PZ4000
Power Analyzer
Communication Interface
USER'S MANUAL
Introduction

Thank you for purchasing YOKOGAWA's PZ4000 Power Analyzer. This Communication Interface User’s Manual describes the functions and commands of the GP-IB and serial interfaces. To ensure proper use of the GP-IB/serial interfaces, please read this manual thoroughly. Keep the manual in a safe place for quick reference whenever a question arises.

Two manuals are provided with the PZ4000 including this Communication Interface User’s Manual.

<table>
<thead>
<tr>
<th>Manual Name</th>
<th>Manual No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PZ4000 Power Analyzer User’s Manual</td>
<td>IM 253710-01E</td>
<td>Describes all functions except for the communications functions and operation procedures of the instrument.</td>
</tr>
</tbody>
</table>

Note

• The contents of this manual are subject to change without prior notice as a result of improvements in instrument’s performance and functions.
• Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA representative listed on the back cover of this manual.
• Copying or reproduction of all or any part of the contents of this manual without YOKOGAWA’s permission is strictly prohibited.

Trademarks

• IBM PC/AT is a registered trademark of International Business Machines Corporation.
• Other product names are trademarks or registered trademarks of their respective holders.

Revisions

1st Edition: April 1999
2nd Edition: April 2000
How to Use this Manual

Structure of this Manual
This User’s Manual consists of five chapters, an Appendix and an Index as described below.

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

Chapter 2 Overview of the Serial Interface
Describes the functions and specifications of serial.

Chapter 3 Before Programming
Describes formats used when sending a command.

Chapter 4 Command
Describes each command.

Chapter 5 Status Report
Describes the status byte, various registers and queues.

Chapter 6 Sample Programs
Sample programs, written in Quick-BASIC, for MS-DOS/V machines equipped with the following GP-IB board: AT-GPIB/TNT IEEE-488.2, from National Instruments.

Appendix
Contains references including the ASCII character code table.

Index
Provides an alphabetically ordered index.

Conventions Used in this Manual

• Symbols used for Notes and Keys

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>k</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e.g.: 640 KB (floppy disk memory capacity)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Provides information that is necessary for proper operation of the instrument.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>[Comm Device]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Refers to a soft key displayed on the screen.</td>
</tr>
</tbody>
</table>

• Symbols used in syntax descriptions
Symbols which are used in the syntax descriptions in Chapter 4 are shown below. These symbols are referred to as

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
<th>Example of Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&gt;</td>
<td>Defined value</td>
<td>CHANnel &lt;&lt;&gt; &lt;&gt; &gt;=1 to 8</td>
<td>--&gt;CHANNEL2</td>
</tr>
<tr>
<td>{}</td>
<td>One of the options in {} is selected.</td>
<td>COUPling {AC</td>
<td>DC</td>
</tr>
<tr>
<td>[]</td>
<td>Exclusive OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[]</td>
<td>Abbreviated</td>
<td>TRIGger [:SIMple]:SLOPe</td>
<td>--&gt;TRIGger:SLOPer</td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Overview of the GP-IB Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1 Names of the Parts and Their Functions</td>
</tr>
<tr>
<td></td>
<td>1.2 Connecting the GP-IB Cable</td>
</tr>
<tr>
<td></td>
<td>1.3 GP-IB Interface Functions</td>
</tr>
<tr>
<td></td>
<td>1.4 GP-IB Interface Specifications</td>
</tr>
<tr>
<td></td>
<td>1.5 Setting Addressable Mode</td>
</tr>
<tr>
<td></td>
<td>1.6 Response to Interface Messages</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Overview of the Serial Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.1 Names of the Parts and Their Functions</td>
</tr>
<tr>
<td></td>
<td>2.2 Serial Interface Functions and Specifications</td>
</tr>
<tr>
<td></td>
<td>2.3 Connecting the Serial Interface Cable</td>
</tr>
<tr>
<td></td>
<td>2.4 Handshaking</td>
</tr>
<tr>
<td></td>
<td>2.5 Matching the Data Format</td>
</tr>
<tr>
<td></td>
<td>2.6 Setting up this Instrument</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Before Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.1 Messages</td>
</tr>
<tr>
<td></td>
<td>3.2 Commands</td>
</tr>
<tr>
<td></td>
<td>3.3 Response</td>
</tr>
<tr>
<td></td>
<td>3.4 Data</td>
</tr>
<tr>
<td></td>
<td>3.5 Synchronization with the Controller</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.1 Command Listing</td>
</tr>
<tr>
<td></td>
<td>4.2 ABORT Group</td>
</tr>
<tr>
<td></td>
<td>4.3 ACQuire Group</td>
</tr>
<tr>
<td></td>
<td>4.4 CHANnel Group</td>
</tr>
<tr>
<td></td>
<td>4.5 COMMunicate Group</td>
</tr>
<tr>
<td></td>
<td>4.6 CURSor Group</td>
</tr>
<tr>
<td></td>
<td>4.7 DISPLAY Group</td>
</tr>
<tr>
<td></td>
<td>4.8 FILE Group</td>
</tr>
<tr>
<td></td>
<td>4.9 HCOPy Group</td>
</tr>
<tr>
<td></td>
<td>4.10 IMAGe Group</td>
</tr>
<tr>
<td></td>
<td>4.11 INPut Group</td>
</tr>
<tr>
<td></td>
<td>4.12 MATH Group</td>
</tr>
<tr>
<td></td>
<td>4.13 MEASure Group</td>
</tr>
<tr>
<td></td>
<td>4.14 NULL Group</td>
</tr>
<tr>
<td></td>
<td>4.15 NUMeric Group</td>
</tr>
<tr>
<td></td>
<td>4.16 SETup Group</td>
</tr>
<tr>
<td></td>
<td>4.17 SSTart Group</td>
</tr>
<tr>
<td></td>
<td>4.18 START Group</td>
</tr>
<tr>
<td></td>
<td>4.19 STATus Group</td>
</tr>
<tr>
<td></td>
<td>4.20 STOP Group</td>
</tr>
<tr>
<td></td>
<td>4.21 SYSTem Group</td>
</tr>
</tbody>
</table>
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.22</td>
<td>TIMebase Group</td>
<td>4-76</td>
</tr>
<tr>
<td>4.23</td>
<td>TRIGger Group</td>
<td>4-77</td>
</tr>
<tr>
<td>4.24</td>
<td>WAveform Group</td>
<td>4-80</td>
</tr>
<tr>
<td>4.25</td>
<td>ZOOM Group</td>
<td>4-83</td>
</tr>
<tr>
<td>4.26</td>
<td>Common Command Group</td>
<td>4-85</td>
</tr>
</tbody>
</table>

Chapter 5  Status Report

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Overview of the Status Report</td>
<td>5-1</td>
</tr>
<tr>
<td>5.2</td>
<td>Status Byte</td>
<td>5-2</td>
</tr>
<tr>
<td>5.3</td>
<td>Standard Event Register</td>
<td>5-3</td>
</tr>
<tr>
<td>5.4</td>
<td>Extended Event Register</td>
<td>5-4</td>
</tr>
<tr>
<td>5.5</td>
<td>Output Queue and Error Queue</td>
<td>5-5</td>
</tr>
</tbody>
</table>

Chapter 6  Sample Program

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Before Programming</td>
<td>6-1</td>
</tr>
<tr>
<td>6.2</td>
<td>Example of Normal Measurement Data Output</td>
<td>6-2</td>
</tr>
<tr>
<td>6.3</td>
<td>Example of Harmonic Measurement Data Output</td>
<td>6-5</td>
</tr>
<tr>
<td>6.4</td>
<td>Output Example of Waveform Data in ASCII Format</td>
<td>6-7</td>
</tr>
<tr>
<td>6.5</td>
<td>Output Example of Waveform Data in Binary Format</td>
<td>6-9</td>
</tr>
</tbody>
</table>

Appendix

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASCII Character Code</td>
<td>App-1</td>
</tr>
<tr>
<td>2</td>
<td>Error Messages</td>
<td>App-2</td>
</tr>
<tr>
<td>3</td>
<td>Overview of IEEE 488.2-1987</td>
<td>App-4</td>
</tr>
</tbody>
</table>

Index
1.1 Names of the Parts and Their Functions

Front Panel

REMOTE indicator
Lit while remote mode is active via communications.

