ΣB} when printing the numeric data list on the built-in printer.

Syntax
```
:HCOPy:PRINter:LIST[:NORMal]:
{ELEMent<x>|SIGMA|SIGMB}
{<Boolean>}
:HCOPy:PRINter:LIST[:NORMal]:
{ELEMent<x>|SIGMA|SIGMB}?
```
<x> = 1 to 4

Example
```
:HCOPY:PRINTER:LIST:NORMAL:
ELEMENT1 ON
:HCOPY:PRINTER:LIST:NORMAL:
ELEMENT1? -> :HCOPY:PRINTER:LIST:
NORMAL:ELEMENT1 1
```

Description
- This command is valid only when the built-in printer (/B5 option) is installed.
- `:HCOPy:PRINter:LIST[:NORMal]:SIGMA` is valid on models with two or more elements. To turn the output ON, wiring unit ΣA must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.
- `:HCOPy:PRINter:LIST[:NORMal]:SIGMB` is valid on models with four elements. To turn the output ON, wiring unit ΣB must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.

:HCOPy:PRINter:LIST[:NORMal]:PRESet<x>

Function Presets the output ON/OFF pattern of the element functions when printing the numeric data list on the built-in printer.

Syntax
```
:HCOPy:PRINter:LIST[:NORMal]:
PRESet<x>
```
<x> = 1 to 2 (preset pattern number)

Example
```
:HCOPY:PRINTER:LIST:NORMAL:PRESET1
```

Description
- This command is valid only when the built-in printer (/B5 option) is installed.
- For details on the print pattern when preset is executed, see the *Expansion Function User's Manual IM760301-51E*.

:HCOPy:PRINter:LIST[:NORMal]:<Function>

Function urns ON/OFF the output of the function when printing the numerical data list using the built-in printer or queries the current setting.

Syntax
```
:HCOPy:PRINter:LIST[:NORMal]:
<Function> {<Boolean>}
:HCOPy:PRINter:LIST[:NORMal]:
<Function>?
```
<Function> = {U|I|P|S|Q|...}(See the function selection list (1) of "DISPlay group" on page 6-44.)

Example
```
:HCOPY:PRINTER:LIST:NORMAL:U ON
:HCOPY:PRINTER:LIST:NORMAL:U? ->
:HCOPY:PRINTER:LIST:NORMAL:U 1
```

Description This command is valid only when the built-in printer (/B5 option) is installed.

6.12 HOLD Group

The commands in this group deal with the hold function of output data.
You can make the same settings and inquiries as when HOLD on the front panel is used.

:HOLD

Function Sets the output data (display, communications, etc.) hold or queries the current setting.

```
Syntax    :HOLD {<Boolean>}
          :HOLD?
Example   :HOLD OFF
          :HOLD? -> :HOLD 0
```

6.13 IMAGe Group

The commands in this group deal with the saving of screen image data.
You can make the same settings and inquiries as when IMAGE SAVE and MENU (SHIFT+ IMAGE SAVE) on the front panel is used.

:IMAGe?

Function Queries all settings related to the saving of screen image data.

Syntax `:IMAGe?`

Example `:IMAGE? -> :IMAGE:FORMAT TIFF;`
`COLOR OFF;COMMENT "THIS IS TEST.";`
`SAVE:ANAMING 1;NAME ""`

:IMAGe:ABORt

Function Aborts the saving of the screen image data.

Syntax `:IMAGe:ABORt`

Example `:IMAGE:ABORT`

:IMAGe:COLor

Function Sets the color tone of the screen image data to be saved or queries the current setting.

Syntax `:IMAGe:COLor {OFF|COLor|REVerse|GRAY}`
`:IMAGe:COLor?`

Example `:IMAGE:COLOR OFF`
`:IMAGE:COLOR? -> :IMAGE:COLOR OFF`

Description This command is valid when the format (:IMAGe:FORMat`) is not PSCRipt.`

:IMAGe:COMMent

Function Sets the comment displayed at the bottom of the screen or queries the current setting.

Syntax `:IMAGe:COMMent {<String>}`
`:IMAGe:COMMent?`
<String > = 25 characters or less (However, only the first 20 characters are displayed.)

Example `:IMAGE:COMMENT "THIS IS TEST."`
`:IMAGE:COMMENT? -> :IMAGE:`
`COMMENT "THIS IS TEST."`

:IMAGe:COMPression

Function Enables or disables the data compression of screen image data in BMP format or queries the current setting.

Syntax `:IMAGe:COMPression {<Boolean>}`
`:IMAGe:COMPression?`

Example `:IMAGE:COMPRESSION ON`
`:IMAGE:COMPRESSION? ->`
`:IMAGE:COMPRESSION 1`

Description This command is valid when the format (`:IMAGe:FORMat`) is BMP and the color tone (:IMAGe:COLor`) is`
{COLor|REVerse|GRAY}.

:IMAGe:EXECute

Function Saves the screen image data.

Syntax `:IMAGe:EXECute`

Example `:IMAGE:EXECUTE`

:IMAGe:FORMat

Function Sets the format of the screen image data to be saved or queries the current setting.

Syntax `:IMAGe:FORMat {TIFF|BMP|PSCRipt|PNG|JPEG}`
`:IMAGe:FORMat?`

Example `:IMAGE:FORMAT TIFF`
`:IMAGE:FORMAT? ->`
`:IMAGE:FORMAT TIFF`

:IMAGe:SAVE?

Function Queries all settings related to the saving of screen image data.

Syntax `:IMAGe:SAVE?`

Example `:IMAGE:SAVE? ->`
`:IMAGE:SAVE:ANAMING 1;NAME ""`

:IMAGe:SAVE:ANAMing

Function Sets whether to automatically name the screen image data files to be saved or queries the current setting.

Syntax `:IMAGe:SAVE:ANAMing {<Boolean>}`
`:IMAGe:SAVE:ANAMing?`

Example `:IMAGE:SAVE:ANAMING ON`
`:IMAGE:SAVE:ANAMING? ->`
`:IMAGE:SAVE:ANAMING 1`

:IMAGe:SAVE:CDIRectory

Function Changes the save destination directory for the screen image data.

Syntax `:IMAGe:CDIRectory {<Filename>}`
<Filename> = Directory name

Example `:IMAGE:CDIRECTORY "IMAGE"`

Description Specify “`..`” to move up to the parent directory.

:IMAGe:SAVE:DRIVe

Function Sets the save destination drive of the screen image data.

Syntax `:IMAGe:SAVE:DRIVe {PCCard[,<NRf>]|NETWork|USB,<NRf>[,<NRf>][,<NRf>]}`

PCCard = PC card drive
<NRf> = Partition (0 to 3)
NETWork = Network drive
USB = USB memory drive
1st <NRf> = ID number (address)
2nd <NRf> = Partition (0 to 3) or LUN (logical unit number: 0 to 3)
3rd <NRf> = Partition (0 to 3) when LUN is specified

Example `:IMAGE:SAVE:DRIVE PCCARD`

Description
- If the drive does not contain partitions, omit the <NRf> corresponding to partitions.
- "NETWork" can be used when the Ethernet interface (/C7 option) is installed.
- "USB" can be used when the USB port (peripheral device) (/C5 option) is installed.
- The second or third <NRf> when USB is selected can be omitted if the drive is not partitioned or divided by LUN.

:IMAGe:SAVE:NAME

Function Sets the name of the file for saving the screen image data or queries the current setting.

Syntax
```
:IMAGe:SAVE:NAME {<Filename>}
:IMAGe:SAVE:NAME?
```

Example
```
:IMAGE:SAVE:NAME "IMAGE1"
:IMAGE:SAVE:NAME? ->
:IMAGE:SAVE:NAME "IMAGE1"
```

Description
- Set the save destination drive with the "`:IMAGe:SAVE:DRIVe`" command and the directory with the "`:IMAGe:SAVE:CDIRectory`" command.
- Specify the file name without the extension.

:IMAGe:SEND?

Function Queries the screen image data.

Syntax `:IMAGe:SEND?`

Example `:IMAGE:SEND? -> #6(number of bytes, 6 digits)(data byte sequence)`

Description
- The number of bytes of <Block data> is `{2 + 6 +` number of data points `+1` (delimiter)`}`.
- For details on <Block data>, see page 5-6.

6.14 INPut Group

The commands in this group deal with the measurement condition of the input element.
You can make the same settings and inquiries as when the keys in the measurement condition setup area (area enclosed in light blue), SCALING, LINE FILTER, FREQ FILTER (SHIFT+LINE FILTER), SYNC SOURCE, and NULL(SHIFT+SYNC SOURCE) on the front panel are used.

:INPut?

Function Queries all settings related to the input element.

Syntax `:INPut?`

Example
```
:INPUT? -> :INPUT:CFACTOR 3;
WIRING P1W2,P1W2,P1W2,P1W2;
INDEPENDENT 0;VOLTAGE:RANGE:
ELEMENT1 1.000E+03;
ELEMENT2 1.000E+03;
ELEMENT3 1.000E+03;
ELEMENT4 1.000E+03;:INPUT:VOLTAGE:
AUTO:ELEMENT1 0;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;:INPUT:
VOLTAGE:MODE:ELEMENT1 RMS;
ELEMENT2 RMS;ELEMENT3 RMS;
ELEMENT4 RMS;:INPUT:CURRENT:RANGE:
ELEMENT1 30.0E+00;
ELEMENT2 30.0E+00;
ELEMENT3 30.0E+00;
ELEMENT4 30.0E+00;:INPUT:CURRENT:
AUTO:ELEMENT1 0;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;:INPUT:
CURRENT:MODE:ELEMENT1 RMS;
ELEMENT2 RMS;ELEMENT3 RMS;
ELEMENT4 RMS;:INPUT:CURRENT:SRATIO:
ELEMENT1 10.0000;ELEMENT2 10.0000;
ELEMENT3 10.0000;ELEMENT4 10.0000;:
INPUT:FILTER:LINE:ELEMENT1 OFF;
ELEMENT2 OFF;ELEMENT3 OFF;
ELEMENT4 OFF;:INPUT:FILTER:
FREQUENCY:ELEMENT1 0;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;:INPUT:
SCALING:STATE:ELEMENT1 0;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;:
INPUT:SCALING:VT:ELEMENT1 1.0000;
ELEMENT2 1.0000;ELEMENT3 1.0000;
ELEMENT4 1.0000;:INPUT:SCALING:CT:
ELEMENT1 1.0000;ELEMENT2 1.0000;
ELEMENT3 1.0000;ELEMENT4 1.0000;:
INPUT:SCALING:SFACTOR:
ELEMENT1 1.0000;ELEMENT2 1.0000;
ELEMENT3 1.0000;ELEMENT4 1.0000;:
INPUT:SYNCHRONIZE:ELEMENT1 I1;
ELEMENT2 I2;ELEMENT3 I3;
ELEMENT4 I4;:INPUT:NULL 0
```

[:INPut]:CFACtor

Function Sets the crest factor or queries the current setting.

Syntax
```
[:INPut]:CFACtor {<NRf>}
[:INPut]:CFACtor?
```
<NRf> = 3 or 6

Example
```
:INPUT:CFACTOR 3
:INPUT:CFACTOR? -> :INPUT:CFACTOR 3
```

[:INPut]:CURRent?

Function Queries all settings related to the current measurement.

Syntax `[:INPut]:CURRent?`

Example
```
:INPUT:CURRENT? -> :INPUT:CURRENT:
RANGE:ELEMENT1 30.0E+00;
ELEMENT2 30.0E+00;
ELEMENT3 30.0E+00;
ELEMENT4 30.0E+00;:INPUT:CURRENT:
AUTO:ELEMENT1 0;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;:INPUT:
CURRENT:MODE:ELEMENT1 RMS;
ELEMENT2 RMS;ELEMENT3 RMS;
ELEMENT4 RMS;:INPUT:CURRENT:SRATIO:
ELEMENT1 10.0000;ELEMENT2 10.0000;
ELEMENT3 10.0000;ELEMENT4 10.0000
```

[:INPut]:CURRent:AUTO?

Function Queries the current auto range setting (ON/OFF) of all elements.

Syntax `[:INPut]:CURRent:AUTO?`

Example
```
:INPUT:CURRENT:AUTO? ->
:INPUT:CURRENT:AUTO:ELEMENT1 0;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0
```

[:INPut]:CURRent:AUTO[:ALL]

Function Collectively turns ON/OFF the current auto range of all elements.

Syntax
```
[:INPut]:CURRent:AUTO
[:ALL] {<Boolean>}
```

Example `:INPUT:CURRENT:AUTO:ALL ON`

[:INPut]:CURRent:AUTO:ELEMent<x>

Function Turns ON/OFF the current auto range of the element or queries the current setting.

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
```

[:INPut]:CURRent:AUTO:{SIGMA|SIGMB}

Function Collectively turns ON/OFF the current auto range of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax
```
[:INPut]:CURRent:AUTO:{SIGMA|
SIGMB} {<Boolean>}
```

Example
```
:INPUT:CURRENT:AUTO:SIGMA ON
```

Description
- [:INPut]:CURRent:AUTO:SIGMA is valid only on models with 2 to 4 elements.
- [:INPut]:CURRent:AUTO:SIGMB is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting ([:INPut]:WIRing).

[:INPut]:CURRent:MODE?

Function Queries the current mode of all elements.

Syntax
```
[:INPut]:CURRent:MODE?
```

Example
```
:INPUT:CURRENT:MODE? ->
:INPUT:CURRENT:MODE:ELEMENT1 RMS;
ELEMENT2 RMS;ELEMENT3 RMS;
ELEMENT4 RMS
```

[:INPut]:CURRent:MODE[:ALL]

Function Collectively sets the current mode of all elements.

Syntax
```
[:INPut]:CURRent:MODE[:ALL] {RMS|
MEAN|DC|RMEAN}
```

Example
```
:INPUT:CURRENT:MODE:ALL RMS
```

[:INPut]:CURRent:MODE:ELEMent<x>

Function Sets the current mode of the element or queries the current setting.

Syntax
```
[:INPut]:CURRent:MODE:
ELEMent<x> {RMS|MEAN|DC|RMEAN}
[:INPut]:CURRent:MODE:ELEMent<x>?
```
<x> = 1 to 4 (element)

Example
```
:INPUT:CURRENT:MODE:ELEMENT1 RMS
:INPUT:CURRENT:MODE:ELEMENT1? ->
:INPUT:CURRENT:MODE:ELEMENT1 RMS
```

[:INPut]:CURRent:MODE:{SIGMA|SIGMB}

Function Collectively sets the current mode of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax
```
[:INPut]:CURRent:MODE:{SIGMA|
SIGMB} {RMS|MEAN|DC|RMEAN}
```

Example
```
:INPUT:CURRENT:MODE:SIGMA RMS
```

Description
- [:INPut]:CURRent:MODE:SIGMA is valid only on models with 2 to 4 elements.
- [:INPut]:CURRent:MODE:SIGMB is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting ([:INPut]:WIRing).

[:INPut]:CURRent:RANGe?

Function Queries the current ranges of all elements.

Syntax
```
[:INPut]:CURRent:RANGe?
```

Example
```
:INPUT:CURRENT:RANGE? -> :INPUT:
CURRENT:RANGE:ELEMENT1 30.0E+00;
ELEMENT2 30.0E+00;
ELEMENT3 30.0E+00;
ELEMENT4 30.0E+00
```

[:INPut]:CURRent:RANGe[:ALL]

Function Collectively sets the current ranges of all elements.

Syntax
```
[:INPut]:CURRent:RANGe[:ALL]
{<Current>|(EXTernal,<Voltage>)}
```

With the (standard) power element models (760301-01 through 760304-04)

- When the crest factor is set to 3
  <Current> = 500 (mA), 1, 2, 5, 10, 20, 30 (A) (for direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (for external current sensor input)
- When the crest factor is set to 6
  <Current> = 250, 500 (mA), 1, 2.5, 5, 10, 15 (A) (for direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (for external current sensor input)

With the low current range power element models (760301-10 through 760304-40)

- When crest factor is set to 3
  <Current> = 5, 10, 20, 50, 100, 200, 500 (mA), 1, 2, (A) (with direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (with external current sensor input)
- When crest factor is set to 6
  <Current> = 2.5, 5, 10, 25, 50, 100, 250, 500 (mA), 1 (A) (with direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (with external current sensor input)

Example
```
:INPUT:CURRENT:RANGE:ALL 30A
:INPUT:CURRENT:RANGE:
ALL EXTERNAL,10V
```

[:INPut]:CURRent:RANGe:ELEMent<x>

Function Sets the current range of the element or queries the current setting.

Syntax
```
[:INPut]:CURRent:RANGe:
ELEMent<x>
{<Current>|(EXTernal,<Voltage>)}
[:INPut]:CURRent:RANGe:ELEMent<x>?
```
<x> = 1 to 4 (element)

With the (standard) power element models (760301-01 through 760304-04)

- When the crest factor is set to 3
  <Current> = 500 (mA), 1, 2, 5, 10, 20, 30 (A) (for direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (for external current sensor input)
- When the crest factor is set to 6
  <Current> = 250, 500 (mA), 1, 2.5, 5, 10, 15 (A) (for direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (for external current sensor input)

With the low current range power element models (760301-10 through 760304-40)

- When crest factor is set to 3
  <Current> = 5, 10, 20, 50, 100, 200, 500 (mA), 1, 2, (A) (with direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (with external current sensor input)
- When crest factor is set to 6
  <Current> = 2.5, 5, 10, 25, 50, 100, 250, 500 (mA), 1 (A) (with direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (with external current sensor input)

Example
```
:INPUT:CURRENT:RANGE:ELEMENT1 30A
:INPUT:CURRENT:RANGE:ELEMENT1? ->
:INPUT:CURRENT:RANGE:
ELEMENT1 30.0E+00

:INPUT:CURRENT:RANGE:
ELEMENT1 EXTERNAL,10V
:INPUT:CURRENT:RANGE:ELEMENT1? ->
:INPUT:CURRENT:RANGE:
ELEMENT1 EXTERNAL,10.00E+00
```

[:INPut]:CURRent:RANGe:{SIGMA|SIGMB}

Function Collectively sets the current range of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax
```
[:INPut]:CURRent:RANGe:{SIGMA|
SIGMB}
{<Current>|(EXTernal,<Voltage>)}
```

With the (standard) power element models (760301-01 through 760304-04)

- When the crest factor is set to 3
  <Current> = 500 (mA), 1, 2, 5, 10, 20, 30 (A) (for direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (for external current sensor input)
- When the crest factor is set to 6
  <Current> = 250, 500 (mA), 1, 2.5, 5, 10, 15 (A) (for direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (for external current sensor input)

With the low current range power element models (760301-10 through 760304-40)

- When crest factor is set to 3
  <Current> = 5, 10, 20, 50, 100, 200, 500 (mA), 1, 2, (A) (with direct current input)
  <Voltage> = 50, 100, 200, 500 (mV), 1, 2, 5, 10 (V) (with external current sensor input)
- When crest factor is set to 6
  <Current> = 2.5, 5, 10, 25, 50, 100, 250, 500 (mA), 1 (A) (with direct current input)
  <Voltage> = 25, 50, 100, 250, 500 (mV), 1, 2.5, 5 (V) (with external current sensor input)

Example
```
:INPUT:CURRENT:RANGE:SIGMA 30A
```

Description
- `[:INPut]:CURRent:RANGe:SIGMA` is valid only on models with 2 to 4 elements.
- `[:INPut]:CURRent:RANGe:SIGMB` is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting (`[:INPut]:WIRing`).

[:INPut]:CURRent:SRATio?

Function Queries the current sensor scaling constants of all elements.

Syntax
```
[:INPut]:CURRent:SRATio?
```
Example
```
:INPUT:CURRENT:SRATIO? -> :INPUT:
CURRENT:SRATIO:ELEMENT1 10.0000;
ELEMENT2 10.0000;ELEMENT3 10.0000;
ELEMENT4 10.0000
```

[:INPut]:CURRent:SRATio[:ALL]

Function Collectively sets the current sensor scaling constants of all elements.

Syntax
```
[:INPut]:CURRent:SRATio
[:ALL] {<NRf>}
```
<NRf> = 0.0001 to 99999.9999

Example
```
:INPUT:CURRENT:SRATIO:ALL 10
```

[:INPut]:CURRent:SRATio:ELEMent<x>

Function Sets the current sensor scaling constant of the element or queries the current setting.

Syntax
```
[:INPut]:CURRent:SRATio:
ELEMent<x> {<NRf>}
[:INPut]:CURRent:SRATio:ELEMent<x>?
```
<x> = 1 to 4 (element)
<NRf> = 0.0001 to 99999.9999

Example
```
:INPUT:CURRENT:SRATIO:ELEMENT1 10
:INPUT:CURRENT:SRATIO:ELEMENT1? ->
:INPUT:CURRENT:SRATIO:
ELEMENT1 10.0000
```

[:INPut]:FILTer?

Function Queries all settings related to the filter.

Syntax
```
[:INPut]:FILTer?
```
Example
```
:INPUT:FILTER? -> :INPUT:FILTER:
LINE:ELEMENT1 OFF;ELEMENT2 OFF;
ELEMENT3 OFF;ELEMENT4 OFF;:INPUT:
FILTER:FREQUENCY:ELEMENT1 0;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0
```

[:INPut]:FILTer:FREQuency?

Function Queries the frequency filter settings of all elements.

Syntax
```
[:INPut]:FILTer:FREQuency?
```
Example
```
:INPUT:FILTER:FREQUENCY? ->
:INPUT:FILTER:FREQUENCY:ELEMENT1 0;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0
```

[:INPut]:FILTer:FREQuency[:ALL]

Function Collectively sets the frequency filter of all elements.

Syntax
```
[:INPut]:FILTer:FREQuency
[:ALL] {<Boolean>}
```
Example
```
:INPUT:FILTER:FREQUENCY:ALL OFF
```

[:INPut]:FILTer:FREQuency:ELEMent<x>

Function Sets the frequency filter of the element or queries the current setting.

Syntax
```
[:INPut]:FILTer:FREQuency:
ELEMent<x> {<Boolean>}
[:INPut]:FILTer:FREQuency:
ELEMent<x>?
```
<x> = 1 to 4 (element)

Example
```
:INPUT:FILTER:FREQUENCY:ELEMENT1 ON
:INPUT:FILTER:FREQUENCY:ELEMENT1?
-> :INPUT:FILTER:FREQUENCY:
ELEMENT1 1
```

[:INPut]:FILTer:LINE?

Function Queries the line filter settings of all elements.

Syntax
```
[:INPut]:FILTer:LINE?
```
Example
```
:INPUT:FILTER:LINE? ->
:INPUT:FILTER:LINE:ELEMENT1 OFF;
ELEMENT2 OFF;ELEMENT3 OFF;
ELEMENT4 OFF
```

[:INPut]:FILTer[:LINE][:ALL]

Function Collectively sets the line filters of all elements.

Syntax
```
[:INPut]:FILTer[:LINE][:ALL]
{OFF|<Frequency>}
```
OFF = Line filter OFF
<Frequency> = 500 Hz, 5.5 kHz, or 50 kHz (line filter ON, cutoff frequency)

Example
```
:INPUT:FILTER:LINE:ALL OFF
```

[:INPut]:FILTer[:LINE]:ELEMent<x>

Function Sets the line filter of the element or queries the current setting.

Syntax
```
[:INPut]:FILTer[:LINE]:
ELEMent<x> {OFF|<Frequency>}
[:INPut]:FILTer[:LINE]:ELEMent<x>?
```
<x> = 1 to 4 (element)
OFF = Line filter OFF
<Frequency> = 500 Hz, 5.5 kHz, 50 kHz (line filter ON, cutoff frequency)

Example
```
:INPUT:FILTER:LINE:ELEMENT1 OFF
:INPUT:FILTER:LINE:ELEMENT1? ->
:INPUT:FILTER:LINE:ELEMENT1 OFF
```

[:INPut]:INDependent

Function Turns ON/OFF the independent setting of input elements or queries the current setting.

Syntax
```
[:INPut]:INDependent {<Boolean>}
[:INPut]:INDependent?
```
Example
```
:INPUT:INDEPENDENT OFF
:INPUT:INDEPENDENT? ->
:INPUT:INDEPENDENT 0
```
Description This command is valid only on models with 2 to 4 elements.

[:INPut]:MODUle?

Function Queries the input element type.

Syntax `[:INPut]:MODUle? {<NRf>}`
`[:INPut]:MODUle?`
<NRf> = 1 to 4 (element)

Example `:INPUT:MODULE? 1 -> 30`
`:INPUT:MODULE? -> 30,30,30,30`

Description
- The response information is as follows:
  30 = (standard) power element (max. current range = 30 A)
  2 = low current range power element (max. current range = 2 A)
  0 = No input element
- If the parameter is omitted, the input element types of all elements are output in order starting with element 1.

[:INPut]:NULL

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

Syntax `[:INPut]:NULL {<Boolean>}`
`[:INPut]:NULL?`

Example `:INPUT:NULL ON`
`:INPUT:NULL? -> :INPUT:NULL 1`

[:INPut]:POVer?

Function Queries the peak over information.

Syntax `[:INPut]:POVer?`

Example `:INPUT:POVER? -> 0`

Description
- The peak over information of each element is mapped as shown below. A sum of decimal values of each bit is returned for the response.
- For example, if the response is “16,” for example, peak over is occurring at U3.

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

Sp: Rotating speed
Tq: Torque

[:INPut]:SCALing?

Function Queries all settings related to scaling.

Syntax `[:INPut]:SCALing?`

Example `:INPUT:SCALING? -> :INPUT:SCALING:STATE:ELEMENT1 0;ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;:INPUT:SCALING:VT:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000;:INPUT:SCALING:CT:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000;:INPUT:SCALING:SFACTOR:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000`

[:INPut]:SCALing:{VT|CT|SFACtor}?

Function Queries the {VT ratio|CT ratio|power factor} of all elements.

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}?`

Example `:INPUT:SCALING:VT? -> :INPUT:SCALING:VT:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000`

[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL]

Function Collectively sets the {VT ratio|CT ratio|power factor} of all elements.

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL] {<NRf>}`
<NRf> = 0.0001 to 99999.9999

Example `:INPUT:SCALING:VT:ALL 1`

[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x>

Function Sets the {VT ratio|CT ratio|power factor} of the element or queries the current setting.

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`

[:INPut]:SCALing:STATe?

Function Queries the scaling ON/OFF states of all elements.

Syntax `[:INPut]:SCALing:STATe?`

Example `:INPUT:SCALING:STATE? -> :INPUT:SCALING:STATE:ELEMENT1 0;ELEMENT2 0;ELEMENT3 0;ELEMENT4 0`

[:INPut]:SCALing[:STATe][:ALL]

Function Collectively turns ON/OFF the scaling of all elements.

Syntax `[:INPut]:SCALing[:STATe][:ALL] {<Boolean>}`

Example `:INPUT:SCALING:STATE:ALL OFF`

[:INPut]:SCALing[:STATe]:ELEMent<x>

Function Turns ON/OFF the scaling of the element or queries the current setting.

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
```