LOCAL key
Press this key to switch from remote mode to local mode so as to enable key operation.

MISC key
Press this key to set communication setting.

Rear Panel

GP-IB connector
Used to connect a controller (personal computer etc.) using a GP-IB cable. For information on how to connect the GP-IB cable, refer to the following page.
1.2 Connecting the GP-IB Cable

GP-IB Cable

The GP-IB connector on the side panel of the PZ4000 is a 24-pin connector that conforms to IEEE Standard 488-1978. Use a GP-IB cable that also conforms to IEEE Standard 488-1978.

Connection Method

Connect the GP-IB cable as shown below.

Connection Precautions

- Be sure to tighten the screws on the GP-IB cable connector firmly.
- The instrument can be connected to more than one item of equipment (e.g. a personal computer) if more than one GP-IB cable is used. However, it is not possible to connect more than 15 items of equipment (including the controller) to a single bus.
- If you connect the instrument to more than one item of equipment, make sure that a different address is used for each item.
- Each connecting cable must be 2 m or less in length.
- The total length of all the cables must not exceed 20 m.
- While communications are in progress, more than two-thirds of the connected equipment items must be turned ON.
- When connecting more than one item of equipment, connect them so that the connection route forms a star or linear configuration. Loop or parallel wiring is not allowed.

CAUTION

Be sure to switch off power to both your PC and the oscilloscope before connecting or disconnecting cables. Failure to switch power off may cause internal circuit failure or improper operation.
1.3 GP-IB Interface Functions

GP-IB Interface Functions

Listener function
- Allows you to make the settings which you can make using the panel keys on the instrument, except for the power ON/OFF and GP-IB communications settings.
- Receives commands from a controller requesting output of set-up and waveform data. Also receives status report commands.

Talker function
- Outputs set-up and waveform data.

Note
The talk-only, listen-only and controller functions are not available on this instrument.

Switching between Remote and Local Modes

When switched from Local to Remote Mode
Remote mode is activated when a REN (Remote Enable) message is received from a controller while local mode is active.

- REMOTE is displayed on.
- All front panel keys except the LOCAL key can no longer be operated any more.
- Settings entered in local mode are retained.

When switched from Remote to Local Mode
Pressing the LOCAL key in remote mode puts the instrument in local mode. However, this is not possible if Local Lockout has been set by the controller (page 1-6).

- The REMOTE indicator is turned off.
- All front panel keys are operative.
- Settings entered in remote mode are retained.
1.4 GP-IB Interface Specifications

GP-IB Interface Specifications


Interface functions: Refer to the table below.


Code: ISO (ASCII) code

Mode: Addressable mode

Address setting: Addresses 0 to 30 can be selected from the GP-IB setting screen, displayed when you press the MISC key.

Remote mode clear: Remote mode can be cleared by pressing the LOCAL key. However, this is not possible if Local Lockout has been set by the controller.

<table>
<thead>
<tr>
<th>Function</th>
<th>Subset Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source handshaking</td>
<td>SH1</td>
<td>Full source handshaking capability</td>
</tr>
<tr>
<td>Acceptor handshaking</td>
<td>AH1</td>
<td>Full acceptor handshaking capability</td>
</tr>
<tr>
<td>Talker</td>
<td>T6</td>
<td>Basic talker capability, serial polling, untalk on MLA (My Listen Address), no talk-only capability</td>
</tr>
<tr>
<td>Listener</td>
<td>L4</td>
<td>Basic listener capability, unlisten on MTA (My Talk Address), no listen-only capability</td>
</tr>
<tr>
<td>Service request</td>
<td>SR1</td>
<td>Full service request capability</td>
</tr>
<tr>
<td>Remote local</td>
<td>RL1</td>
<td>Full remote/local capability</td>
</tr>
<tr>
<td>Parallel poll</td>
<td>PP0</td>
<td>No parallel polling capability</td>
</tr>
<tr>
<td>Device clear</td>
<td>DC1</td>
<td>Full device clear capability</td>
</tr>
<tr>
<td>Device trigger</td>
<td>DT1</td>
<td>Device trigger capability</td>
</tr>
<tr>
<td>Controller</td>
<td>C0</td>
<td>No controller function</td>
</tr>
<tr>
<td>Electrical characteristic</td>
<td>E1</td>
<td>Open collector</td>
</tr>
</tbody>
</table>
1.5 Setting Addressable Mode

Before You Begin

When you make settings which can be made using the front panel keys of the instrument or when you output set-up data or waveform data using the controller, the following settings must be made.

Setting the address

This function allows you to set the instrument’s address for addressable mode within the range of 0 to 30. Each item of equipment connected via a GP-IB interface has its own address, by which it can be identified. Care must be taken to ensure that all interconnected devices are assigned unique addresses.

Note

Do not change the address while the GP-IB interface is being used by the controller.

Operation Procedure

1. Press the MISC key.
2. Press the “GP-IB/RS232” soft key.
3. Press the “Comm Device” soft key to select “GPIB.”
4. Turn the jog shuttle to set the desired address.
1.6 Response to Interface Messages

Response to Interface Messages

Response to a uni-line message
IFC (Interface Clear)
Clears the talker and listener. Stops output if data is being output.

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

IDY (Identify) is not supported.

Response to a multi-line message (address command)
GTL (Go To Local)
Switches to local mode.

SDC (Selected Device Clear)
Clears the program message (command) which is currently being output. Also clears the output queue (page 4-5).

*OPC and *OPC? will be disabled if they are currently being executed.

*WAI and COMMunicate:WAIT will be stopped immediately.

GET(Group Execute Trigger)
Operates in the same way as the TRG command.

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

Response to a multi-line message (universal command)
LLO (Local Lockout)
Invalidates the LOCAL key on the front panel to disable switching to local mode.

DCL (Device Clear)
Same as SDC

SPE (Serial Poll Enable)
Sets the talker function to serial poll mode for all equipment connected to the communications bus. The controller performs polling on equipment sequentially.

SPD (Serial Poll Disable)
Clears serial poll mode as the talker function for all equipment connected to the communications bus.

PPU (Parallel Poll Unconfigure) is not supported.

What is an Interface Message?

An interface message is also called an interface command or bus command, and is issued by the controller. Interface messages are classified as follows.

Uni-line messages
Messages are transferred through a single control line. The following three types of uni-line message are available.

IFC (Interface Clear)
REN (Remote Enable)
IDY (Identify)
1.6 Response to Interface Messages

**Multi-line message**
Eight data lines are used to transmit a message. Multi-line messages are classified as follows.

**Address commands**
Valid when the equipment is designated as a listener or a talker. The following five address commands are available.

Commands valid for equipment designated as a listener
GTL (Go To Local)
SDC (Selected Device Clear)
PPC (Parallel Poll Configure)
GET (Group Execute Trigger)

Command valid for equipment designated as a talker
TCT (Take Control)

**Universal commands**
Valid for any item of equipment, irrespective of whether the item is designated as a listener or a talker. The following five universal commands are available.

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

In addition to the above commands, a listener address, talker address on secondary command can be sent in an interface message.

<table>
<thead>
<tr>
<th>Uni-line Messages</th>
<th>Multi-line Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFC, REN, IDY</td>
<td>Address command</td>
</tr>
<tr>
<td></td>
<td>★GTL, ★SDC, PPC, ★GET, ★TCT</td>
</tr>
<tr>
<td></td>
<td>Universal command</td>
</tr>
<tr>
<td></td>
<td>★LLO, ★DCL, PPU, ★SPE, ★SPD</td>
</tr>
</tbody>
</table>

Listeners address
Talker address
Secondary command

Messages marked with a “★” are interface messages supported by the PZ4000

**Note**
Differences between SDC and DCL
The SDC command is an address command and requires that both the talker and listener be designated; however DCL is a universal command and does not require that the talker and listener be designated. Therefore, SDC is used for particular items of equipment, while DCL can be used for any equipment connected to the communications bus.
2.1 Names of the Parts and Their Functions

Front Panel

REMOTE indicator
Lit while remote mode is active via communications.

LOCAL key
Press this key to switch from remote mode to local mode so as to enable key operation.

MISC key
Press this key to set communication setting.

Rear Panel

Serial (RS-232) connector
Used to connect a controller (personal computer etc.) using a serial cable.
For information on how to connect the serial cable, refer to section 2.3.
2.2 Serial Interface Functions and Specifications

Receiving Function
It is possible to make the same settings via the serial interface as can be made using the front panel keys.
Measured/computed data, panel set-up information and error codes can be received.

Sending Function
Measured/computed data can be output.
Panel set-up information and the status byte can be output.
Error codes which have occurred can be output.

Serial Interface Specifications

- **Electrical characteristics**: Complies with EIA-574 Standard (EIA-232 (RS-232) Standard for 9 pin)
- **Connection**: Point-to-point
- **Communications**: Full-duplex
- **Synchronization**: Start-stop system
- **Baud rate**: 1200, 2400, 4800, 9600, 19200
- **Start bit**: 1 bit (fixed)
- **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**: User can select whether CA or CB signals will always be True, or will be used for control.
- **Software Handshaking**: User can select whether to control only transmission or both transmission and reception using X-on and X-off signals.
  - X-on (ASCII 11H)
  - X-off (ASCII 13H)
- **Receive**: 256 bytes

Switching between Remote and Local Modes

*when switched from Local to Remote Mode*
Remote mode is activated when the “COMMunicate:REMote ON” command is received form a controller while local mode is active.

- REMOTE is displayed on.
- All front panel keys except the LOCAL key can no longer be operated any more.
- Settings entered in local mode are retained.

*When switched from Remote to Local Mode*
Pressing the LOCAL key in remote mode puts the instrument in local mode. However, this is not possible of Local Lockout (when the “COMMunicate:LOCKout ON” command is received) has been set by the controller (page 1-6).
Local mode is activated when the “COMMunicate:REMote OFF” command regardless of Local Lockout.

- The REMOTE indicator is turned off.
- All front panel keys are operative.
- Settings entered in remote mode are retained.
2.3 Connecting the Serial Interface Cable

When connecting this instrument to a computer, make sure that the handshaking method, data transmission rate and data format selected for the instrument match those selected for the computer.

For details, refer to the following pages. Also make sure that the correct interface cable is used.

**Connector and Signal Names**

2. RD (Received Data) : Data received from personal computer
   Signal direction...Input

3. SD (Send Data) : Data transmitted to a personal computer
   Signal direction...Output

5. SG (Signal Ground) : Ground for signals

7. RS (Request to Send) : Signal used for handshaking when receiving data from a personal computer
   Signal direction...Output

8. CS (Clear to Send) : Signal used for handshaking when transmitting data to a personal computer
   Signal direction...Input

Pin Nos. 1, 4, 6 and 9 are not used.

**9-25 Pin Connector**

The number between brackets refer to the pin Nos. of the 25-pin connector.

**Signal Direction**

The figure below shows the direction of the signals used by the Serial interface.
### Table of Serial Standard Signals and their Pin No.

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

### Signal line connection example

The pin numbers shown are that of 9-pin connectors. In general, use a cross cable.

- **OFF-OFF / XON-XON**
  - PC: SD 3, RD 2, RS 7, CS 6, SG 5
  - PZ4000: SD 3, RD 2, RS 7, CS 6, SG 5

- **XON-RTS(XON-RS)**
  - PC: SD 3, RD 2, RS 7, CS 6, SG 5
  - PZ4000: SD 3, RD 2, RS 7, CS 6, SG 5

- **CTS-RTS(CS-RS)**
  - PC: SD 3, RD 2, RS 7, CS 6, SG 5
  - PZ4000: SD 3, RD 2, RS 7, CS 6, SG 5
2.4 Handshaking

To use an serial interface for transferring data between this instrument and a computer, it is necessary to use certain procedures by mutual agreement to ensure the proper transfer of data. These procedures are called “handshaking.” Various handshaking systems are available depending on the computer to be used; the same handshaking system must be used for both the computer and this instrument.

This instrument allows you to choose any handshaking mode from the following four modes.

### Handshake format Descriptions

<table>
<thead>
<tr>
<th>Handshake Method</th>
<th>Data Sending Control (control method when sending data to a computer)</th>
<th>Data Receiving Control (control method when receiving data from a computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Software Handshake</td>
<td>Hardware Handshake</td>
</tr>
<tr>
<td>OFF-OFF</td>
<td>Sending stops when X-off is received, and sending is resumed when X-on is received.</td>
<td>Sending stops when CB(CTS) is False, and sending is resumed when CB is True.</td>
</tr>
<tr>
<td>XON-XON</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>XON-RS</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>CS-RS</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

### 1 OFF-OFF

- **Transmission data control**
  There is no handshake status between the instrument and host computer. The X-OFF and X-ON signal from the host computer is processed as data, and the CS signal is ignored.

- **Reception data control**
  There is no handshake status between the recorder and host computer. When the recorder reception buffer becomes full, the excess data is discarded. RS = True (fixed)

### 2 XON-XON

- **Transmission data control**
  A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission when an X-OFF signal is received from the host computer, and will resume transmission when the next X-ON signal is received. A CS signal from the host computer is ignored.

- **Reception data control**
  A software handshake status is established between the instrument and host computer. When the instruments reception buffer vacancy reaches 64bytes, the X-OFF signal will be sent to the host computer. When the reception buffer vacancy reaches 192 bytes, the X-ON signal will be sent. RS = True (fixed)
3 XON-RS

• **Transmission data control**
A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission when an X-OFF signal is received from the host computer, and will resume transmission when the next X-ON signal is received. A CS signal from the host computer is ignored.

• **Reception data control**
A hardware handshake status is established between the instrument and host computer. When the instrument’s reception buffer vacancy reaches 64 bytes, an “RS = False” status will be established. When the reception buffer vacancy reaches 192 bytes, an “RS = True” status will be established.

4 CS-RS

• **Transmission data control**
A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission if a “CS = False” status is established, and will resume the transmission when a “CS = True” status is established. The X-OFF and X-ON signals from the host computer are processed as data.

• **Reception data control**
A hardware handshake status is established between the instrument and host computer. When the instrument’s reception buffer vacancy reaches 64 bytes, an “RS = False” status will be established. When the reception buffer vacancy reaches 192 bytes, an “RS = True” status will be established.

**Precautions Regarding Data Receiving Control**

When handshaking is used to control the reception of data, data may still be sent from the computer 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 handshaking is in effect or not. Data storage to the buffer will begin again when there is free space in the buffer.

![Diagram showing data receiving control using handshaking](image)

**Note**
It is necessary to create a host computer program which prevents the buffers of both the instrument and the computer from becoming full.
2.5 Matching the Data Format

The serial interface of this instrument performs communications using start-stop synchronization. In start-stop synchronization, one character is transmitted at a time. Each character consists of a start bit, data bits, a parity bit and a stop bit. Refer to the figure below.
2.6 Setting up this Instrument

Before You Begin

When using the controller to set the items which can be set locally using the keys on the instrument, or when outputting the setup information or the waveform data to the controller, set the following items.

**Baud rate**
Select from the following choices.
- 1200, 2400, 4800, 9600, 19200

**Data format**
Select the combination of the data length and the stop bit from the following choices.
- 8-NO-1, 7-EVEN-1, 7-ODD-1, 7-NO-2

**Handshaking method**
Select the transmit data control and the receive data control from the following choices.
- NO-NO, XON-XON, XON-RTS, CTS-RTS

**Terminator**
Select from the following choices. The terminator used when sending the data from this instrument is selected on the menu. Use either “LF” or “CR+LF” for the terminator in receiving the data.
- CR, LF, CR+LF

Operating Procedure

**Displaying the Serial (RS-232) menu**
1. Press the MISC key.
2. Press the “GP-IB/RS232” soft key.
3. Press the “Comm Device” soft key to select “RS232.”

**Selecting the baud rate, the data format and etc.**
4. Press the “BaudRate” (baud rate), “Format” (data format), “Rx-Tx” (handshaking method), and the “Terminator” (terminator) soft keys individually, and set each item.
Chapter 3 Before Programming

3.1 Messages

Blocks of message data are transferred between the controller and this instrument during communications. Messages sent from the controller to this instrument are called program messages, and messages sent back from this instrument to the controller are called response messages.