[:INPut]:SYNChronize?

Function Queries the synchronization source of all elements.

Syntax `[:INPut]:SYNChronize?`

Example
```
INPUT:SYNCHRONIZE? ->
:INPUT:SYNCHRONIZE:ELEMENT1 I1;
ELEMENT2 I2;ELEMENT3 I3;ELEMENT4 I4
```

[:INPut]:SYNChronize[:ALL]

Function Collectively sets the synchronization source of all elements.

Syntax `[:INPut]:SYNChronize[:ALL] {U<x>|I<x>|EXTernal|NONE}`
<x> = 1 to 4 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example `:INPUT:SYNCHRONIZE:ALL I1`

[:INPut]:SYNChronize:ELEMent<x>

Function Sets the synchronization source of the element or queries the current setting.

Syntax `[:INPut]:SYNChronize:ELEMent<x> {U<x>|I<x>|EXTernal|NONE}`
`[:INPut]:SYNChronize:ELEMent<x>?`
<x> = 1 to 4 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example
```
:INPUT:SYNCHRONIZE:ELEMENT1 I1
:INPUT:SYNCHRONIZE:ELEMENT1? ->
:INPUT:SYNCHRONIZE:ELEMENT1 I1
```

[:INPut]:SYNChronize:{SIGMA|SIGMB}

Function Collectively sets the synchronization source of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[:INPut]:SYNChronize:{SIGMA|SIGMB} {U<x>|I<x>|EXTernal|NONE}`

Example `:INPUT:SYNCHRONIZE:SIGMA I1`

Description
- `[:INPut]:SYNChronize:SIGMA` is valid only on models with 2 to 4 elements.
- `[:INPut]:SYNChronize:SIGMB` is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting (`[:INPut]:WIRing`).

[:INPut]:VOLTage?

Function Queries all settings related to the voltage measurement.

Syntax `[:INPut]:VOLTage?`

Example
```
:INPUT:VOLTAGE? -> :INPUT:VOLTAGE:
RANGE:ELEMENT1 1.000E+03;
ELEMENT2 1.000E+03;
ELEMENT3 1.000E+03;
ELEMENT4 1.000E+03;:INPUT:VOLTAGE:
AUTO:ELEMENT1 0;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;:INPUT:
VOLTAGE:MODE:ELEMENT1 RMS;
ELEMENT2 RMS;ELEMENT3 RMS;
ELEMENT4 RMS
```

[:INPut]:VOLTage:AUTO?

Function Queries the voltage auto range setting (ON/OFF) of all elements.

Syntax `[:INPut]:VOLTage:AUTO?`

Example
```
:INPUT:VOLTAGE:AUTO? ->
:INPUT:VOLTAGE:AUTO:ELEMENT1 0;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0
```

[:INPut]:VOLTage:AUTO[:ALL]

Function Collectively turns ON/OFF the voltage auto range of all elements.

Syntax `[:INPut]:VOLTage:AUTO[:ALL] {<Boolean>}`

Example `:INPUT:VOLTAGE:AUTO:ALL ON`

[:INPut]:VOLTage:AUTO:ELEMent<x>

Function Turns ON/OFF the voltage auto range of the element or queries the current setting.

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
```

[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB}

Function Collectively turns ON/OFF the voltage auto range of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB} {<Boolean>}`

Example `:INPUT:VOLTAGE:AUTO:SIGMA ON`

Description
- `[:INPut]:VOLTage:AUTO:SIGMA` is valid only on models with 2 to 4 elements.
- `[:INPut]:VOLTage:AUTO:SIGMB` is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting (`[:INPut]:WIRing`).

[:INPut]:VOLTage:MODE?

Function Queries the voltage mode of all elements.

Syntax `[:INPut]:VOLTage:MODE?`

Example `:INPUT:VOLTAGE:MODE? -> :INPUT:VOLTAGE:MODE:ELEMENT1 RMS; ELEMENT2 RMS;ELEMENT3 RMS; ELEMENT4 RMS`

[:INPut]:VOLTage:MODE[:ALL]

Function Collectively sets the voltage mode of all elements.

Syntax `[:INPut]:VOLTage:MODE[:ALL] {RMS|MEAN|DC|RMEAN}`

Example `:INPUT:VOLTAGE:MODE:ALL RMS`

[:INPut]:VOLTage:MODE:ELEMent<x>

Function Sets the voltage mode of the element or queries the current setting.

Syntax `[:INPut]:VOLTage:MODE:ELEMent<x> {RMS|MEAN|DC|RMEAN}`
`[:INPut]:VOLTage:MODE:ELEMent<x>?`
<x> = 1 to 4 (element)

Example `:INPUT:VOLTAGE:MODE:ELEMENT1 RMS`
`:INPUT:VOLTAGE:MODE:ELEMENT1? -> :INPUT:VOLTAGE:MODE:ELEMENT1 RMS`

[:INPut]:VOLTage:MODE:{SIGMA|SIGMB}

Function Collectively sets the voltage mode of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[:INPut]:VOLTage:MODE:{SIGMA|SIGMB} {RMS|MEAN|DC|RMEAN}`

Example `:INPUT:VOLTAGE:MODE:SIGMA RMS`

Description
- `[:INPut]:VOLTage:MODE:SIGMA` is valid only on models with 2 to 4 elements.
- `[:INPut]:VOLTage:MODE:SIGMB` is valid only on models with 2 to 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting (`[:INPut]:WIRing`).

[:INPut]:VOLTage:RANGe?

Function Queries the voltage ranges of all elements.

Syntax `[:INPut]:VOLTage:RANGe?`

Example `:INPUT:VOLTAGE:RANGE? -> :INPUT:VOLTAGE:RANGE: ELEMENT1 1.000E+03; ELEMENT2 1.000E+03; ELEMENT3 1.000E+03; ELEMENT4 1.000E+03`

[:INPut]:VOLTage:RANGe[:ALL]

Function Collectively sets the voltage range of all elements.

Syntax `[:INPut]:VOLTage:RANGe[:ALL] {<Voltage>}`
- When the crest factor is set to 3
  <Voltage> = 15, 30, 60, 100, 150, 300, 600, or 1000 (V)
- When the crest factor is set to 6
  <Voltage> = 7.5, 15, 30, 50, 75, 150, 300, or 500 (V)

Example `:INPUT:VOLTAGE:RANGE:ALL 1000V`

[:INPut]:VOLTage:RANGe:ELEMent<x>

Function Sets the voltage range of the element or queries the current setting.

Syntax `[:INPut]:VOLTage:RANGe:ELEMent<x> {<Voltage>}`
`[:INPut]:VOLTage:RANGe:ELEMent<x>?`
<x> = 1 to 4 (element)
- When the crest factor is set to 3
  <Voltage> = 15, 30, 60, 100, 150, 300, 600, or 1000 (V)
- When the crest factor is set to 6
  <Voltage> = 7.5, 15, 30, 50, 75, 150, 300, or 500 (V)

Example `:INPUT:VOLTAGE:RANGE:ELEMENT1 1000V`
`:INPUT:VOLTAGE:RANGE:ELEMENT1? -> :INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03`

[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB}

Function Collectively sets the voltage range of all elements belonging to wiring unit {ΣA|ΣB}.

Syntax `[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB} {<Voltage>}`
- When the crest factor is set to 3
  <Voltage> = 15, 30, 60, 100, 150, 300, 600, or 1000 (V)
- When the crest factor is set to 6
  <Voltage> = 7.5, 15, 30, 50, 75, 150, 300, or 500 (V)

Example `:INPUT:VOLTAGE:RANGE:SIGMA 1000V`

Description
- `[:INPut]:VOLTage:RANGe:SIGMA` is valid only on models with 2 to 4 elements.
- `[:INPut]:VOLTage:RANGe:SIGMB` is valid only on models with 4 elements.
- This command is invalid, if the wiring unit {SA|SB} does not exist as a result of the wiring system setting (`[:INPut]:WIRing`).

[:INPut]:WIRing

Function Sets the wiring system or queries the current setting.

Syntax

```
[:INPut]:WIRing {(P1W2|P1W3|P3W3|
P3W4|V3A3)[,(P1W2|P1W3|P3W3|P3W4|
V3A3|NONE)][,(P1W2|P1W3|P3W3|NONE)]
[,(P1W2|NONE)]}
[: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 (three-voltage, three-current) [3P3W(3V3A)]
NONE = No wiring

Example

- Example for a 4-element model

```
:INPUT:WIRING P1W2,P1W2,P1W2,P1W2
:INPUT:WIRING? -> :INPUT:
WIRING P1W2,P1W2,P1W2,P1W2

:INPUT:WIRING P1W3,P3W3
:INPUT:WIRING? ->
:INPUT:WIRING P1W3,P3W3
```

- Example for a 3-element model

```
:INPUT:WIRING P3W3,P1W2
:INPUT:WIRING? -> :INPUT:
WIRING P3W3,P1W2

:INPUT:WIRING P3W4
:INPUT:WIRING? ->
:INPUT:WIRING P3W4
```

Description

- Set the wiring system pattern in order starting from the element with the smallest number.
- Some wiring system patterns may not be selectable depending on the model type. For details on the wiring system patterns, see the *User's Manual IM760301-01E*.
- The pattern is fixed to P1W2 on the 1-element model. All other settings are not allowed.

6.15 INTEGrate Group

The commands in this group deal with integration.
You can make the same settings and inquiries as when INTEG on the front panel is used.

:INTEGrate?

Function Queries all settings related to the integration.

Syntax `:INTEGrate?`

Example
```
:INTEGRATE? -> :INTEGRATE:
MODE NORMAL;ACAL 0;TIMER 1,0,0
```

:INTEGrate:ACAL

Function Turns ON/OFF the auto calibration or queries the current setting.

Syntax
```
:INTEGrate:ACAL {<Boolean>}
:INTEGrate:ACAL?
```

Example
```
:INTEGRATE:ACAL OFF
:INTEGRATE:ACAL? ->
:INTEGRATE:ACAL 0
```

:INTEGrate:MODE

Function Sets the integration mode or queries the current setting.

Syntax
```
:INTEGrate:MODE {NORMal|CONTinuous|
RNORmal|RCONtinuous}
:INTEGrate:MODE?
```

NORMal = Normal integration mode
CONTinuous = Continuous integration mode
RNORmal = Real-time normal integration mode
RCONtinuous = Real-time continuous integration mode

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

:INTEGrate:RESet

Function Resets the integrated value.

Syntax `:INTEGrate:RESet`

Example `:INTEGRATE:RESET`

:INTEGrate:RTIMe?

Function Queries the integration start and stop times for real-time integration mode.

Syntax `:INTEGrate:RTIMe<x>?`

Example
```
:INTEGRATE:RTIME? ->
:INTEGRATE:RTIME:
START 2005,1,1,0,0,0;
END 2005,1,1,1,0,0
```

:INTEGrate:RTIMe:{STARt|END}

Function Sets the integration {start|stop} time for real-time integration mode or queries the current setting.

Syntax
```
:INTEGrate:RTIMe:{STARt|
END} {<NRf>,<NRf>,<NRf>,<NRf>,
<NRf>,<NRf>}
:INTEGrate:RTIMe:{STARt|END}?
```

{<NRf>, <NRf>, <NRf>, <NRf>, <NRf>, <NRf>} = 2001, 1, 1, 0, 0, 0 to 2099, 12, 31, 23, 59, 59

1st <NRf> = 2001 to 2099	(year)
2nd <NRf> = 1 to 12	(month)
3rd <NRf> = 1 to 31	(day)
4th <NRf> = 0 to 23	(hour)
5th <NRf> = 0 to 59	(minute)
6th <NRf> = 0 to 59	(second)

Example
```
:INTEGRATE:RTIME:
START 2005,1,1,0,0,0
:INTEGRATE:RTIME:START? ->
:INTEGRATE:RTIME:
START 2005,1,1,0,0,0
```

:INTEGrate:STARt

Function Starts integration.

Syntax `:INTEGrate:STARt`

Example `:INTEGRATE:START`

:INTEGrate:STATe?

Function Queries the integration condition.

Syntax `:INTEGrate:STATe?`

Example `:INTEGRATE:STATE? -> RESET`

Description The response information is as follows:
RESet = Integration reset
READy = Waiting (real-time integration mode)
STARt = Integration in progress
STOP = Integration stop
ERRor = Abnormal integration termination (integration overflow, power failure)
TIMeup = Integration stop due to integration timer time

:INTEGrate:STOP

Function Stops integration.

Syntax `:INTEGrate:STOP`

Example `:INTEGRATE:STOP`

:INTEGrate:TIMer<x>

Function Sets the integration timer time or queries the current setting.

```
Syntax    :INTEGrate:TIMer {<NRf>,<NRf>,
          <NRf>}
          :INTEGrate:TIMer?
```

{<NRf>, <NRf>, <NRf>} = 0, 0, 0 to 10000, 0, 0

1st <NRf> = 0 to 10000	(hour)
2nd <NRf> = 0 to 59	(minute)
3rd <NRf> = 0 to 59	(second)

```
Example   :INTEGRATE:TIMER 1,0,0
          :INTEGRATE:TIMER? ->
          :INTEGRATE:TIMER 1,0,0
```

6.16 MEASure Group

The commands in this group deal with computation.
You can make the same settings and inquiries as when MEASURE, AVG, “Frequency Meas. Item” menu of ITEM, and “η Formula,” “Compensation,” and “Δ Measure” menus of WIRING on the front panel are used.

:MEASure?

Function Queries all settings related to the computation.

Syntax `:MEASure?`

Example
```
:MEASURE? -> :MEASURE:AVERAGING:
STATE 0;TYPE EXPONENT;COUNT 2;:
MEASURE:FREQUENCY:ITEM1 U1;
ITEM2 I1;:MEASURE:SAMPLING AUTO;
SQFORMULA TYPE1;PC:IEC 1976;
P1 0.5000;P2 0.5000;:MEASURE:
EFFICIENCY:ETA1 PB,PA;ETA2 PA,PB;
ETA3 OFF;ETA4 OFF;UDEF1 P1;
UDEF2 P1;:MEASURE:FUNCTION1:
STATE 0;EXPRESSION "UMN(E1)";
UNIT "V";:MEASURE:FUNCTION2:
STATE 0;EXPRESSION "UMN(E2)";
UNIT "V";:MEASURE:FUNCTION3:
STATE 0;EXPRESSION "UMN(E3)";
UNIT "V";:MEASURE:FUNCTION4:
STATE 0;EXPRESSION "UMN(E4)";
UNIT "V";:MEASURE:FUNCTION5:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION6:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION7:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION8:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION9:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION10:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION11:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION12:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION13:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION14:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION15:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION16:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION17:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION18:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:FUNCTION19:
STATE 0;EXPRESSION "U(E1,ORT)";
UNIT "V";:MEASURE:FUNCTION20:
STATE 0;EXPRESSION "I(E1,ORT)";
UNIT "A";:MEASURE:PHASE 180;
SYNCHRONIZE MASTER;MHOLD 0;
COMPENSATION:WIRING:ELEMENT1 OFF;
ELEMENT2 OFF;ELEMENT3 OFF;
ELEMENT4 OFF;:MEASURE:COMPENSATION:
EFFICIENCY 0
```

:MEASure:AVERaging?

Function Queries all settings related to averaging.

Syntax `:MEASure:AVERaging?`

Example
```
:MEASURE:AVERAGING? ->
:MEASURE:AVERAGING:STATE 0;
TYPE EXPONENT;COUNT 2
```

:MEASure:AVERaging:COUNt

Function Sets the averaging coefficient or queries the current setting.

Syntax
```
:MEASure:AVERaging:COUNt {<NRf>}
:MEASure:AVERaging:COUNt?
```
<NRf> = 2, 4, 8, 16, 32, 64 (attenuation constant when TYPE = EXPonent)
<NRf> = 8, 16, 32, 64, 128, or 256 (moving average count when TYPE = LINear)

Example
```
:MEASURE:AVERAGING:COUNT 2
:MEASURE:AVERAGING:COUNT? ->
:MEASURE:AVERAGING:COUNT 2
```

Description The averaging of measurement functions of harmonic measurement (option) is valid only when TYPE = EXPonent (attenuation constant). For details, see the *User’s Manual IM760301-01E*.

:MEASure:AVERaging[:STATe]

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

Syntax
```
:MEASure:AVERaging
[:STATe] {<Boolean>}
:MEASure:AVERaging:STATe?
```

Example
```
:MEASURE:AVERAGING:STATE ON
:MEASURE:AVERAGING:STATE? ->
:MEASURE:AVERAGING:STATE 1
```

:MEASure:AVERaging:TYPE

Function Sets the averaging type or queries the current setting.

Syntax
```
:MEASure:AVERaging:TYPE {EXPonent|
LINear}
:MEASure:AVERaging:TYPE?
```

Example
```
:MEASURE:AVERAGING:TYPE EXPONENT
:MEASURE:AVERAGING:TYPE? ->
:MEASURE:AVERAGING:TYPE EXPONENT
```

Description The averaging of measurement functions of harmonic measurement (option) is valid only for “EXPonent.” For details, see the *User’s Manual IM760301-01E*.

:MEASure:COMPensation?

Function Queries all settings related to the compensation computation.

Syntax
```
:MEASure:COMPensation?
```

Example
```
:MEASURE:COMPENSATION? ->
:MEASURE:COMPENSATION:WIRING:
ELEMENT1 OFF;ELEMENT2 OFF;
ELEMENT3 OFF;ELEMENT4 OFF;:MEASURE:
COMPENSATION:EFFICIENCY 0
```

:MEASure:COMPensation:EFFiciency

Function Turns ON/OFF the efficiency compensation or queries the current setting.

Syntax
```
:MEASure:COMPensation:
EFFiciency {<Boolean>}
```

Example
```
:MEASURE:COMPENSATION:EFFICIENCY ON
:MEASURE:COMPENSATION:EFFICIENCY? -
> :MEASURE:COMPENSATION:
EFFICIENCY 1
```

:MEASure:COMPensation:V3A3

Function Turns ON/OFF the compensation for the two-wattmeter method or queries the current setting.

Syntax
```
:MEASure:COMPensation:
V3A3 {<Boolean>}
```

Example
```
:MEASURE:COMPENSATION:V3A3 ON
:MEASURE:COMPENSATION:V3A3? ->
:MEASURE:COMPENSATION:V3A3 1
```

Description
- This command is valid only on models with 3 to 4 elements.
- This command is valid only on models with the delta computation function (/DT option).
- This command is valid when the wiring system (`[:INPut]:WIRing`) is set to V3A3.

:MEASure:COMPensation:WIRing?

Function Queries all settings related to the wiring compensation.

Syntax
```
:MEASure:COMPensation:WIRing?
```

Example
```
:MEASURE:COMPENSATION:WIRING? ->
:MEASURE:COMPENSATION:WIRING:
ELEMENT1 OFF;ELEMENT2 OFF;
ELEMENT3 OFF;ELEMENT4 OFF
```

:MEASure:COMPensation:WIRing:ELEMent<x>

Function Sets the wiring compensation of the element or queries the current setting.

Syntax
```
:MEASure:COMPensation:WIRing:
ELEMent<x> {OFF|U_I|I_U}
:MEASure:COMPensation:WIRing:
ELEMent<x>?
```
<x> = 1 to 4 (element)

Example
```
:MEASURE:COMPENSATION:WIRING:
ELEMENT1 OFF
:MEASURE:COMPENSATION:WIRING:
ELEMENT1? -> :MEASURE:COMPENSATION:
WIRING:ELEMENT1 OFF
```

:MEASure:DMeasure?

Function Queries all settings related to the delta computation.

Syntax
```
:MEASure:DMeasure?
```

Example
```
:MEASURE:DMEASURE? ->
:MEASURE:DMEASURE:SIGMA OFF;
SIGMB OFF
```

Description This command is valid only on models with the delta computation function (/DT option).