If a program message contains a message unit, i.e. a command which requests a response, this instrument returns a response message. A single response message is always returned in reply to a program message.

Program Messages

The format of a program message is shown below.

Example

:TRIGger:MODE AUTO;SOURCE 1<PMT>

Response Messages

The format of a response message is shown below.

Example

:TRIGger:MODE AUTO;SOURCE 1<RMT>

Program Message Unit Format

The format of a program message unit is shown below.

Program Message Unit Format

A program message unit consists of one or more program message units; each unit corresponds to one command. This instrument executes commands one by one according to the order in which they are received.

Program message units are delimited by a ";". For a description of the format of the program message unit, refer to the explanation given further below.

Example

:TRIGger:MODE AUTO;SOURCE 1<PMT>

Response Message units

A response message consists of one or more response message units: each response message unit corresponds to one response. Response message units are delimited by a ";". For the response message format, refer to the next page.

Example

:TRIGger:MODE AUTO;SOURCE 1<RMT>

PMT

PMT is a terminator used to terminate each program message. The following three types of terminator are available.

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

^END : END message defined in IEEE488.1. (EOI signal)
(The data byte sent with an END message will be the final item of the program message unit.)

NL*END : NL with an END message attached (NL is not included in the program message unit.)

RMT

RMT is the terminator used for every response message. Only one type of response message is available; NL*END.
3.1 Messages

Response message unit format
The format of a program message unit is shown below.

- **<Response header>**
  A response header sometimes precedes the response data. Response data must be separated from the header by a space. For details, refer to page 3-4.

- **<Response data>**
  Response data is used to define a response. If multiple items of response data are used, they must be separated by a “,” (comma). For details, refer to page 3-5.

Example

- **Data**
  - **Header**
  - **Data**

If a program message contains more than one query, responses are made in the same order as the queries. Normally, each query returns only one response message unit, but there are some queries which return more than one response message unit. The first response message unit always responds to the first query, but it is not always true that the ‘n’ th unit always responds to the ‘n’ th query. Therefore, if you want to make sure that a response is made to each query, the program message must be divided up into individual messages.

Points to Note concerning Message Transmission
- It is always possible to send a program message if the previous message which was sent did not contain any queries.
- If the previous message contained a query, it is not possible to send another program message until a response message has been received. An error will occur if a program message is sent before a response message has been received in its entirety. A response message which has not been received will be discarded.
- If an attempt is made by the controller to receive a response message, even if there is no response message, an error will occur. An error will also occur if the controller makes an attempt to receive a response message before transmission of a program message has been completed.

- If a program message of more than one unit is sent and some of the units are incomplete, this instrument receives program message units which the instrument thinks complete and attempts to execute them. However, these attempts may not always be successful and a response may not always be returned, even if the program message contains queries.

Deadlock
This instrument has a buffer memory in which both program and response messages of 1024 bytes or more can be stored. (The number of bytes available will vary depending on the operating state of the instrument.) If the transmission and reception buffer memories become full at the same time, the instrument will not be able to continue the communication operation. This state is called deadlock. In this case, operation can be resumed by discarding the response message. No deadlock will occur, if the size of the program message including the PMT is kept below 1024 bytes. Furthermore, no deadlock will occur if the program message does not contain a query.
3.2 Commands

There are three types of command (program header) which can be sent from the controller to this instrument. They differ in the format of their program headers.

They are
- Common command header
- Compound header
- Simple header

Common Command Header
Commands defined in IEEE 488.2-1987 are called common commands. The header format of a common command is shown below. An asterisk (*) must always be attached to the beginning of a command.

*<Mnemonic>

An example of a common command
*CLS

Compound Header
Commands designed to be used only with this instrument are classified and arranged in a hierarchy according to their function. The format of a compound header is illustrated below. A colon (:) must be used when specifying a lower-level header.

:<Mnemonic>

An example of a compound header
:ACQuire:DIVision

Simple Header
These commands (headers) are functionally independent of each other and are not arranged hierarchically. The format of a simple header is shown below.

:<Mnemonic>

An example of a simple header
:STARt

When Concatenating Commands
Command Group
A command group is a group of commands which have the same compound header. A command group may contain sub-groups.

Example Commands relating to acquisition settings
:ACQuire?
:ACQuire:DIVision
:ACQuire:RLENgth
:ACQuire:TBASe

When Concatenating Commands of the Same Group
This instrument stores the hierarchical level of the command which is currently being executed, and performs analysis on the assumption that the next command to be sent will also belong to the same level. Therefore, it is possible to omit the header if the commands belong to the same group.

Example :ACQuire:DIVision ON;TBASE INTernal

When Concatenating Commands of Different Groups
A colon (:) must be included before the header of a command, if the command does not belong to the same group as the preceding command.

Example :ACQuire:DIVision ON;:DISPlay:FORMat NUMeric

When Concatenating Simple Headers
When you type in a simple header after another command, you must include a colon (:) before the simple header.

Example :ACQuire:DIVision ON;:STARt

When Concatenating Common Commands
Common commands defined in IEEE 488.2-1987 are independent of hierarchical level. Thus, it is not necessary to add a colon (:) before a common command.

Example :ACQuire:DIVision ON;*CLS;TBASE INTernal

Note
A mnemonic is a character string made up of alphanumeric characters.
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 typed in for each command even when commands of the same command group are being concatenated.

Example: :ACQuire:DIvision ON<PMT>:ACQuire:TBASe INternal<PMT>

Upper-level Query
An upper-level query is a compound header to which a question mark is appended. Execution of an upper-level query allows all a group’s settings to be output at once. Some query groups comprising more than three hierarchical levels can output all their lower level settings.

Example: :TIMebase?<PMT>→:TIMEBASE:OBSERVE 100.00E-03;SRATE 1.000000E+06

In reply to a query, a response can be returned as a program message to this instrument. Transmitting a response can restore the settings made when the query was executed. However, some upper-level queries will not return set-up data which is not currently in use. Note that not all a group’s information will necessarily be sent out as a response.

Header Interpretation Rules
This instrument interprets the header received according to the following rules.

- Mnemonics are not case sensitive.
  Example “CURSor” can also be written as “cursor” or “Cursor”.
- The lower-case part of a header can be omitted.
  Example “CURSor” can also be written as “CURSO” or “CURS”.
- If the header ends with a question mark, the command is a query. It is not possible to omit the question mark.
  Example “CURSor?” cannot be abbreviated to anything shorter than “CURS?”.
- If the “x” at the end of a mnemonic is omitted, it is assumed to be “1”.
  Example If “CHANnel<x>” is written as “CHAN”, this represents “CHANnel1”.
- Any part of a command enclosed by [ ] can be omitted.
  Example “TRIGger[:SIMPlle]:LEVel” can be written as “TRIG:LEV”.
- However, a part enclosed by [ ] cannot be omitted if it is located at the end of an upper-level query.
  Example “TRIGger?” and “TRIGger:SIMPle?” belong to different upper-level query levels.
3.3 Response
On receiving a query from the controller, this instrument returns a response message to the controller. A response message is sent in one of the following two forms.

- **Response consisting of a header and data**
  If the query can be used as a program message without any change, a command header is attached to the query, which is then returned.
  
  **Example:**
  ```plaintext```
  ```plaintext```

- **Response consisting of data only**
  If the query cannot be used as a program message unless changes are made to it (i.e., it is a query-only command), no header is attached and only the data is returned. Some query-only commands can be returned after a header is attached to them.
  
  **Example:**
  ```plaintext```
  :CHANnel1:TYPE?<PMT>→VOLTAGE<RMT>
  ```plaintext```

When returning a response without a header
It is possible to remove the header from a response consisting of a header and data. The "COMMunicate:HEADer" command is used to do this.

**Abbreviated form**
Normally, the lower-case part 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 part enclosed by [ ] is also omitted in the abbreviated form.