:MEASure:DMeasure[:SIGMA]

Syntax Sets the delta computation mode for wiring unit ΣA or queries the current setting.

Syntax
```
:MEASure:DMeasure[:SIGMA] {OFF|
DIFFerence|P3W3_V3A3|ST_DT|DT_ST}
:MEASure:DMeasure:SIGMA?
```

Example
```
:MEASURE:DMEASURE:SIGMA OFF
:MEASURE:DMEASURE:SIGMA? ->
:MEASURE:DMEASURE:SIGMA OFF
```

Description
- This command is valid only on models with the delta computation function (/DT option).
- The selections are as shown below: The wiring system of wiring unit ΣA determines the selectable modes.
  OFF = No delta computation (single-phase, two-wire system (1P2W) only)
  DIFFerence = Differential voltage, differential current (single-phase, three-wire system (1P3W), three-phase, three-wire system (3P3W) only)
  P3W3_V3A3 = 3P3W-to-3V3A conversion (single-phase, three-wire system (1P3W), three-phase, three-wire system (3P3W) only)
  ST_DT = Star-to-Delta conversion (three-phase, four-wire system (3P4W) only)
  DT_ST = Delta->Star conversion (three-phase, three-wire (three-voltage, three current) system [3P3W (3V3A)] only)

:MEASure:DMeasure:SIGMB

Syntax Sets the delta computation mode for wiring unit ΣB or queries the current setting.

Syntax
```
:MEASure:DMeasure:SIGMB {OFF|
DIFFerence|P3W3_V3A3|ST_DT|DT_ST}
:MEASure:DMeasure:SIGMB?
```

Example
```
:MEASURE:DMEASURE:SIGMB OFF
:MEASURE:DMEASURE:SIGMB? ->
:MEASURE:DMEASURE:SIGMB OFF
```

Description
- This command is valid only on 4-element models with the delta computation function (/DT option).
- The selections are the same as the ":MEASure:DMeasure[:SIGMA]" command. The wiring system of wiring unit ΣB determines the selectable modes.

:MEASure:EFFiciency?

Function Queries all settings related to the efficiency computation.

Syntax
```
:MEASure:EFFiciency?
```

Example
```
:MEASURE:EFFICIENCY? ->
:MEASURE:EFFICIENCY:ETA1 PB,PA;
ETA2 PA,PB;ETA3 OFF;ETA4 OFF;
UDEF1 P1;UDEF2 P1
```

:MEASure:EFFiciency:ETA<x>

Function Sets the efficiency equation or queries the current setting.

Syntax
```
:MEASure:EFFiciency:ETA<x> {(OFF|
P<x>|PA|PB|PM|UDEF<x>)[,(P<x>|PA|
PB|PM|UDEF<x>)]}
:MEASure:EFFiciency:ETA<x>?
```
<x> of ETA<x> = 1 to 4 (η1 to η4)
OFF = No computation (the denominator is ignored)
<x> of P<x> = 1 to 4 (element)
PA = PΣA (only on models with 2 to 4 elements)
PB = PΣB (only on models with 4 elements)
PM = Pm (motor output, motor version (-MV) only)
<x> of UDEF<x> = 1 to 2 (Udef1 to Udef2)

Example
```
:MEASURE:EFFICIENCY:ETA1 PB,PA
:MEASURE:EFFICIENCY:ETA1? ->
:MEASURE:EFFICIENCY:ETA1 PB,PA
```

Description
- Set the numerator and then the denominator.
- The numerator can be omitted. The numerator is set to 1 when omitted.
- The numerator is omitted when the numerator is 1 in the response to a query.

:MEASure:EFFiciency:UDEF<x>

Function Sets the user-defined parameter used in the efficiency equation or queries the current setting.

Syntax
```
:MEASure:EFFiciency:UDEF<x> {(NONE|
P<x>|PA|PB|PM)[,(NONE|P<x>|PA|PB|
PM)][,(NONE|P<x>|PA|PB|PM)][,(NONE|
P<x>|PA|PB|PM)]}
:MEASure:EFFiciency:UDEF<x>?
```
<x> of UDEF<x> = 1 to 2 (Udef1 to Udef2)
NONE = No parameters
<x> of P<x> = 1 to 4 (element)
PA = PΣA (only on models with 2 to 4 elements)
PB = PΣB (only on models with 4 elements)
PM = Pm (motor output, motor version (-MV) only)

Example
```
:MEASURE:EFFICIENCY:UDEF1 P1,P2,P3
:MEASURE:EFFICIENCY:UDEF1? ->
:MEASURE:EFFICIENCY:UDEF1 P1,P2,P3
```

Description
- Set the parameters in the order parameter 1, parameter 2, parameter 3, and parameter 4.
- Parameters 2 to 4 can be omitted. Omitted parameters are set to NONE.
- Parameters are omitted when all of the subsequent parameters are NONE for parameters 2 to 4 in the response to a query.

:MEASure:FREQuency?

Function Queries all settings related to frequency measurement.

Syntax `:MEASure:FREQuency?`

Example
```
:MEASURE:FREQUENCY? ->
:MEASURE:FREQUENCY:ITEM1 U1;
ITEM2 I1
```

Description This command is invalid on models with the frequency measurement add-on (/FQ) option, because the frequency can be measured simultaneously on all input elements.

:MEASure:FREQuency:ITEM<x>

Function Sets the frequency measurement item or queries the current setting.

Syntax
```
:MEASure:FREQuency:ITEM<x> {U<x>|
I<x>}
:MEASure:FREQuency:ITEM<x>?
```
<x> of ITEM<x> = 1 or 2 (Freq. 1 or Freq.2)
<x> of U<x>, I<x> = 1 to 4 (element)

Example
```
:MEASURE:FREQUENCY:ITEM1 U1
:MEASURE:FREQUENCY:ITEM1? ->
:MEASURE:FREQUENCY:ITEM1 U1
```

Description This command is invalid on models with the frequency measurement add-on (/FQ) option, because the frequency can be measured simultaneously on all input elements.

:MEASure:FUNCtion<x>?

Function Queries all settings related to user-defined functions.

Syntax `:MEASure:FUNCtion<x>?`
<x> = 1 to 20 (F1 to F20)

Example
```
:MEASURE:FUNCTION1? ->
:MEASURE:FUNCTION1:STATE 1;
EXPRESSION "UMN(E1)";UNIT "V"
```

:MEASure:FUNCtion<x>:EXPRession

Function Sets the equation of the user-defined function or queries the current setting.

Syntax
```
:MEASure:FUNCtion<x>:EXPRession
{<String>}
:MEASure:FUNCtion<x>:EXPRession?
```
<x> = 1 to 20 (F1 to F20)
<String> = Up to 50 characters

Example
```
:MEASURE:FUNCTION1:
EXPRESSION "UMN(E1)"
:MEASURE:FUNCTION1:EXPRESSION? ->
:MEASURE:FUNCTION1:
EXPRESSION "UMN(E1)"
```

:MEASure:FUNCtion<x>[:STATe]

Function Enables (ON) or Disables (OFF) the user-defined function or queries the current setting.

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

Function Sets the unit to be added to the computation result of the user-defined function or queries the current setting.

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 "V"
:MEASURE:FUNCTION1:UNIT? ->
:MEASURE:FUNCTION1:UNIT "V"
```

Description This command does not affect the computation result.

:MEASure:MHOLd

Function Enables (ON) or Disables (OFF) MAX HOLD function used in the user-defined function or queries the current setting.

Syntax
```
:MEASure:MHOLd {<Boolean>}
:MEASure:MHOLd?
```

Example
```
:MEASURE:MHOLD ON
:MEASURE:MHOLD? -> :MEASURE:MHOLD 1
```

Description
- The MAX HOLD operation starts when the MAX HOLD function is specified by the user-defined function and `:MEASure:MHOLd` is set to ON.
- When `:MEASure:MHOLd` is set to OFF, the MAX HOLD operation terminates, and the MAX HOLD value becomes “no data.”
- If ON is specified while `:MEASure:MHOLd` is ON, the MAX HOLD value is reset once, and the MAX HOLD operation starts again.
- For details on the designation of the MAX HOLD function, see the *User’s Manual IM760301-01E*.

:MEASure:PC?

Function Queries all settings related to the computation of Pc (Corrected Power).

Syntax `:MEASure:PC?`

Example
```
:MEASURE:PC? -> :MEASURE:PC:
IEC 1976;P1 0.5000;P2 0.5000
```

:MEASure:PC:IEC

Function Sets the equation used to compute Pc (Corrected Power) or queries the current setting.

Syntax
```
:MEASure:PC:IEC {<NRf>}
:MEASure:PC:IEC?
```
<NRf> = 1976, 1993

Example
```
:MEASURE:PC:IEC 1976
:MEASURE:PC:IEC? ->
:MEASURE:PC:IEC 1976
```

Description Specify the year when the equation used to calculate the Pc was issued by IEC76-1.

:MEASure:PC:P<x>

Function Sets the parameter used to compute Pc (Corrected Power) or queries the current setting.

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 the ":MEASure:PC:IEC" setting is set to "1976(IEC76-1(1976))."

:MEASure:PHASe

Function Sets the display format of the phase difference or queries the current setting.

Syntax
```
:MEASure:PHASe {<NRf>}
:MEASure:PHASe?
```
<NRf> = 180 or 360

Example
```
:MEASURE:PHASE 180
:MEASURE:PHASE? ->
:MEASURE:PHASE 180
```

Description Displays the phase using ±0 to 180° (Lead/Lag) for "180" and 0 to 360° for "360."

:MEASure:SAMPling

Function Sets the sampling frequency or queries the current setting.

Syntax
```
:MEASure:SAMPling {AUTO|CLKA|CLKB|
CLKC}
:MEASure:SAMPling?
```

Example
```
:MEASURE:SAMPLING AUTO
:MEASURE:SAMPLING? ->
:MEASURE:SAMPLING AUTO
```

Description For details on the sampling frequency corresponding to {AUTO|CLKA|CLKB|CLKC}, see the *User's Manual IM760301-01E*.

:MEASure:SQFormula

Function Sets the equation used to compute S (apparent power) and Q (reactive power) or queries the current setting.

Syntax
```
:MEASure:SQFormula {TYPE1|TYPE2|
TYPE3}
:MEASure:SQFormula?
```

Example
```
:MEASURE:SQFORMULA TYPE1
:MEASURE:SQFORMULA? ->
:MEASURE:SQFORMULA TYPE1
```

Description
- For details on the equation corresponding to {TYPE1|TYPE2|TYPE3}, see the *User's Manual IM760301-01E*.
- "TYPE3" is selectable only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:MEASure:SYNChronize

Function Sets the synchronized measurement mode or queries the current setting.

Syntax
```
:MEASure:SYNChronize {MASTer|SLAVe}
:MEASure:SYNChronize?
```

Example
```
:MEASURE:SYNCHRONIZE MASTER
:MEASURE:SYNCHRONIZE? ->
:MEASURE:SYNCHRONIZE MASTER
```

6.17 MOTor Group

The commands in this group deal with the motor evaluation function.
You can make the same settings and inquiries as when MOTOR SET (SHIFT+SCALING) on the front panel is used. However, the commands in this group are valid only on the motor version (-MV).

:MOTor?

Function Queries all settings related to the motor evaluation function.

Syntax `:MOTor?`

Example
```
:MOTOR? -> :MOTOR:SPEED:
TYPE ANALOG;RANGE 20.0E+00;AUTO 0;
SCALING 1.0000;UNIT "rpm";:MOTOR:
TORQUE:TYPE ANALOG;RANGE 20.0E+00;
AUTO 0;SCALING 1.0000;UNIT "Nm";:
MOTOR:PM:SCALING 1.0000;UNIT "W";:
MOTOR:FILTER:LINE OFF;:MOTOR:
SYNCHRONIZE NONE;POLE 2;SSPEED I1
```

:MOTor:FILTer?

Function Queries all settings related to the input filter.

Syntax `:MOTor:FILTer?`

Example
```
:MOTOR:FILTER? ->
:MOTOR:FILTER:LINE OFF
```

:MOTor:FILTer[:LINE]

Function Sets the line filter or queries the current setting.

Syntax
```
:MOTor:FILTer[:LINE]
{OFF|<Frequency>}
:MOTor:FILTer:LINE?
```
OFF = Line filter OFF
<Frequency> = 100 Hz, 50 kHz (line filter ON, cutoff frequency)

Example
```
:MOTOR:FILTER:LINE OFF
:MOTOR:FILTER:LINE? ->
:MOTOR:FILTER:LINE OFF
```

:MOTor:PM?

Function Queries all settings related to the motor output (Pm).

Syntax `:MOTor:PM?`

Example
```
:MOTOR:PM? -> :MOTOR:PM:
SCALING 1.0000;UNIT "W"
```

:MOTor:PM:SCALing

Function Sets the scaling factor used for motor output computation or queries the current setting.

Syntax
```
:MOTor:PM:SCALing {<NRf>}
:MOTor:PM:SCALing?
```
<NRf> = 0.0001 to 99999.9999

Example
```
:MOTOR:PM:SCALING 1
:MOTOR:PM:SCALING? ->
:MOTOR:PM:SCALING 1.0000
```

:MOTor:PM:UNIT

Function Sets the unit to add to the motor output computation result or queries the current setting.

Syntax
```
:MOTor:PM:UNIT {<String>}
:MOTor:PM:UNIT?
```
<String> = Up to 8 characters

Example
```
:MOTOR:PM:UNIT "W"
:MOTOR:PM:UNIT? ->
:MOTOR:PM:UNIT "W"
```

Description This command does not affect the computation result.

:MOTor:POLE

Function Sets the motor's number of poles or queries the current setting.

Syntax
```
:MOTor:POLE {<NRf>}
:MOTor:POLE?
```
<NRf> = 1 to 99

Example
```
:MOTOR:POLE 2
:MOTOR:POLE? -> :MOTOR:POLE 2
```

:MOTor:SPEed?

Function Queries all settings related to the rotating speed.

Syntax `:MOTor:SPEed?`

Example
```
:MOTOR:SPEED? -> :MOTOR:SPEED:
TYPE ANALOG;RANGE 20.0E+00;AUTO 0;
SCALING 1.0000;UNIT "rpm"
```

:MOTor:SPEed:AUTO

Function Turns ON/OFF the voltage auto range of the revolution signal input (analog input format) or queries the current setting.

Syntax
```
:MOTor:SPEed:AUTO {<Boolean>}
:MOTor:SPEed:AUTO?
```

Example
```
:MOTOR:SPEED:AUTO ON
:MOTOR:SPEED:AUTO? ->
:MOTOR:SPEED:AUTO 1
```

Description This command is valid when the revolution signal input type (`:MOTor:SPEed:TYPE`) is "ANALog (analog input)."

:MOTor:SPEed:PRANge

Function Sets the range of the rotating speed (pulse input format) or queries the current setting.

Syntax
```
:MOTor:SPEed:PRANge {<NRf>,<NRf>}
:MOTor:SPEed:PRANge?
```
<NRf> = 0.0000 to 99999.9999

Example
```
:MOTOR:SPEED:PRANGE 10000,0
:MOTOR:SPEED:PRANGE? ->
:MOTOR:SPEED:
PRANGE 10000.0000,0.0000
```

Description
- Set the upper limit and then the lower limit.
- This command is valid when the revolution signal input type (`:MOTor:SPEed:TYPE`) is “PULSe (pulse input).”

:MOTor:SPEed:PULSe

Function Sets the pulse count of the revolution signal input (pulse input) or queries the current setting.

Syntax
```
:MOTor:SPEed:PULSe {<NRf>}
:MOTor:SPEed:PULSe?
```
<NRf> = 1 to 9999

Example
```
:MOTOR:SPEED:PULSE 60
:MOTOR:SPEED:PULSE? ->
:MOTOR:SPEED:PULSE 60
```

Description This command is valid when the revolution signal input type (`:MOTor:SPEed:TYPE`) is “PULSe (pulse input).”

:MOTor:SPEed:RANGe

Function Sets the voltage range of the revolution signal input (analog input format) or queries the current setting.

Syntax
```
:MOTor:SPEed:RANGe {<Voltage>}
:MOTor:SPEed:RANGe?
```
<voltage> = 1, 2, 5, 10, or 20 (V)

Example
```
:MOTOR:SPEED:RANGE 20V
:MOTOR:SPEED:RANGE? ->
:MOTOR:SPEED:RANGE 20.0E+00
```

Description This command is valid when the revolution signal input type (`:MOTor:SPEed:TYPE`) is “ANALog (analog input).”

:MOTor:SPEed:SCALing

Function Sets the scaling factor for rotating speed computation or queries the current setting.

Syntax
```
:MOTor:SPEed:SCALing {<NRf>}
:MOTor:SPEed:SCALing?
```
<NRf> = 0.0001 to 99999.9999

Example
```
:MOTOR:SPEED:SCALING 1
:MOTOR:SPEED:SCALING? ->
:MOTOR:SPEED:SCALING 1.0000
```

:MOTor:SPEed:TYPE

Function Sets the input type of the revolution signal input or queries the current setting.

Syntax
```
:MOTor:SPEed:TYPE {ANALog|PULSe}
:MOTor:SPEed:TYPE?
```

Example
```
:MOTOR:SPEED:TYPE ANALOG
:MOTOR:SPEED:TYPE? ->
:MOTOR:SPEED:TYPE ANALOG
```

:MOTor:SPEed:UNIT

Function Sets the unit to add to the rotating speed computation result or queries the current setting.

Syntax
```
:MOTor:SPEed:UNIT {<String>}
:MOTor:SPEed:UNIT?
```
<String> = Up to 8 characters

Example
```
:MOTOR:SPEED:UNIT "rpm"
:MOTOR:SPEED:UNIT? ->
:MOTOR:SPEED:UNIT "rpm"
```

Description This command does not affect the computation result.

:MOTor:SSPeed(Sync SPeed source)

Function Sets the frequency measurement source used to compute the synchronous speed (SyncSp) or queries the current setting.

Syntax
```
:MOTor:SSPeed {U<x>|I<x>}
:MOTor:SSPeed?
```
<x> = 1 to 4 (element)

Example
```
:MOTOR:SSPEED I1
:MOTOR:SSPEED? -> :MOTOR:SSPEED I1
```

:MOTor:SYNChronize

Function Sets the synchronization source used to compute the rotating speed and torque or queries the current setting.

Syntax
```
:MOTor:SYNChronize {U<x>|I<x>|
EXTernal|NONE}
:MOTor:SYNChronize?
```
<x> = 1 to 4 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example
```
:MOTOR:SYNCHRONIZE NONE
:MOTOR:SYNCHRONIZE? ->
:MOTOR:SYNCHRONIZE NONE
```

:MOTor:TORQue?

Function Queries all settings related to the torque.

Syntax
```
:MOTor:TORQue?
```

Example
```
:MOTOR:TORQUE? -> :MOTOR:TORQUE:
TYPE ANALOG;RANGE 20.0E+00;AUTO 0;
SCALING 1.0000;UNIT "Nm"
```

:MOTor:TORQue:AUTO

Function Turns ON/OFF the voltage auto range of the torque signal input (analog input format) or queries the current setting.

Syntax
```
:MOTor:TORQue:AUTO {<Boolean>}
:MOTor:TORQue:AUTO?
```

Example
```
:MOTOR:TORQUE:AUTO ON
:MOTOR:TORQUE:AUTO? ->
:MOTOR:TORQUE:AUTO 1
```

Description This command is valid when the torque signal input type (`:MOTor:TORQue:TYPE`) is “ANALog (analog input).”

:MOTor:TORQue:PRANge

Function Sets the range of the torque (pulse input format) or queries the current setting.

Syntax
```
:MOTor:TORQue:PRANge {<NRf>,<NRf>}
:MOTor:TORQue:PRANge?
```
<NRf> = –10000.0000 to 10000.0000

Example
```
:MOTOR:TORQUE:PRANGE 50,-50
:MOTOR:TORQUE:PRANGE? ->
:MOTOR:TORQUE:
PRANGE 50.0000,-50.0000
```

Description
- Set the upper limit and then the lower limit.
- This command is valid when the torque signal input type (`:MOTor:TORQue:TYPE`) is “PULSe (pulse input).”

:MOTor:TORQue:RANGe

Function Sets the voltage range of the torque signal input (analog input format) or queries the current setting.

Syntax
```
:MOTor:TORQue:RANGe {<Voltage>}
:MOTor:TORQue:RANGe?
```
<voltage> = 1, 2, 5, 10, or 20 (V)

Example
```
:MOTOR:TORQUE:RANGE 20V
:MOTOR:TORQUE:RANGE? ->
:MOTOR:TORQUE:RANGE 20.0E+00
```

Description This command is valid when the torque signal input type (`:MOTor:TORQue:TYPE`) is “ANALog (analog input).”

:MOTor:TORQue:RATE?

Function Queries all settings related to the rated value of the torque signal (pulse input format).

Syntax
```
:MOTor:TORQue:RATE?
```

Example
```
:MOTOR:TORQUE:RATE? ->
:MOTOR:TORQUE:RATE:
UPPER 50.0000,15.000E+03;
LOWER -50.0000,5.000E+03
```

:MOTor:TORQue:RATE:{UPPer|LOWer}

Function Sets the rated value {upper limit|lower limit} of the torque signal (pulse input format) or queries the current setting.

Syntax
```
:MOTor:TORQue:RATE:{UPPer|
LOWer} {<NRf>,<Frequency>}
```
<NRf> = –10000.0000 to 10000.0000
<Frequency> = 1 Hz to 100 MHz

Example
```
:MOTOR:TORQUE:RATE:UPPER 50,15kHz
:MOTOR:TORQUE:RATE:UPPER? ->
:MOTOR:TORQUE:RATE:UPPER
50.0000,15.000E+03
```

Description This command is valid when the torque signal input type (`:MOTor:TORQue:TYPE`) is “PULSe (pulse input).”

:MOTor:TORQue:SCALing

Function Sets the scaling factor used for torque computation or queries the current setting.

Syntax
```
:MOTor:TORQue:SCALing {<NRf>}
:MOTor:TORQue:SCALing?
```
<NRf> = 0.0001 to 99999.9999

Example
```
:MOTOR:TORQUE:SCALING 1
:MOTOR:TORQUE:SCALING? ->
:MOTOR:TORQUE:SCALING 1.0000
```

:MOTor:TORQue:TYPE

Function Sets the input type of the torque signal input or queries the current setting.

Syntax
```
:MOTor:TORQue:TYPE {ANALog|PULSe}
:MOTor:TORQue:TYPE?
```

Example
```
:MOTOR:TORQUE:TYPE ANALOG
:MOTOR:TORQUE:TYPE? ->
:MOTOR:TORQUE:TYPE ANALOG
```

:MOTor:TORQue:UNIT

Function Sets the unit to add to the torque computation result or queries the current setting.

Syntax
```
:MOTor:TORQue:UNIT {<String>}
:MOTor:TORQue:UNIT?
```
<String> = Up to 8 characters

Example
```
:MOTOR:TORQUE:UNIT "Nm"
:MOTOR:TORQUE:UNIT? ->
:MOTOR:TORQUE:UNIT "Nm"
```

Description This command does not affect the computation result.

6.18 NUMeric Group

The commands in this group deal with numeric data output.
There are no front panel keys that correspond to the commands in this group. The NUMERIC key on the front panel can be used to specify the same settings. The DISPlay group commands can be used to query the settings.

:NUMeric?

Function Queries all settings related to the numeric data output.

Syntax `:NUMeric?`

Example
```
:NUMERIC? -> :NUMERIC:FORMAT ASCII;
NORMAL:NUMBER 15;ITEM1 U,1,TOTAL;
ITEM2 I,1,TOTAL;ITEM3 P,1,TOTAL;
ITEM4 S,1,TOTAL;ITEM5 Q,1,TOTAL;
ITEM6 LAMBDA,1,TOTAL;
ITEM7 PHI,1,TOTAL;ITEM8 FU,1;
ITEM9 FI,1;ITEM10 UPPEAK,1;
ITEM11 UMPEAK,1;ITEM12 IPPEAK,1;
ITEM13 IMPEAK,1;ITEM14 CFU,1;
ITEM15 CFI,1;:NUMERIC:HOLD 0
```

:NUMeric:CBCycle?

Function Queries all settings related to output of numeric list data of Cycle by Cycle measurement.

Syntax `:NUMeric:CBCycle?`

Example
```
:NUMERIC:CBCYCLE? ->
:NUMERIC:CBCYCLE:ITEM U,1;
START 1;END 100
```

Description Only available with the Cycle by Cycle measurement function (/CC option).

:NUMeric:CBCycle:END

Function Sets the output end cycle of the numeric list data output by :NUMeric:CBCycle:VALue? or queries the current setting.

Syntax
```
:NUMeric:CBCycle:END {<NRf>}
:NUMeric:CBCycle:END?
```
<NRf> = 1 to 3000 (cycle number)

Example
```
:NUMERIC:CBCYCLE:END 100
:NUMERIC:CBCYCLE:END ->
:NUMERIC:CBCYCLE:END 100
```

Description Only available with the Cycle by Cycle measurement function (/CC option).

:NUMeric:CBCycle:ITEM

Function Sets the numeric list data output items (function and element) of Cycle by Cycle measurement or queries the current setting.

Syntax
```
:NUMeric:CBCycle:ITEM {<Function>,
<Element>}
:NUMeric:CBCycle:ITEM?
```
<Function> = {FREQ|U|I|P|S|Q|LAMBda|SPEed|TORQue|PM|PKU|PKI|PKSPeed|PKTorque}
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)

Example
```
:NUMERIC:CBCYCLE:ITEM U,1
:NUMERIC:CBCYCLE:ITEM? ->
:NUMERIC:CBCYCLE:ITEM U,1
```

Description
- Only available with the Cycle by Cycle function (/CC option).
- When <Function> = {FREQ|SPEed|TORQue|PM|PKSPeed|PKTorque}, <Element> need not be specified. <Element> is omitted from the response.
- When <Element> is omitted, Element 1 is set.
- {SPEed|TORQue|PM|PKSPeed|PKTorque} is only available with the motor version (-MV).

:NUMeric:CBCycle:STARt

Function Sets the output start cycle of the numeric list data output by :NUMeric:CBCycle:VALue? or queries the current setting.

Syntax
```
:NUMeric:CBCycle:STARt {<NRf>}
:NUMeric:CBCycle:STARt?
```
<NRf> = 1 to 3000 (cycle number)

Example
```
:NUMERIC:CBCYCLE:START 1
:NUMERIC:CBCYCLE:START ->
:NUMERIC:CBCYCLE:START 1
```

Description Only available with the Cycle by Cycle measurement function (/CC option).

:NUMeric:CBCycle:VALue?

Function: Queries the numeric list data from Cycle by Cycle measurement.

Syntax:

```
:NUMeric:CBCycle:VALue?
{<Function>,<Element>}
:NUMeric:CBCycle:VALue?
```

<Function> = {FREQ|U|I|P|S|Q|LAMBda|SPEed|TORQue|PM|PKU|PKI|PKSPeed|PKTorque}

<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)

Example:

- Example when parameters specified (For :NUMeric:CBCycle:STARt 1;END 100)
  ```
  :NUMERIC:CBCYCLE:VALUE? U,1 ->
  100.001E+00,100.002E+00,
  100.003E+00,100.004E+00,
  ..(ommitted)..,100.099E+00,
  100.100E+00
  ```
- Example when parameters omitted (For :NUMeric:CBCycle:ITEM U,1)
  ```
  :NUMERIC:CBCYCLE:VALUE? ->
  100.001E+00,100.002E+00,
  100.003E+00,100.004E+00,
  ..(ommitted)..,100.099E+00,
  100.100E+00
  ```
- Example in which ":NUMeric:FORMat" is set to FLOat.
  :NUMERIC:CBCYCLE:VALUE? -> #6 (number of bytes, six digits) (data byte sequence)

Description:

- Only available with the Cycle by Cycle function (/CC option).
- Numeric list data consists of up to 3000 numeric data from ":NUMeric:CBCycle:STARt" to ":Numeric:CBCycle:END".
- If a parameters are specified, the numeric list data of that item is output. For a description of the contents of <Function> and <Element>, see the description for ":NUMeric:CBCycle:ITEM".
- If the parameter is omitted, the numeric list data for the items specified in ":NUMeric:CBCycle:ITEM" is output.
- For the format of the individual numeric data that are output, see "Numeric Data Format" at the end of this group (page 6-98).

:NUMeric:FLICker?

Function: Queries all settings related to output of numeric data from flicker measurement.

Syntax: `:NUMeric:FLICker?`

Example:

```
:NUMERIC:FLICKER? -> :NUMERIC:
FLICKER:FUNCTION:NUMBER 8;
ITEM1 TIME;ITEM2 UN,1;ITEM3 FU,1;
ITEM4 DC,1,CURRENT;ITEM5 DMAX,1,
CURRENT;ITEM6 DT,1,CURRENT;
ITEM7 PST,1,CURRENT;ITEM8 PLT,1;:
NUMERIC:FLICKER:INFORMATION:
NUMBER 7;ITEM1 JTOTAL,TOTAL;
ITEM2 JDC,1,ALL;ITEM3 JDMAX,1,ALL;
ITEM4 JDT,1,ALL;ITEM5 JPST,1,ALL;
ITEM6 JPLT,1;ITEM7 JTOTAL,1
```

Description: Only available with the flicker measurement function (/FL option).

:NUMeric:FLICker:COUNt?

Function: Queries the number of the measurement within the specified observation period at which flicker measurement stops.

Syntax: `:NUMeric:FLICker:COUNt?`

Example: `:NUMERIC:FLICKER:COUNT? -> 0`

Description:

- Only available with the flicker measurement function (/FL option).
- Returns the number displayed on the right of the Count bar graph in the flicker measurement display screen.

:NUMeric:FLICker:FUNCtion?

Function: Queries all settings related to output of measured flicker data (variable format).

Syntax: `:NUMeric:FLICker:FUNCtion?`

Example:

```
:NUMERIC:FLICKER:FUNCTION? ->
:NUMERIC:FLICKER:FUNCTION:NUMBER 8;
ITEM1 TIME;ITEM2 UN,1;ITEM3 FU,1;
ITEM4 DC,1,CURRENT;ITEM5 DMAX,1,
CURRENT;ITEM6 DT,1,CURRENT;
ITEM7 PST,1,CURRENT;ITEM8 PLT,1
```

Description:

- Only available with the flicker measurement function (/FL option).
- For the values of ":NUMeric:FLICker:FUNCtion:ITEM<x>", only the number of numeric data output items specified in ":NUMeric:FLICker:FUNCtion:NUMber" are output.

:NUMeric:FLICker:FUNCtion:CLEar

Function: Clears (sets to NONE) the output items of measured flicker data (variable format).

Syntax: `:NUMeric:FLICker:FUNCtion:CLEar {ALL|<NRf>[,<NRf>]}`

ALL = Clears all items
1st <NRf> = 1 to 32 (item number to start clearing)
2nd <NRf> = 1 to 32 (item number to stop clearing)

Example: `:NUMERIC:FLICKER:FUNCTION:CLEAR ALL`

Description:
- Only available with the flicker measurement function (/FL option).
- If the 2nd <NRf> is omitted, the output items from the start clear number to the last item (32) are cleared.

:NUMeric:FLICker:FUNCtion:DELete

Function: Deletes the output items of measured flicker data (variable format).

Syntax: `:NUMeric:FLICker:FUNCtion:DELete {<NRf>[,<NRf>]}`

1st <NRf> = 1 to 32 (item number to start deleting)
2nd <NRf> = 1 to 32 (item number to stop deleting)

Example: `:NUMERIC:FLICKER:FUNCTION:DELETE 1`
(Deletes ITEM1 and shifts ITEM2 and subsequent items forward)
`:NUMERIC:FLICKER:FUNCTION:DELETE 1,3`
(Deletes ITEM1 to 3 and shifts ITEM4 and subsequent items forward)

Description:
- Only available with the flicker measurement function (/FL option).
- Output items subsequent to the deleted output items are shifted in order into the deleted items' position, and NONE is set in the open positions on the end.
- If the 2nd <NRf> is omitted, only the output item of the start delete number is deleted.

:NUMeric:FLICker:FUNCtion:ITEM<x>

Function: Sets output items (function, element, and observation period) of measured flicker data (variable format) or queries the current setting.

Syntax: `:NUMeric:FLICker:FUNCtion:ITEM<x> {NONE|<Function>,<Element>[,<Period>]}`
`:NUMeric:FLICker:FUNCtion:ITEM<x>?`

<x> = 1 to 32 (item number)
NONE = No output items
<Function> = {TIME|UN|FU|DC|DMAX|DT|PST|PLT}
<Element> = {<NRf>} (<NRf> = 1 to 4)
<Period> = {CURRent|<NRf>|ALL}
(<NRf> = 1 to 99)

Example: `:NUMERIC:FLICKER:FUNCTION:ITEM1 DC,1,1`
`:NUMERIC:FLICKER:FUNCTION:ITEM1? -> :NUMERIC:FLICKER:FUNCTION:ITEM1 DC,1,1`

Description:
- Only available with the flicker measurement function (/FL option).
- The contents that are output for each of the selections for <Function> are as follows:

<Function> Output Contents	<Element> Designation	<Period> Designation
TIME		
Elapsed measurement time [sec] (the time displayed under Flicker: in the upper right part of the screen)	Not required	Not required
UN		
RMS voltage (rated voltage) Un[V]	Required	Not required
FU		
Voltage frequency Freq[Hz]	Required	Not required
DC		
Relative steady-state voltage change dc[%]	Required	Required
DMAX		
Maximum relative voltage change dmax[%]	Required	Required
DT		
Relative voltage change time d(t)[ms]	Required	Required
PST		
Short-term flicker value Pst	Required	Required
PLT		
Long-term flicker value Plt	Required	Not required

- When <Element> is omitted, Element 1 is set.
- The contents of the selections for <Period> are as follows. If <Period> is omitted, CURRent is set.

CURRent: The observation period currently being measured (rows of the numeric list marked with an asterisk (*) in the flicker measurement display screen). When in measurement complete status, same as ALL.

ALL Overall observation period (Result row of the numeric list in the flicker measurement display screen).

<NRf> = 1 to 99 Specified observation period.

- <Element> or <Period> is omitted from the response to the output items in the table above for which specification of <Element> or <Period> is not required.

:NUMeric:FLICker:FUNCtion:NUMber

Function Sets the number of measured flicker data output by ":NUMeric:FLICker:FUNCtion:VALue?" or queries the current setting.

Syntax
```
:NUMeric:FLICker:FUNCtion:
NUMber {<NRf>|ALL}
:NUMeric:FLICker:FUNCtion:NUMber?
```
<NRf> = 1 to 32(ALL)

Example
```
:NUMERIC:FLICKER:FUNCTION:NUMBER 8
:NUMERIC:FLICKER:FUNCTION:NUMBER ->
:NUMERIC:FLICKER:FUNCTION:NUMBER 8
```

Description
- Only available with the flicker measurement function (/FL option).
- If parameters are omitted from ":NUMeric:FLICker:FUNCtion:VALue?", numeric data from 1 to (specified value) is output in order.
- The initial setting for the number of numeric data is 8.

:NUMeric:FLICker:FUNCtion:VALue?

Function Queries the measured flicker data (variable format).

Syntax
```
:NUMeric:FLICker:FUNCtion:
VALue? {<NRf>}
:NUMeric:FLICker:FUNCtion:VALue?
```
<NRf> = 1 to 32 (item number)

Example
- Example when <NRf> is specified
```
:NUMERIC:FLICKER:FUNCTION:
VALUE? 4 -> 1.52E+00
```
- Example when <NRf> is omitted
```
:NUMERIC:FLICKER:FUNCTION:
VALUE? -> 600,229.75E+00,
50.000E+00,1.52E+00,1.56E+00,
3E+00,0.43E+00,0.17E+00
```
- Example in which ":NUMeric:FORMat" is set to "FLOat".
  `:NUMERIC:FLICKER:FUNCTION:VALUE? -> #4` (number of bytes, four digits) (data byte sequence)

Description
- Only available with the flicker measurement function (/FL option).
- When <NRf> is specified, only the numeric data for that item is output.
- If <NRf> is omitted, numeric data from the item number in ":NUMeric:FLICker:FUNCtion:NUMber" is output in order.
- The format of individual numeric data that is output is as follows:
  (1) Data when normal
  - Elapsed measurement time (TIME)
    ASCII: <NR1> format in units of seconds (Example :for 1 hour (1:00:00), 3600)
    FLOAT: IEEE single-precision floating point (4-byte) format in units of seconds (Example :for 1 hour (1:00:00), 0x45610000)
  - No items (NONE)
    ASCII: "NAN" (Not A Number)
    FLOAT: 0x7E951BEE (9.91E+37)
  - Other than above
    ASCII: <NR3> format (mantissa, 5 digits; exponent, 2 digits, Example :229.87E+00)
    FLOAT: IEEE single-precision floating point (4-byte) format

  (2) Error Data
  - Data does not exist (display: "-----")
    ASCII: "NAN" (Not A Number)
    FLOAT: 0x7E951BEE (9.91E+37)
  - Overrange (display: "-O-L-")
  - Overflow (display: "-O-F-")
  - Data over (display: "Error")
  - No steady-state condition (display: "Undef")
    ASCII: "INF" (INFinity)
    FLOAT: 0x7E94F56A (9.9E+37)

:NUMeric:FLICker:INFOrmation?

Function Queries all settings related to output of flicker judgment results (variable format).

Syntax
```
:NUMeric:FLICker:INFOrmation?
```

Example
```
:NUMERIC:FLICKER:INFORMATION? ->
:NUMERIC:FLICKER:INFORMATION:
NUMBER 7;ITEM1 JTOTAL,TOTAL;
ITEM2 JDC,1,ALL;ITEM3 JDMAX,1,ALL;
ITEM4 JDT,1,ALL;ITEM5 JPST,1,ALL;
ITEM6 JPLT,1;ITEM7 JTOTAL,1
```

Description
- Only available with the flicker measurement function (/FL option).
- For the values of ":NUMeric:FLICker:INFOrmation:ITEM<x>", only the number of numeric data output items specified in ":NUMeric:FLICker:FUNCtion:INFOrmation:NUMber" are output.

:NUMeric:FLICker:INFOrmation:CLEar

Function Clears (sets to NONE) the output items of flicker judgment results (variable format).

Syntax :NUMeric:FLICker:INFOrmation:CLEar {ALL|<NRf>[,<NRf>]}
ALL = Clears all items
1st <NRf> = 1 to 32 (item number to start clearing)
2nd <NRf> = 1 to 32 (item number to stop clearing)

Example :NUMERIC:FLICKER:INFORMATION:CLEAR ALL

Description
- Only available with the flicker measurement function (/FL option).
- If the 2nd <NRf> is omitted, the output items from the start clear number to the last item (32) are cleared.

:NUMeric:FLICker:INFOrmation:DELete

Function Deletes the output items of flicker judgment results (variable format).

Syntax :NUMeric:FLICker:INFOrmation:DELete {<NRf>[,<NRf>]}
1st <NRf> = 1 to 32 (item number to start deleting)
2nd <NRf> = 1 to 32 (item number to stop deleting)

Example :NUMERIC:FLICKER:INFORMATION:DELETE 1
(Deletes ITEM1 and shifts ITEM2 and subsequent items forward)
:NUMERIC:FLICKER:INFORMATION:DELETE 1,3
(Deletes ITEM1Å|3 and shifts ITEM4 and subsequent items forward)

Description
- Only available with the flicker measurement function (/FL option).
- Output items subsequent to the deleted output items are shifted in order into the deleted items' position, and NONE is set in the open positions on the end.
- If the 2nd <NRf> is omitted, only the output item of the start delete number is deleted.

:NUMeric:FLICker:INFOrmation:ITEM<x>

Function Sets the output items (function, element, and observation period) of flicker judgment results (variable format) or queries the current setting.

Syntax :NUMeric:FLICker:INFOrmation:ITEM<x> {NONE|<Function>,<Element>[,<Period>]}
:NUMeric:FLICker:INFOrmation:ITEM<x>?
<x> = 1 to 32 (item number)
NONE = No output items
<Function> = {JTOTal|JDC|JDMAX|JDT|JPST|JPLT}
<Element> = {<NRf>|TOTal} (<NRf> = 1 to 4)
<Period> = {<NRf>|ALL} (<NRf> = 1 to 99)

Example :NUMERIC:FLICKER:INFORMATION:ITEM1 JDC,1,1
:NUMERIC:FLICKER:INFORMATION:ITEM1? -> :NUMERIC:FLICKER:INFORMATION:ITEM1 JDC,1,1

Description
- Only available with the flicker measurement function (/FL option).
- The contents that are output for each of the selections for <Function> are as follows:

<Function> Output Contents	<Element> Designation	<Period> Designation
JTOTal Overall judgment results for dc, dmax, d(t), pst, and plt (the judgment results displayed under [Element# Judgment:] in the upper right part of the screen)	Required	Not required
JDC Judgment results for relative steady-state voltage change dc	Required	Required
JDMAX Judgment results for maximum relative voltage change dmax	Required	Required
JDT Judgment results for relative voltage change time d(t)	Required	Required
JPST Judgment results for short-term flicker value Pst	Required	Required
JPLT Judgment results for long-term flicker value Plt	Required	Not required

- The contents of the selections for <Element> are as follows. When <Element> is omitted, Element 1 is set.
  TOTal The overall judgment result for all measured elements is only available when <Function> = JTOTal (judgment result displayed under [Total Judgment:] in the upper right part of the screen)
  <NRf> = 1 to 4 specified elements

- The contents of the selections for <Period> are as follows.
  If <Period> is omitted, ALL is set.
  ALL Overall observation period (Result row of the numeric list in the flicker measurement display screen)
  <NRf> = 1 to 99 specified observation periods
- <Period> is omitted from the response to output items in the table above for which specification of <Period> is not required.

:NUMeric:FLICker:INFOrmation:NUMber

Function Sets the number of flicker judgment results output by "`:NUMeric:FLICker:INFOrmation:VALue?`" or queries the current setting.

Syntax `:NUMeric:FLICker:INFOrmation:NUMber {<NRf>|ALL}`
`:NUMeric:FLICker:INFOrmation:NUMber?`
<NRf> = 1 to 32(ALL)

Example `:NUMERIC:FLICKER:INFORMATION:NUMBER 7`
`:NUMERIC:FLICKER:INFORMATION:NUMBER -> :NUMERIC:FLICKER:FUNCTION:NUMBER 7`

Description
- Only available with the flicker measurement function (/FL option).
- If parameters are omitted from "`:NUMeric:FLICker:INFOrmation:VALue?`", judgment results from 1 to (specified value) are output in order.
- The initial setting for the number of judgment results is 7.

:NUMeric:FLICker:INFOrmation:VALue?

Function Queries the judgment results (variable format).

Syntax `:NUMeric:FLICker:INFOrmation:VALue? {<NRf>}`
`:NUMeric:FLICker:INFOrmation:VALue?`
<NRf> = 1 to 32 (item number)

Example
- Example when <NRf> is specified
  `:NUMERIC:FLICKER:INFORMATION:VALUE? 1 -> 0`
- Example when <NRf> is omitted
  `:NUMERIC:FLICKER:INFORMATION:VALUE? -> 0,0,0,0,0,0,0`
- Example in which "`:NUMeric:FORMat`" is set to "`FLOat`".
  `:NUMERIC:FLICKER:INFORMATION:VALUE? -> #4` (number of bytes, four digits) (data byte sequence)

Description
- Only available with the flicker measurement function (/FL option).
- When <NRf> is specified, only the judgment results for that item is output.
- If <NRf> is omitted, judgment results from the item number in "`:NUMeric:FLICker:INFOrmation:NUMber`" is output in order.
- The format of individual judgment results that are output is as follows:
  - Judgment result (JTOTal, JDC, JDMAX, JDT, JPST, JPLT)

	ASCII (<NR1> format)	**FLOAT (IEEE single-precision floating point (4-byte) format)**
Pass:	"0"	0x00000000 (0)
Fail:	"–1"	0xBF800000 (–1)
Error:	"–2"	0xC0000000 (–2)
-----:	"1"	0x3F800000 (1)
(space):	"1"	0x3F800000 (1)

- No items (NONE)
  ASCII: "NAN" (Not A Number)
  FLOAT: 0x7E951BEE (9.91E+37)

:NUMeric:FLICker:JUDGement?

Function Queries the judgment results (fixed format).

Syntax `:NUMeric:FLICker:JUDGement? {<NRf>|ALL}`
`:NUMeric:FLICker:JUDGement?`
<NRf> = 1 to9 (observation period number)
ALL = Overall observation period (Result)

Example
- Example in which “`:NUMeric:FORMat`” is set to “`ASCii`”
  ```
  :NUMERIC:FLICKER:JUDGEMENT? 1 ->
  0,0,0,0,0,0,0,-1,-1,-1,-1,-1,-1,
  -1,-1,-1,-1,-1,-1
  ```
- Example in which `:NUMeric:FORMat` is set to “`FLOat`".
  `:NUMERIC:FLICKER:JUDGEMENT? -> #4` (number of bytes, four digits) (data byte sequence)

Description
- Only available with the flicker measurement function (/FL option).
- The contents and order of the judgment results that are output are in the fixed format below. To modify the output contents and order, use the “`:NUMeric:FLICker:INFOrmation`” commands.

* **Output contents and order of Judgment results**

For function names, see the description for “`:NUMeric:FLICker:INFOrmation:ITEM<x>`”. Numbers refer to elements. (ALL) is the overall judgment result for all measured elements.

JTOTal(ALL)→
JDC1→JDMAX1→JDT1→JPST1→
JPLT1→JTOTal1→
JDC2→JDMAX2→JDT2→JPST2→
JPLT2→JTOTal12→(output only for 2 to 4 element models)
JDC3→JDMAX3→JDT3→JPST3→
JPLT3→JTOTal13→ (output only for 3 to 4 element models)
JDC4→JDMAX4→JDT4→JPST4→
JPLT4→JTOTal14→ (output only for 4 element models)

For 1 element models, 7 data from JTOTal(ALL) to JTOTal1 are output.
For 2 element models, 13 data from JTOTal(ALL) to JTOTal2 are output.
For 3 element models, 19 data from JTOTal(ALL) to JTOTal3 are output.
For 4 element models, 25 data from JTOTal(ALL) to JTOTal4 are output.

- If parameters are specified, the judgment results of the specified observation period are output.
- If parameters are omitted, the judgment result of the overall observation period (Result) is output (the same output occurs as when the ALL parameters are specified).
- For the format of individual numeric data, see the description for “`:NUMeric:FLICker:INFOrmation:VALue?`”.

:NUMeric:FLICker:PERiod?

Function Queries the observation period number currently being measured during flicker measurement.

Syntax `:NUMeric:FLICker:PERiod?`

Example `:NUMERIC:FLICKER:PERIOD? -> 5`

Description
- Only available with the flicker measurement function (/FL option).
- Returns the observation period numbers marked with an asterisk (*) in the No. column of the numeric list in the flicker measurement screen. If no asterisks are displayed (such as after a reset or during initialization), 0 is returned.

:NUMeric:FLICker:VALue?

Function Queries the measured flicker data (fixed format).

Syntax
```
:NUMeric:FLICker:VALue? {<NRf>|ALL}
:NUMeric:FLICker:VALue?
```
<NRf> = 1 to 99 (observation period number)
ALL = Overall observation period (Result)

Example
- Example in which ":NUMeric:FORMat" is set to "ASCii".
```
:NUMERIC:FLICKER:VALUE? 1 ->
600,229.75E+00,50.000E+00,
1.52E+00,1.56E+00,3E+00,...
```
- Example in which ":NUMeric:FORMat" is set to "FLOat".
  `:NUMERIC:FLICKER:VALUE? -> #4` (number of bytes, four digits) (data byte sequence)

Description
- Only available with the flicker measurement function (/FL option).
- The contents and order of the numeric data that are output are in the following fixed format. To modify the output contents and order, use the ":NUMeric:FLICker:FUNCtion" commands.

*** Output contents and order of numeric data**
For function names, see the description for ":NUMeric:FLICker:FUNCtion:ITEM<x>".
Numbers refer to elements.
TIME→
UN1→FU1→DC1→DMAX1→DT1→PST1→PLT1→
UN2→FU2→DC2→DMAX2→DT2→PST2→PLT2→ (output only for 2 to 4 element models)
UN3→FU3→DC3→DMAX3→DT3→PST3→PLT3→ (output only for 3 to 4 element models)
UN4→FU4→DC4→DMAX4→DT4→PST4→PLT4→ (output only for 4 element models)
For 1 element models, 8 data from TIME to PLT1 are output.
For 2 element models, 15 data from TIME to PLT2 are output.
For 3 element models, 22 data from TIME to PLT3 are output.
For 4 element models, 29 data from TIME to PLT4 are output.

- If the parameters are specified, the numeric data of the specified observation period is output.
- If the parameters are omitted, the measured data of the current observation period being measured is output. When in measurement complete status, the measured data of the overall observation period (Result) is output.
- For the format of individual numeric data, see the description for ":NUMeric:FLICker:FUNCtion:VALue?".

:NUMeric:FORMat

Function Sets the format of the numeric data that is transmitted by ":NUMeric[:NORMal]:VALue?" or ":NUMeric:LIST:VALue?" or queries the current setting.

Syntax
```
:NUMeric:FORMat {ASCii|FLOat}
:NUMeric:FORMat?
```

Example
```
:NUMERIC:FORMAT ASCII
:NUMERIC:FORMAT? ->
:NUMERIC:FORMAT ASCII
```

Description
- The format of the numeric data that is output varies depending on the ":NUMeric:FORMat" setting as follows:
  (1) When "ASCii" is specified
  Outputs the physical value in <NR3> format (<NR1> format only for the elapsed time of integration (TIME)).
  The data of each item is delimited by a comma.
  (2) When "FLOat" is specified
  A 6-byte or 8-byte header (example "#40060" or "#6000408") is added in front of the numeric data block. The physical value in IEEE single-precision floating point (4-byte) format follows the header. The byte order of the data of each item is MSB First.
- For the format of the individual numeric data, see "Numeric Data Format" at the end of this group of commands (see page 6-97).

:NUMeric:HOLD

Function Sets whether to hold (ON) or release (OFF) all the numeric data or queries the current setting.

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

Example
```
:NUMERIC:HOLD ON
:NUMERIC:HOLD? -> :NUMERIC:HOLD 1
```

Description
- If `:NUMeric:HOLD` is turned ON before executing "`:NUMeric[:NORMal]:VALue?`" or "`:NUMeric:LIST:VALue?`," all the numeric data at that point can be held internally.
- As long as `:NUMeric:HOLD` is ON, the numeric data is held even when the numeric data on the screen is updated.
- For example, if you wish to retrieve various types of numeric data of each element at the same point, do the following:

  `:NUMeric:HOLD ON`

  `:NUMeric[:NORMal]:ITEM1 U,1,TOTAL; ITEM2 I,1,TOTAL;...` (set the numeric data items of element 1)

  `:NUMeric[:NORMal]:VALue?`

  (Receive the numeric data of element 1)

  `:NUMeric[:NORMal]:ITEM1 U,2,TOTAL; ITEM2 I,2,TOTAL;...` (set the numeric data items of element 2)

  `:NUMeric[:NORMal]:VALue?`

  (Receive the numeric data of element 2)

  ...(omitted)...

  `:NUMeric[:NORMal]:ITEM1 U,4,TOTAL; ITEM2 I,4,TOTAL;...` (set the numeric data items of element 4)

  `:NUMeric[:NORMal]:VALue?`

  (Receive the numeric data of element 4)

  `:NUMeric:HOLD OFF`
- If ON is specified when `:NUMeric:HOLD` is ON, the numeric data is cleared once, and the most recent numeric data is held internally. This method can be used when retrieving numeric data continuously (no need to set `:NUMeric:HOLD` to OFF each time).

:NUMeric:LIST?

Function Queries all settings related to the numeric list data output of harmonic measurement.

Syntax
```
:NUMeric:LIST?
```

Example
```
:NUMERIC:LIST? ->
:NUMERIC:LIST:NUMBER 1;
ORDER 100;SELECT ALL;ITEM1 U,1
```

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- For the values of "`:NUMeric:LIST:ITEM<x>`," the numeric list data output items for the amount specified by "`:NUMeric:LIST:NUMber`" are output.

:NUMeric:LIST:CLEar

Function Clears the output items of the numeric list data of harmonic measurement (set to "NONE").

Syntax
```
:NUMeric:LIST:CLEar {ALL|
<NRf>[,<NRf>]}
```
ALL = Clear all items

1st <NRf> = 1 to 64 (Item number to start clearing)

2nd <NRf> = 1 to 64 (Item number to end clearing)

Example
```
:NUMERIC:LIST:CLEAR ALL
```

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- If the 2nd <NRf> is omitted, the output items from the start clear number to the last item (64) are cleared.

:NUMeric:LIST:DELete

Function Deletes the output items of the numeric list data of harmonic measurement.

Syntax `:NUMeric:LIST:DELete {<NRf>[,<NRf>]}`
1st <NRf> = 1 to 64 (Item number to start deleting)
2nd <NRf> = 1 to 64 (Item number to end deleting)

Example `:NUMERIC:LIST:CLEAR 1` (Deletes ITEM1 and shift ITEM2 and subsequent items forward)
`:NUMERIC:LIST:CLEAR 1,3` (Deletes ITEM1 to ITEM3 and shift ITEM4 and subsequent items forward)

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- The subsequent output items fill the positions of deleted output items, and empty sections at the end are set to "NONE."
- If the 2nd <NRf> is omitted, only the output item of the delete start number is deleted.

:NUMeric:LIST:ITEM<x>

Function Sets the output items (function elements) of the numeric list data of harmonic measurement or queries the current setting.

Syntax `:NUMeric:LIST:ITEM<x> {NONE|<Function>,<Element>}`
`:NUMeric:LIST:ITEM<x>?`
<x> = 1 to 64 (item number)
NONE = No output item
<Function> = `{U|I|P|S|Q|LAMBda|PHI|PHIU|PHII|Z|RS|XS|RP|XP|UHDF|IHDF|PHDF}`
<Element> = `{<NRf>|SIGMA|SIGMB}`
(<NRf> = 1 to 4)

Example `:NUMERIC:LIST:ITEM1 U,1`
`:NUMERIC:LIST:ITEM1? ->`
`:NUMERIC:LIST:ITEM1 U,1`

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:NUMeric:LIST:NUMber

Function Sets the number of the numeric list data that is transmitted by "`:NUMeric:LIST:VALue?`" or queries the current setting.

Syntax `:NUMeric:LIST:NUMber {<NRf>|ALL}`
`:NUMeric:LIST:NUMber?`
<NRf> = 1 to 64 (ALL)

Example `:NUMERIC:LIST:NUMBER 5`
`:NUMERIC:LIST:NUMBER ->`
`:NUMERIC:LIST:NUMBER 5`

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- If the parameter is omitted for the "`:NUMeric:LIST:VALue?`" command, the numeric list data from 1 to (the specified value) is output in order.
- By default, the number of numeric data is set to "1."

:NUMeric:LIST:ORDer

Function Sets the maximum output order of the numeric list data of harmonic measurement or queries the current setting.

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`

Description This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).

:NUMeric:LIST:PRESet

Function Sets the output items of harmonic measurement numeric list data to a preset pattern.

Syntax `:NUMeric:LIST:PRESet {<NRf>}`
<NRf> = 1 to 4

Example `:NUMERIC:LIST:PRESET 1`

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- For details on the output items that are preset, see "(2) Preset Pattern of the Numeric List Data Output Items of Harmonic Measurement " (see page 6-100).
- By default, output items of "Pattern 2" is selected.

:NUMeric:LIST:SELect

Function Sets the output component of the numeric list data of harmonic measurement or queries the current setting.

Syntax
```
:NUMeric:LIST:SELect {EVEN|ODD|ALL}
:NUMeric:LIST:SELect?
```

Example
```
:NUMERIC:LIST:SELECT ALL
:NUMERIC:LIST:SELECT? ->
:NUMERIC:LIST:SELECT ALL
```

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- The selections are as shown below:
  EVEN = Outputs the components of TOTal, DC, and even order harmonic
  ODD = Outputs the components of TOTal, DC, and odd order harmonic
  ALL = Outputs all components

:NUMeric:LIST:VALue?

Function Queries the numeric list data of harmonic measurement.

Syntax `:NUMeric:LIST:VALue? {<NRf>}`
<NRf> = 1 to 64 (item number)

Example
- Example when <NRf> is specified
  `:NUMERIC:LIST:VALUE? 1 -> 103.58E+00,0.00E+00,103.53E+00, 0.09E+00,2.07E+00,0.04E+00, ..`(omitted)`..,0.01E+00,0.01E+00` (up to 102 items of data)
- Example when <NRf> is omitted (when "`:NUMeric:LIST:NUMber`" is 5)
  `:NUMERIC:LIST:VALUE? -> 103.58E+00,0.00E+00,103.53E+00, 0.09E+00,2.07E+00,0.04E+00, ..`(omitted)`..,0.00E+00,0.00E+00` (up to 102*5 = 510 items of data)
- Example in which "`:NUMeric:FORMat`" is set to "FLOat"
  `:NUMERIC:LIST:VALUE? -> #6`(number of bytes, 6 digits)(data byte sequence)

Description
- This command is valid only on models with the advanced computation function (/G6 option) or the harmonic measurement function (/G5 option).
- A single numeric list data consists of up to 102 items of numeric data in the following order: TOTal, DC, 1st order, ... "`:NUMeric:LIST:ORDer`."
- If <NRf> is specified, only the numeric list data of the item number is output (up to 102 items of data).
- If <NRf> is omitted, the numeric list data of item numbers from 1 to "`:NUMeric:LIST:NUMber`" is output in order (up to 102*"`:NUMeric:LIST:NUMber`" items of data).
- For the format of the individual numeric data that is output, see "Numeric Data Format" at the end of this group of commands (see page 6-97).

:NUMeric:NORMal?

Function Queries all settings related to the numeric data output.

Syntax `:NUMeric:NORMal?`

Example `:NUMERIC:NORMAL? -> :NUMERIC:NORMAL:NUMBER 15;ITEM1 U,1,TOTAL;ITEM2 I,1,TOTAL;ITEM3 P,1,TOTAL;ITEM4 S,1,TOTAL;ITEM5 Q,1,TOTAL;ITEM6 LAMBDA,1,TOTAL;ITEM7 PHI,1,TOTAL;ITEM8 FU,1;ITEM9 FI,1;ITEM10 UPPEAK,1;ITEM11 UMPEAK,1;ITEM12 IPPEAK,1;ITEM13 IMPEAK,1;ITEM14 CFU,1;ITEM15 CFI,1`

Description For the values of ":NUMeric[:NORMal]:ITEM<x>," the numeric data output items for the amount specified by ":NUMeric[:NORMal]:NUMber" are output.

:NUMeric[:NORMal]:CLEar

Function Clears the numeric data output item (sets "NONE").

Syntax `:NUMeric[:NORMal]:CLEar {ALL|<NRf>[,<NRf>]}`
ALL = Clear all items
1st <NRf> = 1 to 255 (Item number to start clearing)
2nd <NRf> = 1 to 255 (Item number to end clearing)

Example `:NUMERIC:NORMAL:CLEAR ALL`

Description If the 2nd <NRf> is omitted, the output items from the start clear number to the last item (255) are cleared.

:NUMeric[:NORMal]:DELete

Function Deletes the output items of numeric data.

Syntax `:NUMeric[:NORMal]:DELete {<NRf>[,<NRf>]}`
1st <NRf> = 1 to 255 (Item number to start deleting)
2nd <NRf> = 1 to 255 (Item number to end deleting)

Example `:NUMERIC:NORMAL:CLEAR 1` (Deletes ITEM1 and shift ITEM2 and subsequent items forward)
`:NUMERIC:NORMAL:CLEAR 1,3` (Deletes ITEM1 to ITEM3 and shift ITEM4 and subsequent items forward)

Description
- The subsequent output items fill the positions of deleted output items, and empty sections at the end are set to "NONE."
- If the 2nd <NRf> is omitted, only the output item of the delete start number is deleted.

:NUMeric[:NORMal]:ITEM<x>

Function Sets the numeric data output items (function, element, and harmonic order) or queries the current setting.

Syntax `:NUMeric[:NORMal]:ITEM<x> {NONE|<Function>,<Element>[,<Order>]}`
`:NUMeric[:NORMal]:ITEM<x>?`
<x> = 1 to 255 (item number)
NONE = No output item
<Function> = {U|I|P|S|Q|...}(See the function selection list (1) of "DISPlay group" on page 6-44.)
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)
<Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 100)

Example `:NUMERIC:NORMAL:ITEM1 U,1,TOTAL`
`:NUMERIC:NORMAL:ITEM1? -> :NUMERIC:NORMAL:ITEM1 U,1,TOTAL`

Description
- If <Element> is omitted, element 1 is set.
- If <Order> is omitted, TOTal is set.
- <Element> or <Order> is omitted from response to functions that do not need them.

:NUMeric[:NORMal]:NUMber

Function Sets the number of the numeric data that is transmitted by ":NUMeric[:NORMal]:VALue?" or queries the current setting.

Syntax `:NUMeric[:NORMal]:NUMber {<NRf>|ALL}`
`:NUMeric[:NORMal]:NUMber?`
<NRf> = 1 to 255(ALL)

Example `:NUMERIC:NORMAL:NUMBER 15`
`:NUMERIC:NORMAL:NUMBER -> :NUMERIC:NORMAL:NUMBER 15`

Description
- If the parameter is omitted for the ":NUMeric[:NORMal]:VALue?" command, the numeric data from 1 to (the specified value) is output in order.
- By default, the number of numeric data is set to "15."

:NUMeric[:NORMal]:PRESet

Function Presets the output item pattern of numeric data.

Syntax `:NUMeric[:NORMal]:PRESet {<NRf>}`
<NRf> = 1 to 4

Example `:NUMERIC:NORMAL:PRESET 1`

Description
- For details on the output items that are preset, see "(1) Preset Pattern of Numeric Data Output Items" (page 6-98).
- By default, output items of "Pattern 2" is selected.

:NUMeric[:NORMal]:VALue?

Function Queries the numeric data.

Syntax `:NUMeric[:NORMal]:VALue? {<NRf>}`
<NRf> = 1 to 255 (item number)

Example
- Example when <NRf> is specified
  `:NUMERIC:NORMAL:VALUE? 1 -> 104.75E+00`
- Example when <NRf> is omitted
  `:NUMERIC:NORMAL:VALUE? -> 104.75E+00,105.02E+00, -0.38E+00,..`(omitted)`..,49.868E+00`
- Example in which "`:NUMeric:FORMat`" is set to "FLOat"
  `:NUMERIC:NORMAL:VALUE? -> #4`(number of bytes, 4 digits)(data byte sequence)

Description
- If <NRf> is specified, only the numeric data of the item number is output.
- If <NRf> is omitted, the numeric data of item numbers from 1 to "`:NUMeric[:NORMal]:NUMber`" is output in order.
- For the format of the individual numeric data that is output, see "Numeric Data Format" at the end of this group of commands (see page 6-97).

* Numeric Data Format

(1)Normal Data

- Frequency (FU and FI)
  ASCII: <NR3> format (mantissa: 5 digits, exponent: 2 digits, example: 50.000E+00)
  FLOAT: IEEE single-precision floating point (4-byte) format
- Elapsed time of integration (TIME)
  ASCII: <NR1> format in units of seconds (example: for 1 hour (1:00:00), 3600)
  FLOAT: IEEE single-precision floating point (4-byte) format in units of seconds (example: for 1 hour (1:00:00), 0x45610000)
- Peak information (PKU, PKI, PKSPeed, PKTorque) for Cycle by Cycle measurement

	ASCII (<NR1> format)	FLOAT (IEEE single-precision floating point (4-byte) format)
No peak “ ”:	“0”	0x00000000 (0)
Positive peak “↑+”:	“1”	0x3F800000 (1)
Negative peak “↓–”:	“2”	0x40000000 (2)
Positive and negative peak “↕ ±”:	“3”	0x40400000 (3)

Note

If the main unit’s peak over detection function makes a detection during the measurement period, 4 is added to the numbers above.

- No items (NONE)
  ASCII: “NAN” (Not A Number)
  FLOAT: 0x7E951BEE (9.91E+37)
- Other than above
  ASCII: <NR3> format (mantissa: maximum significant digits = 6, exponent: 2 digits, example: [-]123.456.45E+00)
  FLOAT: IEEE single-precision floating point (4-byte) format

(2)Error Data

- Data does not exist (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)

Note

- For the 180° (Lead/Lag) display of the phase difference ϕ (PHI) of elements 1 to 4, the values are output in the range between -180.000 to 180.000 with lead (D) and lag (G) set to negative and positive values, respectively.
- For the Σ of power values (P, S, Q, and PC), the number of digits of the mantissa may be equal to 7 (the maximum significant digits) depending on the combination of the voltage range and current range (power range). See the list of power ranges in the *User’s Manual IM760301-01E*.

* List of Numeric Data Output Items That Are Preset

The list of function names used in the commands and the corresponding function names used on the screen menu of the WT3000 is given in the Function Selection List in the DISPlay group.

Note

The List of Numeric Data Output Items That Are Preset indicates the measurement function and element that are assigned to each item number (ITEM<x>). Items that are not set to be measured are displayed or output in the same fashion as when the data does not exist. For example, if frequency FI of the current of element 2 is not set to be measured, the output of item number ITEM19 is the same as the output when the data does not exist (NAN for ASCII).

(1) Preset Pattern of Numeric Data Output Items

Applicable command "`:NUMeric[:NORMal]:PRESet`"

- **Pattern 1**

ITEM<x>	<Function>,	<Element>,	<Order>
1	U,	1,	TOTal
2	I,	1,	TOTal
3	P,	1,	TOTal
4	S,	1,	TOTal
5	Q,	1,	TOTal
6	LAMBda,	1,	TOTal
7	PHI,	1,	TOTal
8	FU,	1,	(TOTal)
9	FI,	1,	(TOTal)
10	NONE,		
11 to 19	U to FI,	2,	TOTal
20	NONE,		
21 to 29	U to FI,	3,	TOTal
30	NONE,		
31 to 39	U to FI,	4,	TOTal
40	NONE,		
41 to 49	U to FI,	SIGMA,	TOTal
50	NONE,		
51 to 59	U to FI,	SIGMB,	TOTal
60	NONE,		
61 to 255	NONE,		

- **Pattern 2**

ITEM<x>	<Function>,	<Element>,	<Order>
1	U,	1,	TOTal
2	I,	1,	TOTal
3	P,	1,	TOTal
4	S,	1,	TOTal
5	Q,	1,	TOTal
6	LAMBda,	1,	TOTal
7	PHI,	1,	TOTal
8	FU,	1,	(TOTal)
9	FI,	1,	(TOTal)
10	UPPeak,	1,	(TOTal)
11	UMPeak,	1,	(TOTal)

12	IPPeak,	1,	(TOTal)
13	IMPeak,	1,	(TOTal)
14	CFU,	1,	(TOTal)
15	CFI,	1,	(TOTal)
16 to 30	U to CFI,	2,	TOTal
31 to 45	U to CFI,	3,	TOTal
46 to 60	U to CFI,	4,	TOTal
61 to 75	U to CFI,	SIGMA,	TOTal
76 to 90	U to CFI,	SIGMB,	TOTal
91 to 255	NONE,		

- **Pattern 3**

ITEM<x>	<Function>,	<Element>,	<Order>
1	U,	1,	TOTal
2	I,	1,	TOTal
3	P,	1,	TOTal
4	S,	1,	TOTal
5	Q,	1,	TOTal
6	TIME,	1,	(TOTal)
7	WH,	1,	(TOTal)
8	WHP,	1,	(TOTal)
9	WHM,	1,	(TOTal)
10	AH,	1,	(TOTal)
11	AHP,	1,	(TOTal)
12	AHM,	1,	(TOTal)
13	WS,	1,	(TOTal)
14	WQ,	1,	(TOTal)
15	NONE,		
16 to 29	U to WQ,	2,	TOTal
30	NONE,		
31 to 44	U to WQ,	3,	TOTal
45	NONE,		
46 to 59	U to WQ,	4,	TOTal
60	NONE,		
61 to 74	U to WQ,	SIGMA,	TOTal
75	NONE,		
76 to 89	U to WQ,	SIGMB,	TOTal
90	NONE,		
91 to 255	NONE,		

- **Pattern 4**

ITEM<x>	<Function>,	<Element>,	<Order>
1	U,	1,	TOTal
2	I,	1,	TOTal
3	P,	1,	TOTal
4	S,	1,	TOTal
5	Q,	1,	TOTal
6	LAMBda,	1,	TOTal
7	PHI,	1,	TOTal
8	FU,	1,	(TOTal)

9	FI,	1,	(TOTal)
10	UPPeak,	1,	(TOTal)
11	UMPeak,	1,	(TOTal)
12	IPPeak,	1,	(TOTal)
13	IMPeak,	1,	(TOTal)
14	CFU,	1,	(TOTal)
15	CFI,	1,	(TOTal)
16	PC,	1,	(TOTal)
17	TIME,	1,	(TOTal)
18	WH,	1,	(TOTal)
19	WHP,	1,	(TOTal)
20	WHM,	1,	(TOTal)
21	AH,	1,	(TOTal)
22	AHP,	1,	(TOTal)
23	AHM,	1,	(TOTal)
24	WS,	1,	(TOTal)
25	WQ,	1,	(TOTal)
26 to 50	U to WQ,	2,	TOTal
51 to 75	U to WQ,	3,	TOTal
76 to 100	U to WQ,	4,	TOTal
101 to 125	U to WQ,	SIGMA,	TOTal
126 to 150	U to WQ,	SIGMB,	TOTal
151 to 255	NONE,		

(2) Preset Pattern of the Numeric List Data Output Items of Harmonic Measurement

Applicable command "`: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	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,	

- **Pattern 4**

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	Z,	1
11	RS,	1
12	XS,	1
13	RP,	1
14	XP,	1
15 to 28	U to XP,	2
29 to 42	U to XP,	3
43 to 56	U to XP,	4
57 to 64	NONE,	

6.19 RATE Group

The commands in this group deal with the data update rate.
You can make the same settings and inquiries as when UPDATE RATE on the front panel is used.

:RATE

Function Sets the data update rate or queries the current setting.

Syntax
```
:RATE {<Time>}
:RATE?
```
<Time> = 50, 100, 250, 500 (ms), 1, 2, 5, 10, or 20 (s)

Example
```
:RATE 500MS
:RATE? -> :RATE 500.0E-03
```

6.20 STATus Group

The commands in the STATus group are used to make settings and inquiries related to the status report. There are no front panel keys that correspond to the commands in this group. For details on the status report, see chapter 7.

:STATus?

Function Queries all settings related to the communication status function.

Syntax `:STATus?`

Example
```
:STATUS? -> :STATUS:EESE 0;
FILTER1 NEVER;FILTER2 NEVER;
FILTER3 NEVER;FILTER4 NEVER;
FILTER5 NEVER;FILTER6 NEVER;
FILTER7 NEVER;FILTER8 NEVER;
FILTER9 NEVER;FILTER10 NEVER;
FILTER11 NEVER;FILTER12 NEVER;
FILTER13 NEVER;FILTER14 NEVER;
FILTER15 NEVER;FILTER16 NEVER;
QENABLE 1;QMESSAGE 1
```

:STATus:CONDition?

Function Queries the contents of the condition register.

Syntax `:STATus:CONDition?`

Example `:STATUS:CONDITION? -> 16`

Description For details on the condition register, see chapter 7, “Status Report.”

:STATus:EESE (Extended Event Status Enable register)

Function Sets the extended event enable register or queries the current setting.

Syntax
```
:STATus:EESE <Register>
:STATus:EESE?
```
<Register> = 0 to 65535

Example
```
:STATUS:EESE #B0000000000000000
:STATUS:EESE? -> :STATUS:EESE 0
```

Description For details on the extended event enable register, see chapter 7, “Status Report.”

:STATus:EESR? (Extended Event Status Register)

Function Queries the content of the extended event register and clears the register.

Syntax `:STATus:EESR?`

Example `:STATUS:EESR? -> 0`

Description For details on the extended event register, see chapter 7, “Status Report.”

:STATus:ERRor?

Function Queries the error code and message information (top of the error queue).

Syntax `:STATus:ERRor?`

Example
```
:STATUS:ERROR? ->
113,"Underfined Header"
```

Description
- When there is no error, “0, “No error”” is returned.
- The message cannot be returned in Japanese.
- You can specify whether to add the message using the “STATus:QMESsage” command.

:STATus:FILTer<x>

Function Sets the transition filter or queries the current setting.

Syntax
```
:STATus:FILTer<x> {RISE|FALL|BOTH|
NEVer}
:STATus:FILTer<x>?
```
<x> = 1 to 16

Example
```
:STATUS:FILTER2 RISE
:STATUS:FILTER2? -> :STATUS:FILTER2
RISE
```

Description
- Specify how each bit of the condition register is to change to set the event. If “RISE” is specified, the event is set when the bit changes from 0 to 1.
- For details on the transition, see chapter 7, “Status Report.”

:STATus:QENable

Function Sets whether to store messages other than errors to the error queue (ON/OFF) or queries the current setting.

Syntax
```
:STATus:QENable {<Boolean>}
:STATus:QENable?
```

Example
```
:STATUS:QENABLE ON
:STATUS:QENABLE? ->
:STATUS:QENABLE 1
```

:STATus:QMESsage

Function Sets whether to attach message information to the response to the “STATus:ERRor?” query (ON/OFF) or queries the current setting.

Syntax
```
:STATus:QMESsage {<Boolean>}
:STATus:QMESsage?
```

Example
```
:STATUS:QMESSAGE ON
:STATUS:QMESSAGE? ->
:STATUS:QMESSAGE 1
```

:STATus:SPOLl? (Serial Poll)

Function Executes serial polling.

Syntax `:STATus:SPOLl?`

Example `:STATUS:SPOLL? -> :STATUS:SPOLL 0`

Description This command is dedicated to the optional RS-232, USB, or Ethernet interface. An interface message is available for the GP-IB interface.

6.21 STORe Group

The commands in this group deal with store and recall.
You can make the same settings and inquiries as when STORE and STORE SET (SHIFT+STORE) on the front panel is used.

:STORe?

Function Queries all settings related to store and recall.

Syntax `:STORe?`

Example
```
:STORE? -> STORE:MODE STORE;
DIRECTION MEMORY;SMODE MANUAL;
COUNT 100;INTERVAL 0,0,0;
ITEM NUMERIC;NUMERIC:NORMAL:
ELEMENT1 1;ELEMENT2 0;ELEMENT3 0;
ELEMENT4 0;SIGMA 0;SIGMB 0;U 1;I 1;
P 1;S 1;Q 1;LAMBDA 1;PHI 1;FU 1;
FI 1;UPPEAK 0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0;:STORE:
MEMORY:ALERT 1
```

:STORe:COUNt

Function Sets the store count or queries the current setting.

Syntax
```
:STORe:COUNt {<NRf>}
:STORe:COUNt?
```
<NRf> = 1 to 999999

Example
```
:STORE:COUNT 100
:STORE:COUNT? -> :STORE:COUNT 100
```

:STORe:DIRection

Function Sets the store destination or queries the current setting.

Syntax
```
:STORe:DIRection {MEMory|FILE}
:STORe:DIRection?
```

Example
```
:STORE:DIRECTION MEMORY
:STORE:DIRECTION? ->
:STORE:DIRECTION MEMORY
```

:STORe:FILE?

Function Queries all settings related to the saving of the stored data.

Syntax `:STORe:FILE?`

Example
```
:STORE:FILE? -> :STORE:FILE:
TYPE ASCII;ANAMING 1;NAME "DATA1";
COMMENT "CASE1"
```

:STORe:FILE:ANAMing

Function Sets whether to automatically name the files when saving the stored data or queries the current setting.

Syntax
```
:STORe:FILE:ANAMing {<Boolean>}
:STORe:FILE:ANAMing?
```

Example
```
:STORE:FILE:ANAMING ON
:STORE:FILE:ANAMING? ->
:STORE:FILE:ANAMING 1
```

:STORe:FILE:COMMent

Function Sets the comment to be added to the file when saving the stored data or queries the current setting.

Syntax
```
:STORe:FILE:COMMent {<String>}
:STORe:FILE:COMMent?
```
<String> = Up to 25 characters

Example
```
:STORE:FILE:COMMENT "CASE1"
:STORE:FILE:COMMENT? ->
:STORE:FILE:COMMENT "CASE1"
```

:STORe:FILE:NAME

Function Sets the name of the file when saving the stored data or queries the current setting.

Syntax
```
:STORe:FILE:NAME {<Filename>}
:STORe:FILE:NAME?
```

Example
```
:STORE:FILE:NAME "DATA1"
:STORE:FILE:NAME? ->
:STORE:FILE:NAME "DATA1"
```

Description Set the save destination drive and directory of the stored data using the following commands.

- Destination drive: "`:FILE:DRIVe`"
- Directory: "`:FILE:CDIRectory`"

The save destination path can be queried using the "`:FILE:PATH?`" command.

:STORe:FILE:TYPE

Function Sets the data format when saving the stored data or queries the current setting.

Syntax
```
:STORe:FILE:TYPE {ASCii|FLOat}
:STORe:FILE:TYPE?
```

Example
```
:STORE:FILE:TYPE ASCII
:STORE:FILE:TYPE? ->
:STORE:FILE:TYPE ASCII
```

:STORe:INTerval

Function Sets the store interval or queries the current setting.

Syntax `:STORe:INTerval {<NRf>,<NRf>,<NRf>}`
`:STORe:INTerval?`
1st <NRf> = 0 to 99 (hour)
2nd <NRf> = 0 to 59 (minute)
3rd <NRf> = 1 to 59 (second)

Example `:STORE:INTERVAL 0,0,0`
`:STORE:INTERVAL? ->`
`:STORE:INTERVAL 0,0,0`

:STORe:ITEM

Function Sets the stored item or queries the current setting.

Syntax `:STORe:ITEM {NUMeric|WAVE|NWAVe}`
`:STORe:ITEM?`
NUMeric = Store only the numeric data.
WAVE = Store only the waveform display data
NWAVe = Store both the numeric data and waveform display data

Example `:STORE:ITEM NUMERIC`
`:STORE:ITEM? -> :STORE:ITEM NUMERIC`

:STORe:MEMory?

Function Queries all settings related to the storage memory.

Syntax `:STORe:MEMory?`

Example `:STORE:MEMORY? ->`
`:STORE:MEMORY:ALERT 1`

:STORe:MEMory:ALERt

Function Sets whether to display a confirmation message when clearing the storage memory or queries the current setting.

Syntax `:STORe:MEMory:ALERt {<Boolean>}`
`:STORe:MEMory:ALERt?`

Example `:STORE:MEMORY:ALERT ON`
`:STORE:MEMORY:ALERT? ->`
`:STORE:MEMORY:ALERT 1`

Description The initialization takes place immediately when initializing the storage memory using the "`:STORe:MEMory:INITialize`" command regardless of the setting specified with this command.

:STORe:MEMory:CONVert:ABORt

Function Abort converting the stored data from the memory to the file.

Syntax `:STORe:MEMory:CONVert:ABORt`

Example `:STORE:MEMORY:CONVERT:ABORT`

:STORe:MEMory:CONVert:EXECute

Function Executes the converting of the stored data from the memory to the file.

Syntax `:STORe:MEMory:CONVert:EXECute`

Example `:STORE:MEMORY:CONVERT:EXECUTE`

Description
- The convert destination file is set using the "`:STORe:FILE:...`" command.
- When file conversion is executed, the WT3000 accesses the file twice.
  To confirm the completion of the file conversion, use the "`COMMUNICATE:WAIT 64`" command (checks the change in bit 6 (ACS) of the condition register) and check the completion of the file access of the WT3000 twice. An example is indicated below.
  `"STATUS:EESR?"`
  (Clear the extended event register)
  `"STORE:MEMORY:CONVERT:EXECUTE"`
  (Start the file conversion)
  `"COMMUNICATE:WAIT 64"`
  (Wait for the conversion to finish, the first time)
  `"STATUS:EESR?"`
  (Clear the extended event register)
  `"COMMUNICATE:WAIT 64"`
  (Wait for the conversion to finish, the second time)
  `"STATUS:EESR?"`
  (Clear the extended event register)

:STORe:MEMory:INITialize

Function Executes the initialization of the storage memory.

Syntax `:STORe:MEMory:INITialize`

Example `:STORE:MEMORY:INITIALIZE`

:STORe:MODE

Function Sets the data storage/recall or queries the current setting.

Syntax `:STORe:MODE {STORe|RECall}`
`:STORe:MODE?`

Example `:STORE:MODE STORE`
`:STORE:MODE? -> :STORE:MODE STORE`

:STORe:NUMeric?

Function Queries all settings related to the storage of numeric data.

Syntax `:STORe:NUMeric?`

Example
```
:STORE:NUMERIC? -> :STORE:NUMERIC:
NORMAL:ELEMENT1 1;ELEMENT2 0;
ELEMENT3 0;ELEMENT4 0;SIGMA 0;
SIGMB 0;U 1;I 1;P 1;S 1;Q 1;
LAMBDA 1;PHI 1;FU 1;FI 1;UPPEAK 0;
UMPEAK 0;IPPEAK 0;IMPEAK 0;CFU 0;
CFI 0;PC 0;TIME 0;WH 0;WHP 0;WHM 0;
AH 0;AHP 0;AHM 0;WS 0;WQ 0;ETA1 0;
ETA2 0;ETA3 0;ETA4 0;F1 0;F2 0;
F3 0;F4 0;F5 0;F6 0;F7 0;F8 0;F9 0;
F10 0;F11 0;F12 0;F13 0;F14 0;
F15 0;F16 0;F17 0;F18 0;F19 0;F20 0
```

:STORe:NUMeric:NORMal?

Function Queries all settings related to the stored items of numeric data.

Syntax `:STORe:NUMeric:NORMal?`

Example
```
:STORE:NUMERIC:NORMAL? ->
:STORE:NUMERIC:NORMAL:ELEMENT1 1;
ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;
SIGMA 0;SIGMB 0;U 1;I 1;P 1;S 1;
Q 1;LAMBDA 1;PHI 1;FU 1;FI 1;
UPPEAK 0;UMPEAK 0;IPPEAK 0;
IMPEAK 0;CFU 0;CFI 0;PC 0;TIME 0;
WH 0;WHP 0;WHM 0;AH 0;AHP 0;AHM 0;
WS 0;WQ 0;ETA1 0;ETA2 0;ETA3 0;
ETA4 0;F1 0;F2 0;F3 0;F4 0;F5 0;
F6 0;F7 0;F8 0;F9 0;F10 0;F11 0;
F12 0;F13 0;F14 0;F15 0;F16 0;
F17 0;F18 0;F19 0;F20 0
```

:STORe:NUMeric[:NORMal]:ALL

Function Collectively turns ON/OFF the output of all element functions when storing the numerical data.

Syntax
```
:STORe:NUMeric[:NORMal]:
ALL {<Boolean>}
```

Example `:STORE:NUMERIC[:NORMAL]:ALL ON`

:STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB}

Function Turns ON/OFF the output of {each element|ΣA|ΣB} when storing the 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
```

Description
- `:STORe:NUMeric[:NORMal]:SIGMA` is valid on models with two or more elements. To turn the output ON, wiring unit ΣA must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.
- `:STORe:NUMeric[:NORMal]:SIGMB` is valid on models with four elements. To turn the output ON, wiring unit ΣB must exist by setting the wiring system beforehand using the `[:INPut]WIRing` command.

:STORe:NUMeric[:NORMal]:PRESet<x>

Function Presets the output ON/OFF pattern of the element function for storing the numeric data.

Syntax `:STORe:NUMeric[:NORMal]:PRESet<x>`
<x> = 1 to 2 (preset pattern number)

Example `:STORE:NUMERIC:NORMAL:PRESET1`

Description For details on the storage pattern when preset is executed, see the *User's Manual IM760301-01E*.

:STORe:NUMeric[:NORMal]:<Function>

Function Turns ON/OFF the output of the function when storing the numerical data or queries the current setting.

Syntax
```
:STORe:NUMeric[:NORMal]:<Function>
{<Boolean>}
:STORe:NUMeric[:NORMal]:<Function>?
```
<Function> = `{U|I|P|S|Q|...}`(See the function selection list (1) of "DISPlay group" on page 6-44.)

Example
```
:STORE:NUMERIC:NORMAL:U ON
:STORE:NUMERIC:NORMAL:U? ->
:STORE:NUMERIC:NORMAL:U 1
```

:STORe:RECall

Function Sets the data number to be recalled or queries the current setting.

Syntax `:STORe:RECall {<NRf>}`
`:STORe:RECall?`
<NRf> = 1 to 999999

Example `:STORE:RECALL 1`
`:STORE:RECALL? -> :STORE:RECALL 1`

:STORe:RTIMe?

Function Queries the store reservation time for real-time store mode.

Syntax `:STORe:RTIMe?`

Example `:STORE:RTIME? -> :STORE:RTIME: START 2005,1,1,0,0,0; END 2005,1,1,1,0,0`

:STORe:RTIMe:{STARt|END}

Function Sets the store {start|stop} reservation date/time for real-time store mode or queries the current setting.

Syntax `:STORe:RTIMe:{STARt|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}`
`:STORe:RTIMe:{STARt|END}?`
{<NRf>, <NRf>, <NRf>, <NRf>, <NRf>, <NRf>} = 2001, 1, 1, 0, 0, 0 to 2099, 12, 31, 23, 59, 59
1st <NRf> = 2001 to 2099 (year)
2nd <NRf> = 1 to 12 (month)
3rd <NRf> = 1 to 31 (day)
4th <NRf> = 0 to 23 (hour)
5th <NRf> = 0 to 59 (minute)
6th <NRf> = 0 to 59 (second)

Example `:STORE:RTIME:START 2005,1,1,0,0,0`
`:STORE:RTIME:START? ->`
`:STORE:RTIME:START 2005,1,1,0,0,0`

Description This command is valid when the store mode (`:STORe:SMODe`) is set to RTIMe (real-time store mode).

:STORe:SMODe

Function Sets the store mode or queries the current setting.

Syntax `:STORe:SMODe {MANual|RTIMe|INTEGrate}`
`:STORe:SMODe?`
MANual = Manual store mode
RTIMe = Real-time store mode
INTEGrate = Integration synchronization store mode

Example `:STORE:SMODE MANUAL`
`:STORE:SMODE? ->`
`:STORE:SMODE MANUAL`

:STORe:STARt

Function Starts the data store operation.

Syntax `:STORe:STARt`

Example `:STORE:START`

Description When "`:STORe:SMODe`" is set to MANual, the storage operation is executed. When set to `{RTIMe|INTEGrate}` the WT3000 enters the store wait state.

:STORe:STOP

Function Stops the data storage operation.

Syntax `:STORe:STOP`

Example `:STORE:STOP`

:STORe:WAVE?

Function Queries all settings related to the storage of waveform display data.

Syntax `:STORe:WAVE?`

Example `:STORE:WAVE? -> :STORE:WAVE:U1 1; U2 0;U3 0;U4 0;I1 1;I2 0;I3 0;I4 0`

:STORe:WAVE:ALL

Function Collectively turns ON/OFF the output of all waveforms when storing waveform display data.

Syntax `:STORe:WAVE:ALL {<Boolean>}`

Example `:STORE:WAVE:ALL ON`

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

Function Turns ON/OFF the output of the waveform when storing the waveform display data or queries the current setting.

Syntax `:STORe:WAVE:{U<x>|I<x>|SPEed|TORQue} {<Boolean>}`
`:STORe:WAVE:{U<x>|I<x>|SPEed|TORQue}?`
<x> = 1 to 4

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

Description `{SPEed|TORQue}` are valid only on the motor version (-MV).

6.22 SYSTem Group

The commands in this group deal with the system.
You can make the same settings and inquiries as when MISC on the front panel is used.

:SYSTem?

Function Queries all settings related to the system.

Syntax `:SYSTem?`

Example
```
:SYSTEM? -> :SYSTEM:LANGUAGE:
MESSAGE ENGLISH;MENU ENGLISH;:
SYSTEM:FONT GOTHIC;KLOCK 0;SLOCK 0;
LCD:BRIGHTNESS 2;COLOR:GRAPH:
MODE DEFAULT;:SYSTEM:LCD:COLOR:
TEXT:MODE PRESET1
```

:SYSTem:CLOCk?

Function Sets all date/time related settings or queries the current setting.

Syntax `:SYSTem:CLOCk?`

Example
```
:SYSTEM:CLOCK? ->
:SYSTEM:CLOCK:DISPLAY 1;TYPE MANUAL
```

:SYSTem:CLOCk:DISPlay

Function Turns ON/OFF the date/time display or queries the current setting.

Syntax
```
:SYSTem:CLOCk:DISPlay {<Boolean>}
:SYSTem:CLOCk:DISPlay?
```

Example
```
:SYSTEM:CLOCK:DISPLAY ON
:SYSTEM:CLOCK:DISPLAY? ->
:SYSTEM:CLOCK:DISPLAY 1
```

:SYSTem:CLOCk:SNTP?

Function Sets all SNTP-based date/time related settings or queries the current setting.

Syntax `:SYSTem:CLOCk:SNTP?`

Example
```
:SYSTEM:CLOCK:SNTP? ->
:SYSTEM:CLOCK:SNTP:GMTTIME "09:00"
```

Description Available only with Ethernet (/C7 option).

:SYSTem:CLOCk:SNTP[:EXECute]

Function Sets the date/time via SNTP.

Syntax `:SYSTem:CLOCk:SNTP[:EXECute]`

Example `:SYSTEM:CLOCK:SNTP:EXECUTE`

Description Available only with Ethernet (/C7 option).

:SYSTem:CLOCk:SNTP:GMTTime

Function Sets the difference from Greenwich Mean Time or queries the current setting.

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
- Available only with Ethernet (/C7 option).
- Available when the date/time setting method (`:SYSTem:CLOCk:TYPE`) is SNTP.

:SYSTem:CLOCk:TYPE

Function Sets the date/time setting method or queries the current setting.

Syntax
```
:SYSTem:CLOCk:TYPE {MANual|SNTP}
:SYSTem:CLOCk:TYPE?
```

Example
```
:SYSTEM:CLOCK:TYPE MANUAL
:SYSTEM:CLOCK:TYPE? ->
:SYSTEM:CLOCK:TYPE MANUAL
```

Description SNTP is available only with Ethernet (/C7 option).

:SYSTem:DATE

Function Sets the date or queries the current setting.

Syntax
```
:SYSTem:DATE {<String>}
:SYSTem:DATE?
```
<String> = “YY/MM/DD” (YY = year, MM = month, DD = day)

Example
```
:SYSTEM:DATE "05/01/01"
:SYSTEM:DATE? -> "05/01/01"
```

Description “Year” is the lowest two digits of the year.

:SYSTem:ECLear

Function Clears the error message displayed on the screen.

Syntax `:SYSTem:ECLear`

Example `:SYSTEM:ECLEAR`

:SYSTem:FONT

Function Sets the display font or queries the current setting.

Syntax
```
:SYSTem:FONT {GOTHic|ROMan}
:SYSTem:FONT?
```

Example
```
:SYSTEM:FONT GOTHIC
:SYSTEM:FONT? ->
:SYSTEM:FONT GOTHIC
```

:SYSTem:KLOCk

Function Turns ON/OFF the key lock or queries the current setting.

Syntax
```
:SYSTem:KLOCk {<Boolean>}
:SYSTem:KLOCk?
```
Example
```
:SYSTEM:KLOCK OFF
:SYSTEM:KLOCK? -> :SYSTEM:KLOCK 0
```

:SYSTem:LANGuage?

Function Queries all settings related to the display language.

Syntax
```
:SYSTem:LANGuage?
```
Example
```
:SYSTEM:LANGUAGE? ->
:SYSTEM:LANGUAGE:MESSAGE ENGLISH;
MENU ENGLISH
```

:SYSTem:LANGuage:MENU

Function Sets the menu language or queries the current setting.

Syntax
```
:SYSTem:LANGuage:MENU {JAPANese|
ENGLish}
:SYSTem:LANGuage:MENU?
```
Example
```
:SYSTEM:LANGUAGE:MENU ENGLISH
:SYSTEM:LANGUAGE:MENU? ->
:SYSTEM:LANGUAGE:MENU ENGLISH
```

:SYSTem:LANGuage:MESSage

Function Sets the message language or queries the current setting.

Syntax
```
:SYSTem:LANGuage:MESSage {JAPANese|
ENGLish}
:SYSTem:LANGuage:MESSage?
```
Example
```
:SYSTEM:LANGUAGE:MESSAGE ENGLISH
:SYSTEM:LANGUAGE:MESSAGE? ->
:SYSTEM:LANGUAGE:MESSAGE ENGLISH
```

:SYSTem:LCD?

Function Queries all settings related to the LCD monitor.