3.4 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.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Decimal&gt;</td>
<td>Value expressed as a decimal number</td>
</tr>
<tr>
<td></td>
<td>(Example: Number of displayed digits for numerical data)</td>
</tr>
<tr>
<td></td>
<td>→(SETup:RESolution 5)</td>
</tr>
<tr>
<td>&lt;Voltage&gt;&lt;Current&gt;</td>
<td>Physical value</td>
</tr>
<tr>
<td></td>
<td>(Example: Waveform observation time)</td>
</tr>
<tr>
<td></td>
<td>→(VOLtm:BASE:OBServe 100N)</td>
</tr>
<tr>
<td>&lt;Register&gt;</td>
<td>Register value expressed as either binary, octal, decimal or hexadecimal</td>
</tr>
<tr>
<td></td>
<td>(Example: Extended event register value)</td>
</tr>
<tr>
<td></td>
<td>→(STATus:EESetpoint)</td>
</tr>
<tr>
<td>&lt;Character data&gt;</td>
<td>Specified character string (mnemonic). Can be selected from { }</td>
</tr>
<tr>
<td></td>
<td>(Example: Measurement mode)</td>
</tr>
<tr>
<td></td>
<td>→(SETup[:MODE] {NORMAL</td>
</tr>
<tr>
<td>&lt;Boolean&gt;</td>
<td>Indicates ON/OFF. Set to ON, OFF or value</td>
</tr>
<tr>
<td></td>
<td>(Example: CH2 waveform display ON)</td>
</tr>
<tr>
<td></td>
<td>→(CHANnel2:DISPlay ON)</td>
</tr>
<tr>
<td>&lt;Character string data&gt;</td>
<td>Arbitrary character string</td>
</tr>
<tr>
<td></td>
<td>(Example: Waveform label of CH1)</td>
</tr>
<tr>
<td></td>
<td>→(CHANnel:LABel &quot;CH1&quot;)</td>
</tr>
<tr>
<td>&lt;Filename&gt;</td>
<td>Gives the name of a file.</td>
</tr>
<tr>
<td></td>
<td>(Example: Name of file to be saved)</td>
</tr>
<tr>
<td></td>
<td>→FIE:SAVE[:WAVE]:EXECute &quot;CASE1&quot;</td>
</tr>
<tr>
<td>&lt;Block data&gt;</td>
<td>Arbitrary 8-bit data</td>
</tr>
<tr>
<td></td>
<td>(Example: Response to acquired waveform data)</td>
</tr>
<tr>
<td></td>
<td>→#8000000B080CDEFGH13</td>
</tr>
</tbody>
</table>

**<Decimal>**

<Decimal> indicates a value expressed as a decimal number, as shown in the table below. Decimal values are given in the NR form specified in ANSI X3.42-1975. Decimal values which are sent from the controller to this instrument can be sent in any of the forms to <NR3>. In this case, <NRf> appears. For response messages which are returned from this instrument to the controller, the form (<NR1> to <NR3> to be used) is determined by the query. The same form is used, irrespective of whether the value is large or small.

In the case of <NR3>, the "+" after the "E" can be omitted, but the "-" cannot.

If a value outside the setting range is entered, the value will be normalized so that it is just inside the range.

If the value has more than the significant number of digits, the value will be rounded.
3.4 Data

<Voltage>, <Current>, <Time>, <Frequency>
<Voltage>, <Current>, <Time> and <Frequency> indicate decimal values which have physical significance. <Multiplier> or <Unit> can be attached to <NRf>. They can be entered in any of the following forms.

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NRf&gt;&lt;Multiplier&gt;&lt;Unit&gt;</td>
<td>5MV</td>
</tr>
<tr>
<td>&lt;NRf&gt;&lt;Unit&gt;</td>
<td>5E-3V</td>
</tr>
<tr>
<td>&lt;NRf&gt;&lt;Multiplier&gt;</td>
<td>5M</td>
</tr>
<tr>
<td>&lt;NRf&gt;</td>
<td>5E-3</td>
</tr>
</tbody>
</table>

<Multiplier>
Multipliers which can be used are shown below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Exa</td>
<td>$10^{18}$</td>
</tr>
<tr>
<td>P</td>
<td>Peta</td>
<td>$10^{15}$</td>
</tr>
<tr>
<td>T</td>
<td>Tera</td>
<td>$10^{12}$</td>
</tr>
<tr>
<td>G</td>
<td>Giga</td>
<td>$10^{9}$</td>
</tr>
<tr>
<td>M</td>
<td>Mega</td>
<td>$10^{6}$</td>
</tr>
<tr>
<td>K</td>
<td>Kilo</td>
<td>$10^{3}$</td>
</tr>
<tr>
<td>H</td>
<td>Mil</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>U</td>
<td>Micro</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>N</td>
<td>Nano</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>P</td>
<td>Pico</td>
<td>$10^{-12}$</td>
</tr>
<tr>
<td>F</td>
<td>Femto</td>
<td>$10^{-15}$</td>
</tr>
</tbody>
</table>

<Multiplier> and <Unit> are not case sensitive.

"U" is used to indicate "µ".

"MA" is used for Mega (M) to distinguish it from Mili, except for in the case of Megahertz, which is expressed as "MHz". Hence, it is not permissible to use "M" (Mili) for Hertz.

If both <Multiplier> and <Unit> are omitted, the default unit will be used.

Response messages are always expressed in <NR3> form. Neither <Multiplier> nor <Unit> is used, therefore the default unit is used.

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

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NRf&gt;</td>
<td>1</td>
</tr>
<tr>
<td>#H&lt;Hexadecimal value made up of the digits 0 to 9, and A to F&gt;</td>
<td>#H0F</td>
</tr>
<tr>
<td>#O&lt;Octal value made up of the digits 0 to 7&gt;</td>
<td>#O777</td>
</tr>
<tr>
<td>#B&lt;Binary value made up of the digits 0 and 1&gt;</td>
<td>#B01100</td>
</tr>
</tbody>
</table>

<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 3-4.

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{NORMal</td>
<td>HARMonics}</td>
</tr>
</tbody>
</table>

As with a header, the “COMMunicate:VERBose” command can be used to return a response message in its full form. Alternatively, the abbreviated form can be used.

The “COMMunicate:HEADer” command does not affect <character data>.

<Boolean>
<Boolean> is data which indicates ON or OFF, and is expressed in one of the following forms.

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ON</td>
<td>OFF</td>
</tr>
</tbody>
</table>

When <Boolean> is expressed in <NRf> form, OFF is selected if the rounded integer value is "0" and ON is selected if the rounded integer is “Not 0”.

A response message is always “1” if the value is ON and “0” if it is OFF.

<Character String Data>
<Character string data> is not a specified character string like <Character data>. It is an arbitrary character string. A character string must be enclosed in single quotation marks (’) or double quotation marks (“’)

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Character string data&gt;</td>
<td>“ABC” “IEEE488.2-1987”</td>
</tr>
</tbody>
</table>

Response messages are always enclosed in double quotation marks.
If a character string contains a double quotation mark ("), the double quotation mark will be replaced by two concatenated double quotation marks ("""). This rule also applies to a single quotation mark (').

`<Character string data>` is an arbitrary character string, therefore this 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 will be detected if a quotation mark is omitted.

`<Filename>`
Gives the name of a file. The format is as follows.

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{&lt;NRi}</td>
<td>&lt;Character data&gt;</td>
</tr>
</tbody>
</table>

If you input an `<NRi>` value, the system converts the value (after rounding to the nearest integer) to the corresponding 8-character ASCII string. (If you set the value to 1, the name becomes "00000001".) Note that negative values are not allowed.

If you enter a `<character data>` or `<character string>` argument that is longer than eight characters, only the first eight characters are used.

Response messages always return filenames as `<character string>` arguments.

`<Block data>`

`<Block data>` is arbitrary 8-bit data. `<Block data>` is only used for response messages. Response messages are expressed in the following form.

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>`$N$&lt;N-digits decimal value&gt;</td>
<td>&lt;Data byte string&gt;`</td>
</tr>
</tbody>
</table>

`#N` Indicates that the data is `<Block data>`. "N" is an ASCII character string number (digits) which indicates the number of data bytes that follow.

`<N-digits decimal value>` Indicates the number of bytes of data. (Example: 00000010 = 10 bytes)

`<Data byte string>` The actual data. (Example: ABCDEFGHIJ)

Data is comprised of 8-bit values (0 to 255). This means that the ASCII code "0AH", which stands for "NL", can also be a code used for data. Hence, care must be taken when programming the controller.

### 3.5 Synchronization with the Controller

#### Overlap Commands and Sequential Commands

There are two kinds of command; overlap commands and sequential commands. Execution of an overlap command may start before execution of the previously sent command is completed.

The `[CHANnel1:VOLTage:RANGe]` command, for example, is a sequential command. Assume that you set a new voltage range value and immediately request return of the new value, as follows:

`:CHANnel1:VOLTage:RANGe 200V;RANGe?<PMT>`

In this case, the oscilloscope always returns the newest setting ("200V"). This is because it always completes processing of the current sequential command (in this case, "RANGe 200V") before moving on to the next command ("RANGe?").

In contrast, assume that you begin a file load and then immediately query the voltage range value:

`:FILE:LOAD:SETup "FILE1";:CHANnel1:VOLTage:RANGe?`

Because "FILE:LOAD:SETup" is an overlapped command, the oscilloscope will advance to the "CHANnel1:VOLTage:RANGe?" command before it finishes the load. The returned voltage range value will not show the newest setting, but will rather show the setting in use before the setup was changed.

Obviously, use of overlapped commands may in some cases produce inappropriate results. Where necessary, you can avoid such problems as described below.

#### Synchronization with an Overlap Command Using the *WAI command

The *WAI command causes the commands which follow it to wait until an overlap command has been executed.

Example

`:COMMunicate:OPSE #0040;:FILE:LOAD:SETup "FILE1";*WAI;:CHANnel1:VOLTage:RANGe?`

The "COMMunicate:OPSE" command is used to designate which commands are to be subject to the *WAI command. In the above example, only auto set-up is designated.

Since a *WAI command is executed just before "CHANnel1:VOLTage:RANGe?", "CHANnel1:VOLTage:RANGe?" will not be executed until auto set-up has been completed.
3.5 Synchronization with the Controller

Using the COMMunicate:OVERlap command
The "COMMunicate:OVERlap" command is used to enable or disable overlap operation.
Example

Using the *OPC command
The *OPC command causes the OPC bit (bit 0) of the standard event register (page 5-3) to be set to "1" when an overlap operation has been completed.
Example
:COMMunicate:OPSE #H0040; *ESE 1 ; *ESR? ; *SRE 32 ; :FILE:LOAD:SETup "FILE1" ; *OPC<PMT>
(Response to *ESR? is decoded.)

Using the *OPC? query
The *OPC? query generates a response when an overlap operation has been completed.
Example
:COMMunicate:OPSE #H0040; :FILE:LOAD:SETup "FILE1" ; *OPC?<PMT>
(Response to *OPC? is decoded.)

Note
Most commands are sequential commands. Commands used in Chapter 4 are sequential commands unless otherwise specified.

Synchronization with Non-Overlap Commands
Synchronization is sometimes required for reasons other than communications-related reasons, such as the activation of a trigger, even if a sequential command is used.

Using STATus:CONDition? query
A "STATus:CONDition?" query is used to make a query about the contents of the condition register (page 5-4). It is possible to judge whether acquisition is in progress or not by reading bit 0 of the condition register. Bit 0 is "1" if acquisition is in progress, and "0" if acquisition is stopped.
Example
:SSTart<PMT>
:STATus:CONDition?<PMT>
(Returns to the previous status if bit 0 is found to be "1" when the response is decoded.)
:WAVeform:SEND?<PMT>
A "WAVeform:SEND?" query will not be executed until bit 0 of the condition register has been set to "0".

Using STATus:CONDition? query
A "STATus:CONDition?" query is used to make a query about the contents of the condition register (page 5-4). It is possible to judge whether acquisition is in progress or not by reading bit 0 of the condition register. Bit 0 is "1" if acquisition is in progress, and "0" if acquisition is stopped.
Example
:SSTart<PMT>
:STATus:CONDition?<PMT>
(Returns to the previous status if bit 0 is found to be "1" when the response is decoded.)
:WAVeform:SEND?<PMT>
A "WAVeform:SEND?" query will not be executed until bit 0 of the condition register has been set to "0".

Using STATus:CONDition? query
A "STATus:CONDition?" query is used to make a query about the contents of the condition register (page 5-4). It is possible to judge whether acquisition is in progress or not by reading bit 0 of the condition register. Bit 0 is "1" if acquisition is in progress, and "0" if acquisition is stopped.
Example
:SSTart<PMT>
:STATus:CONDition?<PMT>
(Returns to the previous status if bit 0 is found to be "1" when the response is decoded.)
:WAVeform:SEND?<PMT>
A "WAVeform:SEND?" query will not be executed until bit 0 of the condition register has been set to "0".
Using the extended event register

Changes in the condition register are reflected in the extended event register (page 5-4).

Example

```
:STATus:FILTER1 FALL;:STATus:EESE 1;EESR?
*SRE 8;SSTart<PMT>
```

(Response to STATus:EESR? is decoded.)
(Service request is awaited.)

```
:SSTart<PMT>
```

(*SRE 8 is decoded.)
(Service request is awaited.)

The "STATus:FILTER1 FALL" command sets the transition filter such that Bit 0 (FILTER1) of the Extended Event Register sets to 1 when Bit 0 of the Condition Register changes from 1 to 0.

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

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

The "*SRE" command is used to generate a service request caused solely by the extended event register.

"WAVeform:SEND?" will not be 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?
SSTart<PMT>
```

(Response to STATus:EESR? is decoded.)