Syntax
```
:SYSTem:LCD?
```
Example
```
:SYSTEM:LCD? ->
:SYSTEM:LCD:BRIGHTNESS 2;COLOR:
GRAPH:MODE DEFAULT;:SYSTEM:LCD:
COLOR:TEXT:MODE PRESET1
```

:SYSTem:LCD:BRIGhtness

Function Sets the brightness of the LCD monitor or queries the current setting.

Syntax
```
:SYSTem:LCD:BRIGhtness {<NRf>}
:SYSTem:LCD:BRIGhtness?
```
<NRf> = –1 to 3

Example
```
:SYSTEM:LCD:BRIGHTNESS 2
:SYSTEM:LCD:BRIGHTNESS? ->
:SYSTEM:LCD:BRIGHTNESS 2
```

:SYSTem:LCD:COLor?

Function Queries all settings related to the display colors of the LCD monitor.

Syntax
```
:SYSTem:LCD:COLor?
```
Example
```
:SYSTEM:LCD:COLOR? -> :SYSTEM:LCD:
COLOR:GRAPH:MODE DEFAULT;:SYSTEM:
LCD:COLOR:TEXT:MODE PRESET1
```

:SYSTem:LCD:COLor:GRAPh?

Function Queries all settings related to the display colors of the graphic items.

Syntax
```
:SYSTem:LCD:COLor:GRAPh?
```
Example
```
:SYSTEM:LCD:COLOR:GRAPH? ->
:SYSTEM:LCD:COLOR:GRAPH:MODE USER;
BACKGROUND 0,0,0;GRATICULE 6,6,6;
CURSOR 7,7,7;U1 7,7,0;U2 7,0,7;
U3 7,0,0;U4 0,4,7;I1 0,7,0;
I2 0,7,7;I3 7,4,0;I4 5,5,5
```

:SYSTem:LCD:COLor:GRAPh:{BACKground|GRATicule|CURSor|U<x>|I<x>}

Function Sets the display color of the {background|graticule|cursor|voltage waveform|current waveform} or queries the current setting.

Syntax
```
:SYSTem:LCD:COLor:GRAPh:
{BACKground|GRATicule|CURSor|U<x>|
I<x>} {<NRf>,<NRf>,<NRf>}
:SYSTem:LCD:COLor:GRAPh:
{BACKground|GRATicule|CURSor|U<x>|
I<x>}?
```
<x> = 1 to 4
<NRf> = 0 to 7

Example
```
:SYSTEM:LCD:COLOR:GRAPH:
BACKGROUND 0,0,0
:SYSTEM:LCD:COLOR:GRAPH:BACKGROUND?
-> :SYSTEM:LCD:COLOR:GRAPH:
BACKGROUND 0,0,0
```

Description Set the color in the order R, G, and B.
This command is valid when the display color mode of graphic items (:SYSTem:LCD:COLor:GRAPh:MODE) is set to “USER.”

:SYSTem:LCD:COLor:GRAPh:MODE

Function Sets the display color mode of the graphic items or queries the current setting.

Syntax
```
:SYSTem:LCD:COLor:GRAPh:
MODE {DEFault|USER}
:SYSTem:LCD:COLor:GRAPh:MODE?
```
Example
```
:SYSTEM:LCD:COLOR:GRAPH:
MODE DEFAULT
:SYSTEM:LCD:COLOR:GRAPH:MODE? ->
:SYSTEM:LCD:COLOR:GRAPH:
MODE DEFAULT
```

:SYSTem:LCD:COLor:TEXT?

Function Queries all settings related to the display colors of the text items.

Syntax `:SYSTem:LCD:COLor:TEXT?`

Example
```
:SYSTEM:LCD:COLOR:TEXT? ->
:SYSTEM:LCD:COLOR:TEXT:MODE USER;
LETTER 7,7,7;BACKGROUND 2,2,6;
BOX 0,0,7;SUB 3,3,3;SELECTED 0,4,7
```

:SYSTem:LCD:COLor:TEXT:{LETTer|BACKground|BOX|SUB|SELected}

Function Sets the display color of the {text (Menu Fore)|menu background (Menu Back)|selected menu (Select Box)|pop-up menu (Sub Menu)|selected key (Selected Key)} or queries the current setting.

Syntax
```
:SYSTem:LCD:COLor:TEXT:{LETTer|
BACKground|BOX|SUB|SELected} {<
NRf>,<NRf>,<NRf>}
:SYSTem:LCD:COLor:TEXT:{LETTer|
BACKground|BOX|SUB|SELected}?
```
<NRf> = 0 to 7

Example
```
:SYSTEM:LCD:COLOR:TEXT:LETTER 7,7,7
:SYSTEM:LCD:COLOR:TEXT:LETTER? ->
:SYSTEM:LCD:COLOR:TEXT:LETTER 7,7,7
```

Description Set the color in the order R, G, and B.
This command is valid when the display color mode of text items (`:SYSTem:LCD:COLor:TEXT:MODE`) is set to “USER.”

:SYSTem:LCD:COLor:TEXT:MODE

Function Sets the display color mode of the text items or queries the current setting.

Syntax
```
:SYSTem:LCD:COLor:TEXT:
MODE {PRESet<x>|USER}
:SYSTem:LCD:COLor:TEXT:MODE?
```
<x> = 1 to 3

Example
```
:SYSTEM:LCD:COLOR:TEXT:MODE PRESET1
:SYSTEM:LCD:COLOR:TEXT:MODE? ->
:SYSTEM:LCD:COLOR:TEXT:MODE PRESET1
```

:SYSTem:SLOCk

Function Sets whether to continue the SHIFT key ON state or queries the current setting.

Syntax
```
:SYSTem:SLOCk {<Boolean>}
:SYSTem:SLOCk?
```

Example
```
:SYSTEM:SLOCK OFF
:SYSTEM:SLOCK? -> :SYSTEM:SLOCK 0
```

:SYSTem:TIME

Function Sets the time or queries the current setting.

Syntax
```
:SYSTem:TIME {<String>}
:SYSTem:TIME?
```
<String> = “HH:MM:SS” (HH = hour, MM = minute, SS = second)

Example
```
:SYSTEM:TIME "14:30:00"
:SYSTEM:TIME? -> "14:30:00"
```

:SYSTem:USBKeyboard

Function Sets the USB keyboard type (language) or queries the current setting.

Syntax
```
:SYSTem:USBKeyboard {JAPANese|
ENGLish}
:SYSTem:USBKeyboard?
```

Example
```
:SYSTEM:USBKEYBOARD JAPANESE
:SYSTEM:USBKEYBOARD? ->
:SYSTEM:USBKEYBOARD JAPANESE
```

Description This command is valid only on models with the USB port (peripheral device) (/C5 option).

6.23 WAVeform Group

The commands in this group deal with the output of the retrieved waveform display data.
There are no front panel keys that correspond to the commands in this group.

:WAVeform?

Function Queries all settings related to the output of waveform display data.

Syntax `:WAVeform?`

Example
```
:WAVEFORM? -> :WAVEFORM:TRACE U1;
FORMAT ASCII;START 0;END 1001;
HOLD 0
```

:WAVeform:BYTeorder

Function Sets the output byte order of the waveform display data (FLOAT format) that is transmitted by "`:WAVeform:SEND?`" or queries the current setting.

Syntax
```
:WAVeform:BYTeorder {LSBFirst|
MSBFirst}
:WAVeform:BYTeorder?
```

Example
```
:WAVEFORM:BYTEORDER LSBFIRST
:WAVEFORM:BYTEORDER? ->
:WAVEFORM:BYTEORDER LSBFIRST
```

Description This value is valid when "`:WAVeform:FORMat`" is set to "`{FLOat}`."

:WAVeform:END

Function Sets the output end point of the waveform display data that is transmitted by "`:WAVeform:SEND?`" or queries the current setting.

Syntax
```
:WAVeform:END {<NRf>}
:WAVeform:END?
```
<NRf> = 0 to 1001

Example
```
:WAVEFORM:END 1001
:WAVEFORM:END? ->
:WAVEFORM:END 1001
```

:WAVeform:FORMat

Function Sets the format of the waveform display data that is transmitted by "`:WAVeform:SEND?`" or queries the current setting.

Syntax
```
:WAVeform:FORMat {ASCii|FLOat}
:WAVeform:FORMat?
```

Example
```
:WAVEFORM:FORMAT FLOAT
:WAVEFORM:FORMAT? ->
:WAVEFORM:FORMAT FLOAT
```

Description For the differences in the waveform display data output due to the format setting, see the description for "`:WAVeform:SEND?`."

:WAVeform:HOLD

Function Sets whether to hold (ON) or release (OFF) all the waveform display data or queries the current setting.

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

Example
```
:WAVEFORM:HOLD ON
:WAVEFORM:HOLD? -> :WAVEFORM:HOLD 1
```

Description
- If `:WAVeform:HOLD` is turned ON before executing "`:WAVeform:SEND?`," all the waveform data at that point can be held internally.
- As long as `:WAVeform:HOLD` is ON, the waveform data is held even when the waveform display on the screen is updated.
- For example, if you wish to retrieve the waveform display data of U1 and I1 at the same point, do the following:
```
    :WAVeform:HOLD ON
    :WAVeform:TRACe U1
    :WAVeform:SEND?
```
  (Receive the waveform display data of U1)
```
    :WAVeform:TRACe I1
    :WAVeform:SEND?
```
  (Receive the waveform display data of I1)
```
    :WAVeform:HOLD OFF
```
- If ON is specified when `:WAVeform:HOLD` is ON, the waveform display data is cleared once, and the most recent waveform data is held internally. This method can be used when retrieving waveform display data continuously (no need to set `:WAVeform:HOLD` to OFF each time).

:WAVeform:LENGth?

Function Queries the total number of points of the waveform specified by `:WAVeform:TRACe`.

Syntax `:WAVeform:LENGth?`

Example `:WAVEFORM:LENGTH? -> 1002`

Description The number of data points is fixed. "1002" is always returned.

:WAVeform:SEND?

Function Queries the waveform display data specified by ":WAVeform:TRACe".

Syntax `:WAVeform:SEND?`

Example
- When ":WAVeform:FORMat" is set to {ASCii}
  `:WAVEFORM:SEND? -> <NR3>,<NR3>,...`
- When ":WAVeform:FORMat" is set to {FLOat}
  `:WAVEFORM:SEND? -> #4`(number of bytes, 4 digits)(data byte sequence)

Description
- The format of the waveform display data that is output varies depending on the ":WAVeform:FORMat" setting as follows:
  (1) When "ASCii" is specified
  The physical value is output in the <NR3> format. The data of each point is delimited by a comma.
  (2) When "FLOat" is specified
  The physical value is output in IEEE single-precision floating point (4-byte) format.
  The output byte order of the data of each point follows the order that is set using the ":WAVeform:BYTeorder" command.
- If there is no waveform display data even when the display mode (:DISPlay:MODE) is set to a mode to display waveforms, the data is output as follows:
  (1) When "ASCii" is specified
  The data of all points are output as "NAN."
  (2) When "FLOat" is specified
  The data of all points are output as "0(0x00000000)."

:WAVeform:SRATe?

Function Queries the sample rate of the retrieved waveform.

Syntax `:WAVeform:SRATe?`

Example `:WAVEFORM:SRATE? -> 200.000E+03`

:WAVeform:STARt

Function Sets the output start point of the waveform display data that is transmitted by ":WAVeform:SEND?" or queries the current setting.

Syntax `:WAVeform:STARt {<NRf>}`
`:WAVeform:STARt?`
<NRf> = 0 to 1001

Example `:WAVEFORM:START 0`
`:WAVEFORM:START? -> :WAVEFORM:START 0`

:WAVeform:TRACe

Function Sets the target waveform for ":WAVeform:SEND?" or queries the current setting.

Syntax `:WAVeform:TRACe {U<x>|I<x>|SPEed|TORQue|MATH<x>}`
`:WAVeform:TRACe?`
<x> of U<x>, I<x> = 1 to 4 (element)
<x> of MATH<x> = 1 to 2 (MATH)

Example `:WAVEFORM:TRACE U1`
`:WAVEFORM:TRACE? -> :WAVEFORM:TRACE U1`

Description {SPEed|TORQue} are valid only on the motor version (-MV).

:WAVeform:TRIGger?

Function Queries the trigger position of the retrieved waveform.

Syntax `:WAVeform:TRIGger?`

Example `:WAVEFORM:TRIGGER? -> 0`

Description Since the trigger position is always at the beginning of the waveform display data, "0" is returned.

6.24 Common Command Group

The commands in the common group are defined in the IEEE488.2-1992 and are independent of the instrument's functions. There are no front panel keys that correspond to the commands in this group.

`*CAL?` (CALibrate)

Function Executes zero calibration (zero-level compensation, same operation as pressing CAL (SHIFT+SINGLE)) and queries the result.

Syntax `*CAL?`

Example `*CAL? -> 0`

Description If the calibration terminates normally, 0 is returned. If an error is detected, 1 is returned.

`*CLS` (CLear Status)

Function Clears the standard event register, extended event register, and error queue.

Syntax `*CLS`

Example `*CLS`

Description
- If the `*CLS` command is located immediately after the program message terminator, the output queue is also cleared.
- For details on the register and queue, see chapter 7.

`*ESE` (standard Event Status Enable register)

Function Sets the standard event enable register or queries the current setting.

Syntax `*ESE {<NRf>}`
`*ESE?`
<NRf> = 0 to 255

Example `*ESE 251`
`*ESE? -> 251`

Description
- Specify the value as a sum of decimal values of each bit.
- For example, specifying "`*ESE 251`" will cause the standard enable register to be set to "11111011." In this case, bit 2 of the standard event register is disabled which means that bit 5 (ESB) of the status byte register is not set to 1, even if a "query error" occurs.
- The default value is "`*ESE 0`" (all bits disabled).
- A query using `*ESE?` will not clear the contents of the standard event enable register.
- For details on the standard event enable register, see page 7-5.

`*ESR?` (standard Event Status Register)

Function Queries the standard event register and clears the register.

Syntax `*ESR?`

Example `*ESR? -> 32`

Description
- A sum of decimal values of each bit is returned.
- You can check what type of events occurred when an SRQ is generated.
- For example, if a value of "32" is returned, this indicates that the standard event register is set to "00100000." In this case, you can see that the SRQ occurred due to a "command syntax error."
- A query using `*ESR?` will clear the contents of the standard event register.
- For details on the standard event register, see page 7-5.

`*IDN?` (IDeNtify)

Function Queries the instrument model.

Syntax `*IDN?`

Example `*IDN? ->`
`YOKOGAWA,760304-04-SV,0,F4.01`

Description
- The information is returned in the following form: <Manufacturer>,<Model>,<Serial No.>,<Firmware version>
- <Model> is in the format "model (6 digits)-element configuration (2 digits)-version." The combination of model and element can be one of the following: 760301-01, 760302-02, 760303-03, 760304-04, 760301-10, 760302-20, 760303-30, 760304-40. The version name is SV for the standard version and MV for the motor version.
- In actuality, <Serial No.> is not returned (always 0).

*OPC (OPeration Complete)

Function Sets bit 0 (OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.

Syntax `*OPC`

Example `*OPC`

Description
- For the description regarding how to synchronize the program using `*OPC`, see page 5-8.
- The "`COMMunicate:OPSE`" command is used to specify the overlap command.
- If `*OPC` is not the last command of the message, the operation is not guaranteed.

*OPC? (OPeration Complete)

Function ASCII code "1" is returned when the specified overlap command is completed.

Syntax `*OPC?`

Example `*OPC? -> 1`

Description
- For the description regarding how to synchronize the program using `*OPC?`, see page 5-8.
- The "`COMMunicate:OPSE`" command is used to specify the overlap command.
- If `*OPC?` is not the last command of the message, the operation is not guaranteed.

*OPT? (OPTion)

Function Queries the installed options.

Syntax `*OPT?`

Example `*OPT? ->G6,B5,DT,FQ,DA,V1,C2,C7,C5,CC,FL`

Description
- The presence or absence of the following is returned: harmonic measurement (G5), harmonic computation function (G6), built-in printer (B5), delta computation (DT), frequency measurement addition (FQ), 20chDA output (DA), VGA output (V1), RS-232 communications (C2), USB port (for PC, C12), Ethernet (C7), USB port (for peripherals, C5), Cycle by Cycle function (CC), and flicker measurement (FL).
- If none of the options is installed, an ASCII code "0" is returned.
- The `*OPT?` query must be the last query of the program message. An error occurs if there is a query after this query.

*PSC (Power-on Status Clear)

Function Sets whether to clear the registers below at power on or queries the current setting. The register is cleared when the value rounded to an integer is a non-zero value.
- Standard event enable register
- Extended event enable register
- Transition filter

Syntax `*PSC {<NRf>}`
`*PSC?`
<NRf> = 0 (not clear), non-zero (clear)

Example `*PSC 1`
`*PSC? -> 1`

Description For details on the registers, see chapter 7.

*RST (ReSeT)

Function Initializes the settings.

Syntax `*RST`

Example `*RST`

Description
- Also clears `*OPC` and `*OPC?` commands that have been sent earlier.
- All settings except communication settings are reset to factory default values.

*SRE (Service Request Enable register)

Function Sets the service request enable register or queries the current setting.

Syntax `*SRE {<NRf>}`
`*SRE?`
<NRf> = 0 to 255

Example `*SRE 239`
`*SRE? -> 175`(since the bit 6 (MSS) setting is ignored)

Description
- Specify the value as a sum of decimal values of each bit.
- For example, specifying "`*SRE 239`" will cause the service request enable register to be set to "11101111." In this case, bit 4 of the service request enable register is disabled which means that bit 4 (MAV) of the status byte register is not set to 1, even if "the output queue is not empty."
- Bit 6 (MSS) of the status byte register is the MSS bit itself, and therefore, is ignored.
- The default value is "`*SRE 0`" (all bits disabled).
- A query using `*SRE?` will not clear the contents of the service request enable register.
- For details on the service request enable register, see page 7-3.

*STB? (STatus Byte)

Function Queries the status byte register.

Syntax `*STB?`

Example `*STB? -> 4`

Description
- The sum of the bits is returned as a decimal value.
- Since the register is read without executing serial polling, bit 6 is a MSS bit not RQS.
- For example, if a value of 4 is returned, this indicates that the status byte register is set to "00000100." In this case, you can see that "the error queue is not empty" (an error occurred).
- A query using `*STB?` will not clear the contents of the status byte register.
- For details on the status byte register, see page 7-3.

*TRG (TRiGger)

Function Executes single measurement (the same operation as when SINGLE is pressed).

Syntax `*TRG`

Example `*TRG`

Description The multi-line message GET (Group Execute Trigger) also performs the same operation as this command.

*TST? (TeST)

Function Performs a self-test and queries the result.

Syntax `*TST?`

Example `*TST? -> 0`

Description
- The self-test involves internal memory tests.
- "0" is returned if the self-test is successful, "1" if it is not.
- It takes approximately 90 s for the test to complete. When receiving a response from the WT3000, set the timeout to a relatively large value.

*WAI (WAIt)

Function Holds the subsequent command until the completion of the specified overlap operation.

Syntax `*WAI`

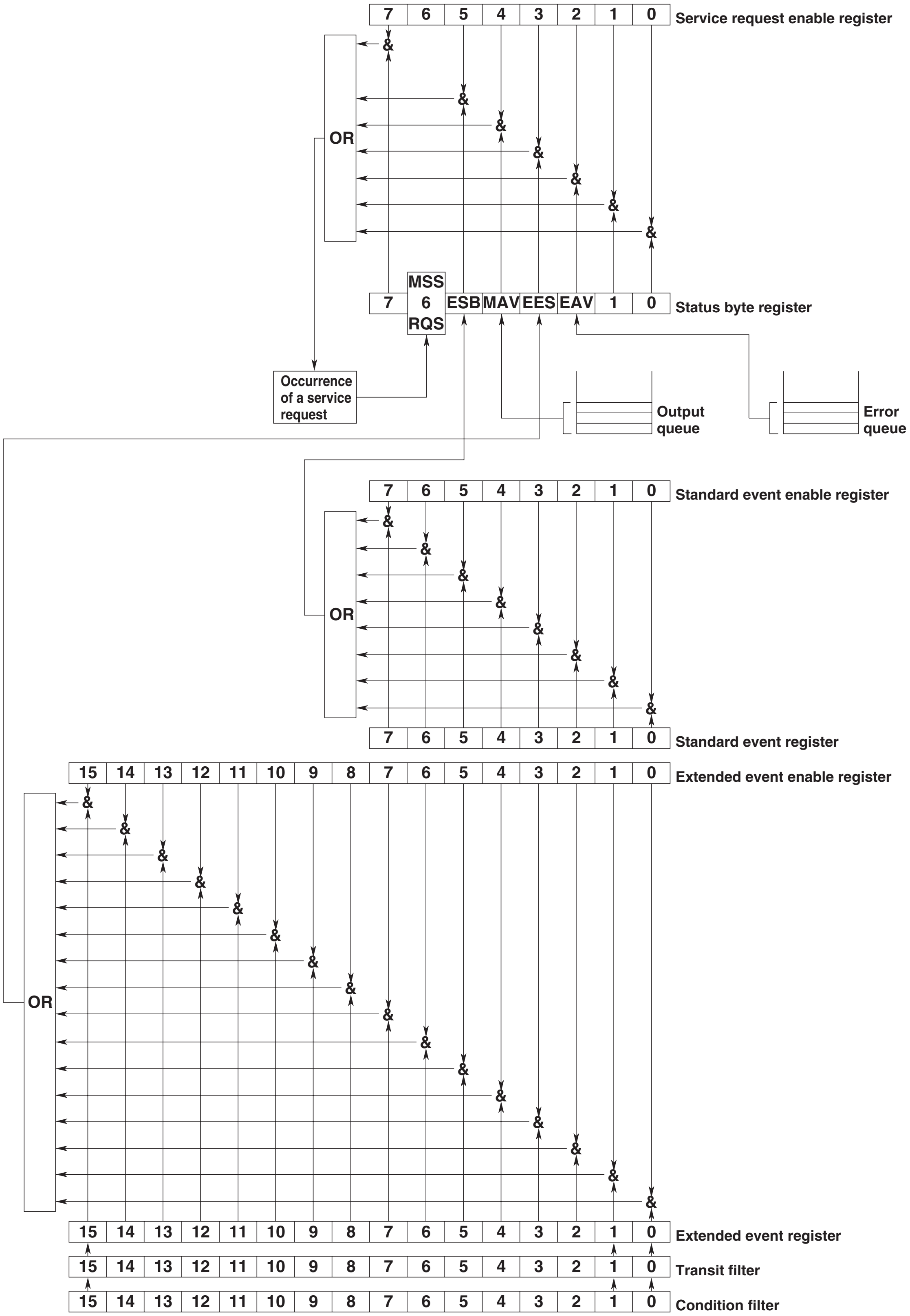
Example `*WAI`

Description
- For the description regarding how to synchronize the program using `*WAI`, see page 5-7.
- The "`COMMunicate:OPSE`" command is used to specify the overlap command.

7.1 Status Reports

Status Reports

The figure below shows the status report that is read by serial polling. This status report is an extended version of the status report defined in IEEE 488.2-1992.