```
:COMMunicate:WAIT 1;:WAVeform:SEND?<PMT>
```

For a description of "STATus:FILTER1 FALL" and "STATus:EESR?", refer to "Using the extended event register" on this page.

"COMMunicate:WAIT 1" means that communications is halted until bit 0 of the extended event register is set to "1".
# Chapter 4 Commands

## 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABORt Group</strong></td>
<td><strong>ABORt</strong> Aborts data acquisition.</td>
<td>4-11</td>
</tr>
<tr>
<td><strong>ACQuire Group</strong></td>
<td><strong>ACQuire?</strong> Queries all settings related to data acquisition.</td>
<td>4-11</td>
</tr>
<tr>
<td></td>
<td><strong>ACQuire:DIVision</strong> Sets whether or not to divide the record length or queries the current setting.</td>
<td>4-11</td>
</tr>
<tr>
<td></td>
<td><strong>ACQuire:RENGTH</strong> Sets the record length or queries the current setting.</td>
<td>4-11</td>
</tr>
<tr>
<td></td>
<td><strong>ACQuire:TBASE</strong> Sets the sampling block or queries the current setting.</td>
<td>4-11</td>
</tr>
<tr>
<td><strong>CHANnel Group</strong></td>
<td><strong>CHANnel&lt;x&gt;</strong> Queries all settings related to the vertical axis of each channel.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:CURRENT?</strong> Queries all settings related to the current input channel.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:CURRENT:RANGE</strong> Sets the current range of the current input channel or queries the current setting.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:CURRENT:SRATIO</strong> Sets the current sensor’s scaling constant of the current input channel or queries the current setting.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:CURRENT:TERMINAL</strong> Sets the current measurement terminal of the current input channel or queries the current setting.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:DISPLAY</strong> Turns ON/OFF the waveform display of each channel or queries the current setting.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:LABEL</strong> Sets the waveform label of each channel.</td>
<td>4-13</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:POSITION</strong> Sets the vertical position (the GND position) of each channel or queries the current setting.</td>
<td>4-14</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:SPEED?</strong> Queries all settings related to the revolution sensor signal input channel.</td>
<td>4-14</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:SPEED:FRANGE</strong> Sets the frequency range of the revolution sensor signal input channel (pulse input) or queries the current setting.</td>
<td>4-14</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:SPEED:RANGE</strong> Sets the input range of the revolution sensor signal input channel or queries the current setting.</td>
<td>4-14</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:SPEED:TYPE</strong> Sets the input type of the revolution sensor signal input channel or queries the current setting.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:TORQUE?</strong> Queries all settings related to the torque meter signal input channel.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:TORQUE:RANGE</strong> Sets the input range of the torque meter signal input channel or queries the current setting.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:TYPE?</strong> Queries the input type of each channel.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:VOLTage?</strong> Queries all settings related to the voltage input channel.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:VOLTage:RANGE</strong> Sets the voltage range of the voltage input channel or queries the current setting.</td>
<td>4-15</td>
</tr>
<tr>
<td></td>
<td><strong>CHANnel&lt;x&gt;:VZOOM</strong> Sets the vertical zoom factor or queries the current setting.</td>
<td>4-15</td>
</tr>
<tr>
<td><strong>COMMunicate Group</strong></td>
<td><strong>COMMunicate?</strong> Queries all settings related to communications.</td>
<td>4-16</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:HEADER</strong> Sets whether or not to attach headers to response data or queries the current setting.</td>
<td>4-16</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:LOCKout</strong> Sets/releases local lockout.</td>
<td>4-16</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:OPSE</strong> Sets the overlap commands for *OPC, *OPC?, and *WAI or queries the current setting.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:OPSR?</strong> Queries the operation pending status register.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:OVERlap</strong> Sets the commands to permit overlap operation or queries the current setting.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:REMOTE</strong> Switches between remote and local.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:STATus?</strong> Queries the line-specific status.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:VERBose</strong> Sets whether to use the full or abbreviated form for response data or queries the current setting.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:WAIT</strong> Waits for an extended event to occur.</td>
<td>4-17</td>
</tr>
<tr>
<td></td>
<td><strong>COMMunicate:WAIT?</strong> Generates a response when one of the specified extended events occurs.</td>
<td>4-17</td>
</tr>
<tr>
<td><strong>CURSor Group</strong></td>
<td><strong>CURSor?</strong> Queries all settings related to cursor measurements.</td>
<td>4-19</td>
</tr>
<tr>
<td></td>
<td><strong>CURSor:HORIZontal?</strong> Queries all settings related to the H cursor.</td>
<td>4-19</td>
</tr>
<tr>
<td></td>
<td><strong>CURSor:HORIZontal:DY</strong> Queries the Y-axis value between the H cursors.</td>
<td>4-19</td>
</tr>
<tr>
<td></td>
<td><strong>CURSor:HORIZontal:POSITION&lt;x&gt;</strong> Sets the H cursor position or queries the current setting.</td>
<td>4-19</td>
</tr>
</tbody>
</table>
### 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CURSor:HORIZONTAL:TRACE</td>
<td>Sets the waveform on which to place the H cursor or queries the current setting.</td>
<td>4-19</td>
</tr>
<tr>
<td>:CURSor:HORIZONTAL:Y&lt;x&gt;?</td>
<td>Queries the Y-axis value of the H cursor.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer?</td>
<td>Queries all settings related to the marker.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:DX?</td>
<td>Queries the X-axis value between the marker.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:DY?</td>
<td>Queries the Y-axis value between the marker.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:FFT&lt;x&gt;</td>
<td>Sets the X-axis value of the marker position for the FFT result or queries the current setting.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:JUMP</td>
<td>Jumps to the zoomed waveform of the marker.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:POSition&lt;x&gt;</td>
<td>Queries the 1/Δ value of the horizontal axis between the marker.</td>
<td>4-20</td>
</tr>
<tr>
<td>:CURSor:MARKer:POSition&lt;x&gt;</td>
<td>Sets the marker position or queries the current setting.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:MARKer:POSition&lt;x&gt;</td>
<td>Sets the waveform on which to place the marker or queries the current setting.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:MARKer:POSition&lt;x&gt;</td>
<td>Queries the X-axis value of the marker position.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:MARKer:POSition&lt;x&gt;</td>
<td>Queries the Y-axis value of the marker position.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:MARKer:TYPE</td>
<td>Sets the cursor type or queries the current setting.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:VERTical?</td>
<td>Queries all settings related to the V cursor.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:VERTical:DX?</td>
<td>Queries the X-axis value between the V cursors.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:VERTical:FFT&lt;x&gt;</td>
<td>Sets the V cursor position with respect to the FFT result.</td>
<td>4-21</td>
</tr>
<tr>
<td>:CURSor:VERTical:FREQuency?</td>
<td>Queries the 1/Δ value of the horizontal axis between the V cursors.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:VERTical:POSition&lt;x&gt;</td>
<td>Sets the V cursor position or queries the current setting.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:VERTical:TRAce</td>
<td>Sets the waveform on which to place the V cursor or queries the current setting.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:VERTical:X&lt;x&gt;?</td>
<td>Queries the X-axis value of the V cursor position.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:XY?</td>
<td>Queries all settings related to the XY cursor.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:XY:DX?</td>
<td>Queries the X-axis value between the XY cursors.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:XY:POSition&lt;x&gt;</td>
<td>Sets the XY cursor position or queries the current setting.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:XY:TRAce?</td>
<td>Queries the waveform on which the XY cursor is placed.</td>
<td>4-22</td>
</tr>
<tr>
<td>:CURSor:XY:X&lt;x&gt;?</td>
<td>Queries the X-axis value of the XY cursor position.</td>
<td>4-22</td>
</tr>
</tbody>
</table>

#### DISPlay Group

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay?</td>
<td>Queries all settings related to the screen display.</td>
<td>4-25</td>
</tr>
<tr>
<td>:DISPlay:BAR?</td>
<td>Queries all settings related to the bar graph display.</td>
<td>4-25</td>
</tr>
<tr>
<td>:DISPlay:BAR:CURSor&lt;x&gt;</td>
<td>Sets the marker position (harmonic order) on the bar graph display or queries the current setting.</td>
<td>4-25</td>
</tr>
<tr>
<td>:DISPlay:BAR:ITEM&lt;x&gt;</td>
<td>Sets the bar graph display items (function, element) or queries the current setting.</td>
<td>4-25</td>
</tr>
<tr>
<td>:DISPlay:BAR:ORDER</td>
<td>Sets the start and end harmonic orders of the bar graph display or queries the current setting.</td>
<td>4-25</td>
</tr>
<tr>
<td>:DISPlay:DATE</td>
<td>Turns ON/OFF the date and time displays or queries the current setting.</td>
<td>4-26</td>
</tr>
<tr>
<td>:DISPlay:FORMat</td>
<td>Sets the display format or queries the current setting.</td>
<td>4-26</td>
</tr>
<tr>
<td>:DISPlay:NUMeric?</td>
<td>Queries all settings related to the numerical display.</td>
<td>4-26</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics?</td>
<td>Queries all settings related to the numerical display during harmonic measurement.</td>
<td>4-26</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:FORMat</td>
<td>Sets the numerical display format during harmonic measurement or queries the current setting.</td>