Overview of the Registers and Queues

Name (Function)	Writing	Reading
Status byte	–	Serial polling (RQS) *STB?(MSS)
Service request enable register (Status byte mask)	*SRE	*SRE?
Standard event register (Changes in device status)	–	*ESR?
Standard event enable register (Standard event register mask)	*ESE	*ESE?
Extended event register (Changes in device status)	–	:STATus:EESR?
Extended event enable register (Extended event register mask)	:STATus:EESE	:STATus:EESE?
Condition register (Current device status)	–	:STATus:CONDition?
Transition filter (Conditions that change the extended event register)	:STATus:FILTer<x>	:STATus:FILTer<x>?
Output queue (Stores a response message to a query)	All query commands	–
Error queue (Stores the error No. and message)	–	:STATus:ERRor?

Registers and Queues That Affect the Status Byte

Registers that affect the bits of the status byte are shown below.

Standard Event Register

Sets bit 5 (ESB) of the status byte to 1 or 0.

Output Queue

Sets bit 4 (MAV) of the status byte to 1 or 0.

Extended Event Register

Sets bit 3 (EES) of the status byte to 1 or 0.

Error Queue

Sets bit 2 (EAV) of the status byte to 1 or 0.

Enable Registers

Registers that are used to mask a bit so that the bit will not affect the status byte even when it is set to 1, are shown below.

Status Byte

Mask the bits using the service request enable register.

Standard Event Register

Mask the bits using the standard event enable register.

Extended Event Register

Mask the bits using the extended event enable register.

Writing/Reading from Registers

The *ESE command is used to set the bits in the standard event enable register to 1’s or 0’s. The *ESE? command is used to query whether the bits in the standard event enable register are 1’s or 0’s. For details regarding these commands, see chapter 6.

7.2 Status Byte

Status Byte

	RQS						
7	6	ESB	MAV	EES	EAV	1	0
	MSS						

Bits 0, 1, and 7

Not used (always 0)

Bit 2 EAV (Error Available)

Set to 1 when the error queue is not empty. In other words, this bit is set to 1 when an error occurs. See the page 7-9.

Bit 3 EES (Extend Event Summary Bit)

Set to 0 when the logical product of the extended event register and the corresponding enable register is 1. In other words, this bit is set to 1 when an event takes place inside the instrument. See the page 7-7.

Bit 4 MAV (Message Available)

Set to 1 when the output queue is not empty. In other words, this bit is set to 1 when there are data to be transmitted. See the page 7-9.

Bit 5 ESB (Event Summary Bit)

Set to 0 when the logical product of the standard event register and the corresponding enable register is 1. In other words, this bit is set to 1 when an event takes place inside the instrument. See the page 7-5.

Bit 6 RQS (Request Service)/MSS (Master Status Summary)

Set to 1 when the logical AND of the status byte excluding Bit 6 and the service request enable register is not 0. In other words, this bit is set to 1 when the instrument is requesting service from the controller.

RQS is set to 1 when the MSS bit changes from 0 to 1, and cleared when serial polling is carried out or when the MSS bit changes to 0.

Bit Masking

If you wish to mask a certain bit of the status byte so that it does not cause a SRQ, set the corresponding bit of the service request enable register to 0. For example, to mask bit 2 (EAV) so that service is not requested when an error occurs, set bit 2 of the service request enable register to 0. This can be done using the `*SRE` command. To query whether each bit of the service request enable register is 1 or 0, use `*SRE?`. For details on the `*SRE` command, see chapter 6.

Operation of the Status Byte

A service request is issued when bit 6 of the status byte becomes 1. Bit 6 is set to 1 when any of the other bits becomes a 1 (when the corresponding bit of the service request enable register is also set to 1).For example, if an event occurs and any of the bits of the logical AND of the standard event register and the corresponding enable register becomes a 1, then bit 5 (ESB) is set to 1. In this case, if bit 5 of the service request enable register is 1, bit 6 (MSS) is set to 1, thus requesting service from the controller. It is also possible to check what type of event has occurred by reading the contents of the status byte.

Reading from the Status Byte

The following two methods are provided for reading the status byte.

Inquiry Using the *STB? Query

Making an inquiry using the `*STB?` query sets bit 6 to MSS. This causes the MSS to be read. After completion of the read-out, none of the bits in the status byte will be cleared.

Serial Polling

Execution of a serial polling changes bit 6 to RQS. This causes RQS to be read. After completion of the read-out, only RQS is cleared. It is not possible to read MSS using serial polling.

Clearing the Status Byte

No method is provided for forcibly clearing all the bits in the status byte. The bits that are cleared for each operation are shown below.

When a Query Is Made Using the *STB? Command

No bits are cleared.

When a Serial Polling Is Executed

Only the RQS bit is cleared.

When a *CLS Command Is Received

When the `*CLS` command is received, the status byte itself is not cleared, but the contents of the standard event register (which affects the bits in the status byte) are cleared. As a result, the corresponding bits in the status byte are cleared, except bit 4 (MAV), since the output queue cannot be emptied by the `*CLS` command. However, the output queue is also cleared if the `*CLS` command is received just after a program message terminator.

7.3 Standard Event Register

Standard Event Register

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Bit 7 PON (Power ON)
Set to 1 when the power is turned ON.

Bit 6 URQ (User Request)
Not used (always 0)

Bit 5 CME (Command Error)
Set to 1 when the command syntax is incorrect.

Example Received a command name with a spelling error or character data not in the selection.

Bit 4 EXE (Execution Error)
Set to 1 when the command syntax is correct but the command cannot be executed in the current state.

Example Received a command with a parameter outside the range or a command dealing with an unsupported option.

Bit 3 DDE (Device Error)
Set to 1 when execution of the command is not possible due to an internal problem in the instrument that is not a command error or an execution error.

Bit 2 QYE (Query Error)
Set to 1 if the output queue is empty or if the data is missing even after a query has been sent.

Example No response data; data is lost due to an overflow in the output queue.

Bit 1 RQC (Request Control)
Not used (always 0)

Bit 0 OPC (Operation Complete)
Set to 1 when the operation designated by the `*OPC` command (see chapter 6) has been completed.

Bit Masking

If you wish to mask a certain bit of the standard event register so that it does not cause bit 5 (ESB) of the status byte to change, set the corresponding bit of the standard event enable register to 0. For example, to mask bit 2 (QYE) so that ESB is not set to 1, even if a query error occurs, set bit 2 of the standard event enable register to 0. This can be done using the `*ESE` command. To query whether each bit of the standard event enable register is 1 or 0, use the `*ESE?`. For details on the `*ESE` command, see chapter 6.

Operation of the Standard Event Register

The standard event register is provided for eight different kinds of event which can occur inside the instrument. Bit 5 (ESB) of the status byte is set to 1 when any of the bits in this register becomes 1 (or when the corresponding bit of the standard event enable register becomes 1).

Example

1. A query error occurs.
2. Bit 2 (QYE) is set to 1.
3. Bit 5 (ESB) of the status byte is set to 1 if bit 2 of the standard event enable register is 1.

It is also possible to check what type of event has occurred inside the instrument by reading the contents of the standard event register.

Reading from the Standard Event Register

The contents of the standard event register can be read by the `*ESR?` command. After the register is read, it is cleared.

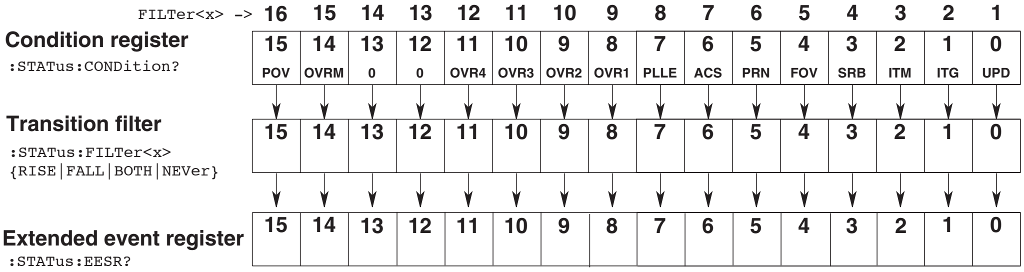
Clearing the Standard Event Register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read using the `*ESR?` command.
- When a `*CLS` Command Is Received
- When the instrument is power cycled.

7.4 Extended Event Register

Reading the extended event register tells you whether changes in the condition register (reflecting internal conditions) have occurred. A transition filter can be applied which allows you to decide which events are reported to the extended event register.



FILTer<x> ->	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Condition register :STATus:CONDition?	15 POV	14 OVRM	13 0	12 0	11 OVR4	10 OVR3	9 OVR2	8 OVR1	7 PLLE	6 ACS	5 PRN	4 FOV	3 SRB	2 ITM	1 ITG	0 UPD
Transition filter :STATus:FILTer<x> {RISE|FALL|BOTH|NEVer}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Extended event register :STATus:EESR?	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The meaning of each bit of the condition register is as follows:

Bit 0 UPD (Updating)
Set to 1 when the measured data is being updated. The falling edge of UPD (1 -> 0) signifies the end of the updating.

Bit 1 ITG (Integrate Busy)
Set to 1 while integration is in progress.

Bit 2 ITM (Integrate Timer Busy)
Set to 1 while the integration timer is running.

Bit 3 SRB (Store/Recall Busy)
Set to 1 while storing or recalling data.

Bit 4 FOV (Frequency Over)
Set to 1 when the frequency is in error.

Bit 5 PRN (Printing)
Set to 1 while the built-in printer is in operation or while data is being output to a network printer.

Bit 6 ACS (Accessing)
Set to 1 when the PC card interface, USB storage medium, or network drive is being accessed.

Bit 7 PLLE (PLL Source Input Error)
Set to 1 when there is no input to the PLL source and synchronization cannot be achieved during harmonic measurement.

Bit 8 OVR1(Element1 Measured Data Over)
Set to 1 when the voltage or current of element 1 is overrange.

Bit 9 OVR2(Element2 Measured Data Over)
Set to 1 when the voltage or current of element 2 is overrange.

Bit 10 OVR3 (Element3 Measured Data Over)
Set to 1 when the voltage or current of element 3 is overrange.

Bit 11 OVR4 (Element4 Measured Data Over)
Set to 1 when the voltage or current of element 4 is overrange.

Bit 14 OVRM (Motor Measured Data Over)
Set to 1 when the speed or torque of the motor input is overrange.

Bit 15 POV (ElementX Input Peak Over)
Set to 1 when peak over (input exceeding the peak) is detected in any of the elements.

The transition filter parameters detect changes in the specified bit (numerical suffix, 1 to 16) of the condition register in the following manner and overwrite the extended event register.

RISE	The specified bit of the extended event register is set to 1 when the bit of the condition register changes from 0 to 1.
FALL	The specified bit of the extended event register is set to 1 when the bit of the condition register changes from 1 to 0.
BOTH	The specified bit of the extended event register is set to 1 when the bit of the condition register changes from 0 to 1 or 1 to 0.
NEVer	Always 0.

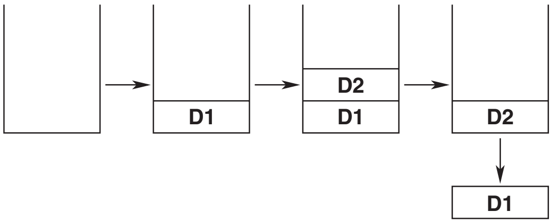
7.5 Output Queue and Error Queue

Output Queue

The output queue is provided to store response messages to queries. For example, if you send the WAVeform:SEND? command, which requests the output of acquired data, the data is stored in the output queue until it is read. As shown below, data are stored in order and read from the oldest ones first. The output queue is emptied in the following cases (in addition to when read-out is performed).

- When a new message is received from the controller.
- When a deadlock occurs (see page 5-2).
- When a device clear command (DCL or SDC) is received.
- When the instrument is power cycled.

The output queue cannot be emptied using the *CLS command. To see whether the output queue is empty or not, check bit 4 (MAV) of the status byte.



Error Queue

The error queue stores the error No. and message when an error occurs. For example, if the controller sends an incorrect program message, the error number and message "113, "Undefined header"" are stored in the error queue when the error is displayed. The contents of the error queue can be read using the STATus:ERRor? query. As with the output queue, the messages are read from the oldest ones first. When the error queue overflows, the last message is replaced by the message "350, "Queue overflow"." The error queue is emptied in the following cases (in addition to when read-out is performed).

- When a *CLS command is received
- When the instrument is power cycled.

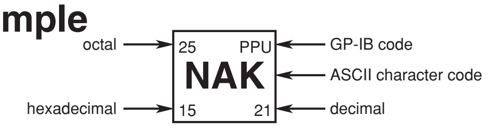
To see whether the error queue is empty or not, check bit 2 (EAV) of the status byte.

Appendix 1 ASCII Character Codes

The following table shows the ASCII character codes.

	0	1	2	3	4	5	6	7
0	0 NUL 0 0	20 DEL 10 16	40 0 SP 20 32	60 16 0 30 48	100 0 @ 40 64	120 16 P 50 80	140 0 ‘ 60 96	160 16 p 70 112
1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 1 ! 21 33	61 17 1 31 49	101 1 A 41 65	121 17 Q 51 81	141 1 a 61 97	161 17 q 71 113
2	2 STX 2 2	22 DC2 12 18	42 2 ” 22 34	62 18 2 32 50	102 2 B 42 66	122 18 R 52 82	142 2 b 62 98	162 18 r 72 114
3	3 ETX 3 3	23 DC3 13 19	43 3 # 23 35	63 19 3 33 51	103 3 C 43 67	123 19 S 53 83	143 3 c 63 99	163 19 s 73 115
4	4 SDC EOT 4 4	24 DCL DC4 14 20	44 4 $ 24 36	64 20 4 34 52	104 4 D 44 68	124 20 T 54 84	144 4 d 64 100	164 20 t 74 116
5	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 5 % 25 37	65 21 5 35 53	105 5 E 45 69	125 21 U 55 85	145 5 e 65 101	165 21 u 75 117
6	6 ACK 6 6	26 SYN 16 22	46 6 & 26 38	66 22 6 36 54	106 6 F 46 70	126 22 V 56 86	146 6 f 66 102	166 22 v 76 118
7	7 BEL 7 7	27 ETB 17 23	47 7 ’ 27 39	67 23 7 37 55	107 7 G 47 71	127 23 W 57 87	147 7 g 67 103	167 23 w 77 119
8	10 GET BS 8 8	30 SPE CAN 18 24	50 8 (28 40	70 24 8 38 56	110 8 H 48 72	130 24 X 58 88	150 8 h 68 104	170 24 x 78 120
9	11 TCT HT 9 9	31 SPD EM 19 25	51 9) 29 41	71 25 9 39 57	111 9 I 49 73	131 25 Y 59 89	151 9 i 69 105	171 25 y 79 121
A	12 LF A 10	32 SUB 1A 26	52 10 * 2A 42	72 26 : 3A 58	112 10 J 4A 74	132 26 Z 5A 90	152 10 j 6A 106	172 26 z 7A 122
B	13 VT B 11	33 ESC 1B 27	53 11 + 2B 43	73 27 ; 3B 59	113 11 K 4B 75	133 27 [5B 91	153 11 k 6B 107	173 27 { 7B 123
C	14 FF C 12	34 FS 1C 28	54 12 , 2C 44	74 28 < 3C 60	114 12 L 4C 76	134 28 \ 5C 92	154 12 l 6C 108	174 28 | 7C 124
D	15 CR D 13	35 GS 1D 29	55 13 - 2D 45	75 29 = 3D 61	115 13 M 4D 77	135 29] 5D 93	155 13 m 6D 109	175 29 } 7D 125
E	16 SO E 14	36 RS 1E 30	56 14 . 2E 46	76 30 > 3E 62	116 14 N 4E 78	136 30 ^ 5E 94	156 14 n 6E 110	176 30 ~ 7E 126
F	17 SI F 15	37 US 1F 31	57 15 / 2F 47	77 UNL ? 3F 63	117 15 O 4F 79	137 UNT _ 5F 95	157 15 o 6F 111	177 DEL (RUBOUT) 7F 127
	Address Command	Universal Command	Listener Address		Talker Address		Secondary Command	

Example



Appendix 2 Error Messages

This section explains the communication error messages.

- The messages can be displayed in English or Japanese on the WT3000. However, if a messages is queried with the `:STATus:ERRor?` command and displayed on a PC, the message is displayed in English.
- If servicing is required, contact your nearest YOKOGAWA dealer for repairs.
- Only error messages related to communications are listed here. For other error messages, see *User's Manual IM760301-01E*.

Error in Communication Command (100-199)

Code	Messages	Corrective Action	Page
102	Syntax error	Invalid syntax.	Chapter 5, 6
103	Invalid separator	Use a comma to separate the data.	5-1
104	Data type error	See pages 5-5 and 5-6 and write using the correct data form.	5-5 and 5-6
108	Parameter not allowed	Check the number of data points.	5-5, chapter 6
109	Missing parameter	Enter the required data.	5-5, chapter 6
111	Header separator error	Use a space to separate the header and data.	5-1
112	Program mnemonic too long	Check the mnemonic (alphanumerical character string).	Chapter 6
113	Undefined header	Check the header.	Chapter 6
114	Header suffix out of range	Check the header.	Chapter 6
120	Numeric data error	A number is required in the <NRf> form.	5-5
123	Exponent too large	Use a smaller exponent for <NR3> format.	5-5, chapter 6
124	Too many digits	The value must be less than equal to 255 digits.	5-5, chapter 6
128	Numeric data not allowed	Enter in a format other than <NRf> format.	5-5, chapter 6
131	Invalid suffix	Check the unit of the <Voltage>, <Current>, <Time>, and <Frequency>.	5-5
134	Suffix too long	Check the unit of the <Voltage>, <Current>, <Time>, and <Frequency>.	5-5
138	Suffix not allowed	No units are allowed other than <Voltage>, <Current>, <Time>, and <Frequency>.	5-5
141	Invalid character data	Select character data from the selections available in `{...|...|...}`.	Chapter 6
144	Character data too long	Check the spelling of the character strings in `{...|...|...}`.	Chapter 6
148	Character data not allowed	Write in a data form other than `{...|...|...}`.	Chapter 6
150	String data error	Enclose <String> in double quotation or single quotation marks.	5-6
151	Invalid string data	<String> is too long or contains characters which cannot be used.	Chapter 6
158	String data not allowed	Enter in a data format other than <String>.	Chapter 6

Code	Messages	Corrective Action	Page
161	Invalid block data	<Block data> is not allowed.	5-6, chapter 6
168	Block data not allowed	<Block data> is not allowed.	5-6, chapter 6
171	Invalid expression	Equations cannot be used.	Chapter 6
178	Expression data not allowed	Equations cannot be used.	Chapter 6
181	Invalid outside macro definition	Macro functions defined in IEEE488.2 are not supported.	–

Error in Communication Execution (200 to 299)

Code	Messages	Corrective Action	Page
221	Setting conflict	Check the relevant settings.	Chapter 6
222	Data out of range	Check the range.	Chapter 6
223	Too much data	Check the length of the data.	Chapter 6
224	Illegal parameter value	Check the range.	Chapter 6
241	Hardware missing	Check the installed options.	–
260	Expression error	Equations cannot be used.	–
270	Macro error	Macro functions defined in IEEE488.2 are not supported.	–
272	Macro execution error	Macro functions defined in IEEE488.2 are not supported.	–
273	Illegal macro label	Macro functions defined in IEEE488.2 are not supported.	–
275	Macro definition too long	Macro functions defined in IEEE488.2 are not supported.	–
276	Macro recursion error	Macro functions defined in IEEE488.2 are not supported.	–
277	Macro redefinition not allowed	Macro functions defined in IEEE488.2 are not supported.	–
278	Macro header not found	Macro functions defined in IEEE488.2 are not supported.	–

Error in Communication Query (400 to 499)

Code	Messages	Corrective Action	Page
410	Query INTERRUPTED	Check transmission/reception order.	5-2
420	Query UNTERMINATED	Check transmission/reception order.	5-2
430	Query DEADLOCKED	Limit the length of the program message including <PMT> to 1024 bytes or less.	5-2
440	Query UNTERMINATED after indefinite response	Do not specify a query after the *IDN? or *OPT? command.	–

Error in System Operation (912 to 914)

Code	Messages	Corrective Action	Page
912	Fatal error in Communication-driver	Servicing required.	–

Warning (5)

Code	Messages	Corrective Action	Page
5	*OPC/? exists in message	Place the *OPC or *OPC? command at the end of the program message.	–

Other Errors (350, 390)

Code	Messages	Corrective Action	Page
350	Queue overflow	Read the error queue.	7-9
390	Overrun error (RS-232 only)	Lower the baud rate.	–

Note

Code 350 indicates overflow of error queue. This code is returned as a response to the “STATus:ERRor?” query; it does not appear on the screen.

Appendix 3 IEEE 488.2-1992

The GP-IB interface of the instrument conforms to the IEEE 488.2-1992 Standard. This standard specifies that the following 23 points be stated in the document. This section describes these points.

(1) Of the IEEE 488.1 interface functions, the subsets that are supported
See section 1.3, “GP-IB Interface Specifications.”

(2) The operation of the device when it is assigned an address outside the 0 to 30 range
The address of this instrument cannot be set to an address outside the 0 to 30 range.

(3) Reaction of the device when the user changes the address
The address change occurs when the address is specified using the MISC key menu. The new address is valid until the next time it is changed.

(4) Device settings at power-up. The commands that can be used at power-up.
Basically, the previous settings (i.e. the settings which were valid when power was turned OFF) are valid.
All commands can be used at power-up.

(5) Message exchange options

a) Input buffer size
1024 bytes

b) Queries that return multiple response units
See the example of the commands given in chapter 6.

c) Queries that create response data when the command syntax is being analyzed
All queries create response data when the command syntax is analyzed.

d) Queries that create response data during reception
There are no queries of which the response data are created upon receiving a send request from the controller.

e) Commands that have parameters the restrict one another
See the example of the commands given in chapter 6.

(6) Items that are included in the functional or composite header elements constituting a command
See chapter 5 and 6.

(7) Buffer sizes that affect block data transmission
During block data transmission, the output queue is expanded according to the size.

(8) A list of program data elements that can be used in equations and their nesting limitations
No equations can be used.

(9) Syntax of the responses to queries
See the example of the commands given in chapter 6.

(10) Communication between devices that do not follow the response syntax
None.

(11) Size of the response data block
1 to 308922 bytes

(12) A list of supported common commands
See section 6.24, “Common Command Group.”

(13) Device condition after a successful calibration
Measurement execution condition.

(14) The maximum length of block data that can be used for the *DDT trigger macro definition
Not supported.

(15) The maximum length of the macro label for defining macros, the maximum length of block data that can be used for the macro definition, and the process when recursion is used in macro definitions
Macro functions are not supported.

(16) Reply to the IDN? query
See section 6.24, “Common Command Group.”

(17) The size of the storage area for protected user data for *PUD and *PUD? commands
*PUD and *PUD? are not supported.

(18) The length of the *RDT and *RDT? resource names
*RDT and *RDT? are not supported.

(19) The change in the status due to *RST, *LRN?, *RCL, and *SAV
*RST
See section 6.24, “Common Command Group.”

*LRN?, *RCL, and *SAV
These common commands are not supported.

(20) The extent of the self-test using the *TST? command
Executes all the MEMORY tests (each internal memory) of the Self Test menu of the MISC key.

(21) The structure of the extended return status
See chapter 7.

(22) Whether each command is processed in an overlap fashion or sequentially
See section 5.5, “Synchronization with the Controller” and chapter 6.

(23) The description of the execution of each command
See the functions of each command in chapter 6, *User’s Manual IM760301-01E,* and *Expansion Function User’s Manual IM760301-51E*.

Index