<td>4-26</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:CURSor</td>
<td>Sets the cursor position of the numerical display during harmonic measurement or queries the current setting.</td>
<td>4-27</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:ITEM&lt;x&gt;</td>
<td>Sets the numerical displayed items during harmonic measurement or queries the current setting.</td>
<td>4-27</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:CURSor</td>
<td>Sets the cursor position on the list display during harmonic measurement or queries the current setting.</td>
<td>4-27</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:LIST&lt;x&gt;</td>
<td>Sets the list display items during harmonic measurement or queries the current setting.</td>
<td>4-27</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:HARMonics:PRESet</td>
<td>Sets the numerical display items to a preset pattern during harmonic measurement.</td>
<td>4-27</td>
</tr>
</tbody>
</table>
### 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay[:NUMeric]:NORMal?</td>
<td>Queries all settings related to the numerical display during normal measurement.</td>
<td>4-27</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:NORMal:FCURsor</td>
<td>Sets the cursor position of the numerical display (All display) during normal measurement or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:NORMal:IAMount</td>
<td>Sets the numerical display format during normal measurement or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:NORMal:ICURsor</td>
<td>Sets the cursor position of the numerical display (split display) during normal measurement or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:NORMal:ITEM&lt;x&gt;</td>
<td>Sets the numerical displayed item during normal measurement or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay[:NUMeric]:NORMal:PRESet</td>
<td>Sets the numerical display items to a preset pattern during normal measurement.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay:VECTor?</td>
<td>Queries all settings related to the vector display.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay:VECTor:IMAG</td>
<td>Sets the zoom factor of the current display during vector display or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay:VECTor:NUMeric</td>
<td>Turns ON/OFF the numerical data display during vector display or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay:VECTor:UMAG</td>
<td>Sets the zoom factor of the voltage display during vector display or queries the current setting.</td>
<td>4-28</td>
</tr>
<tr>
<td>:DISPlay:WAVE?</td>
<td>Queries all settings related to the waveform display.</td>
<td>4-29</td>
</tr>
<tr>
<td>:DISPlay:WAVE:{CHANnel&lt;x&gt;</td>
<td>MATH&lt;x&gt;}</td>
<td>Turns ON/OFF the channel/computed waveform display or queries the current setting.</td>
</tr>
<tr>
<td>:DISPlay:WAVE:FORMat</td>
<td>Sets the display format of the waveform or queries the current setting.</td>
<td>4-29</td>
</tr>
<tr>
<td>:DISPlay:WAVE:GRAticule</td>
<td>Sets the graticule type (grid) or queries the current setting.</td>
<td>4-29</td>
</tr>
<tr>
<td>:DISPlay:WAVE:INTERpolate</td>
<td>Sets the interpolation method of the waveform or queries the current setting.</td>
<td>4-29</td>
</tr>
<tr>
<td>:DISPlay:WAVE:MAPPING?</td>
<td>Queries all settings related to the waveform mapping to the split screen.</td>
<td>4-29</td>
</tr>
<tr>
<td>:DISPlay:WAVE:MAPPING:{CHANnel&lt;x&gt;</td>
<td>MATH&lt;x&gt;}</td>
<td>Sets the {channel waveform</td>
</tr>
<tr>
<td>:DISPlay:WAVE:MAPPING[:MODE]</td>
<td>Sets the waveform mapping method for the split screen or queries the current setting.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:WAVE:SVAlue</td>
<td>Turns ON/OFF the scale value display or queries the current setting.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:WAVE:TLABel</td>
<td>Turns ON/OFF the waveform label display or queries the current setting.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:XY?</td>
<td>Queries all settings related to the X-Y display.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:XY:FFT</td>
<td>Sets the range of the FFT waveform to be displayed on the X-Y display or queries the current setting.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:XY:INTERpolate</td>
<td>Sets the interpolation method of the waveform or queries the current setting.</td>
<td>4-30</td>
</tr>
<tr>
<td>:DISPlay:XY:POSITION</td>
<td>Sets the range of the T-Y waveform to be displayed on the X-Y display or queries the current setting.</td>
<td>4-31</td>
</tr>
<tr>
<td>:DISPlay:XY:XTRace</td>
<td>Sets the channel to assign to the X-axis of the X-Y display or queries the current setting.</td>
<td>4-31</td>
</tr>
</tbody>
</table>

### FILE Group

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:FILE?</td>
<td>Queries all settings related to file operations.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:CDIRectory</td>
<td>Changes the current directory.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:DELete:IMAGE:{TIFF</td>
<td>BMP</td>
<td>PSCRipt}</td>
</tr>
<tr>
<td>:FILE:DELete:NUMeric:{ASCii</td>
<td>FLOAT}</td>
<td>Deletes a numerical data file.</td>
</tr>
<tr>
<td>:FILE:DELete:SETup</td>
<td>Deletes a setup parameter file.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:DELete:WAVE:{BINary</td>
<td>ASCii</td>
<td>FLOAT}</td>
</tr>
<tr>
<td>:FILE:DRIVE</td>
<td>Sets the drive (medium) setting.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:FORMAT</td>
<td>Formats the floppy disk.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:FREE?</td>
<td>Queries the free space (bytes) on the drive.</td>
<td>4-34</td>
</tr>
</tbody>
</table>
### Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:FILE:LOAD:ABORT</td>
<td>Aborts loading a file.</td>
<td>4-34</td>
</tr>
<tr>
<td>:FILE:LOAD:SETup</td>
<td>Loads a setup parameter file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:LOAD:WAVE</td>
<td>Loads a waveform data file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:MDIRectory</td>
<td>Creates a directory.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:PATH?</td>
<td>Queries the absolute path of the current directory.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE?</td>
<td>Queries all settings related to saving a file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:ABORT</td>
<td>Aborts saving the file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:ANAMing</td>
<td>Sets whether or not to automatically assign file names or queries the current setting.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:COMment</td>
<td>Sets the comment that is attached to the file being saved or queries the current setting.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:NUMeric?</td>
<td>Queries all settings related to saving the numerical data to a file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:NUMeric[:EXECute]</td>
<td>Saves the numerical data to a file.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:NUMeric:LIST?</td>
<td>Queries all settings related to saving the numerical list data to a file during harmonic measurement.</td>
<td>4-35</td>
</tr>
<tr>
<td></td>
<td>Turns ON/OFF the output of each element when saving numerical list data to a file during harmonic measurement or queries the current setting.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:NUMeric:LIST:{&lt;Element&gt;</td>
<td>&lt;List-Function&gt;</td>
<td>&lt;SIGMo&gt;}</td>
</tr>
<tr>
<td>:FILE:SAVE:NUMeric:TYPE</td>
<td>Sets ON/OFF the output of each function when saving numerical list data to a file during harmonic measurement or queries the current setting.</td>
<td>4-35</td>
</tr>
<tr>
<td>:FILE:SAVE:SETup[:EXECute]</td>
<td>Saves the setup parameters to a file.</td>
<td>4-36</td>
</tr>
<tr>
<td>:FILE:SAVE:WAVE?</td>
<td>Queries all settings related to saving the waveform data to a file.</td>
<td>4-36</td>
</tr>
<tr>
<td>:FILE:SAVE:WAVE[:EXECute]</td>
<td>Saves the waveform data to a file.</td>
<td>4-36</td>
</tr>
<tr>
<td>:FILE:SAVE:WAVE:RANGe</td>
<td>Sets the range of the waveform to save to the file or queries the current setting.</td>
<td>4-36</td>
</tr>
<tr>
<td>:FILE:SAVE:WAVE:TRACE</td>
<td>Sets the waveform to save to the file or queries the current setting.</td>
<td>4-36</td>
</tr>
<tr>
<td>:FILE:SAVE:WAVE:TYPE</td>
<td>Sets the format of the waveform data being saved or queries the current setting.</td>
<td>4-36</td>
</tr>
<tr>
<td>:HCOPY?</td>
<td>Queries all settings related to screen data output.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:ABORT</td>
<td>Aborts data output and paper feeding.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:CENTronics?</td>
<td>Queries all settings related to the external printer output.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:CENTronics:COLOR</td>
<td>Sets the color (ON/OFF) of the external printer output or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:CENTronics:FORMAT</td>
<td>Sets the command format that is output to the printer or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:COMMENT</td>
<td>Sets the comment that is printed at the lower section of the screen or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:DIRection</td>
<td>Sets the output destination of the data or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:EXECute</td>
<td>Executes data output.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:FORMat</td>
<td>Sets the output data format or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:PRINTER:DLIst</td>
<td>Executes output of the numerical data list to the built-in printer.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:PRINTER:FEED</td>
<td>Feeds the paper (built-in printer).</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:SAVE?</td>
<td>Queries all settings related to saving the file.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:SAVE:ANAMing</td>
<td>Sets whether or not to automatically assign file names or queries the current setting.</td>
<td>4-38</td>
</tr>
<tr>
<td>:HCOPY:SAVE:COMment</td>
<td>Sets the comment that is attached to the file being saved or queries the current setting.</td>
<td>4-39</td>
</tr>
<tr>
<td>:HCOPY:SAVE:NAME</td>
<td>Sets the file name or queries the current setting.</td>
<td>4-39</td>
</tr>
<tr>
<td>:HCOPY:[TIFF</td>
<td>BMP]?</td>
<td>Queries all settings related to the TIFF/BMP format.</td>
</tr>
<tr>
<td>:HCOPY:[TIFF</td>
<td>BMP]:COLOR</td>
<td>Sets the color for the TIFF/BMP format or queries the current setting.</td>
</tr>
<tr>
<td>:HCOPY:[TIFF</td>
<td>BMP]:COMPression</td>
<td>Sets whether or not to compress the data in TIFF/BMP format or queries the current setting.</td>
</tr>
<tr>
<td>:IMAGE?</td>
<td>Queries all settings related to the output of the screen image data.</td>
<td>4-40</td>
</tr>
<tr>
<td>:IMAGE:COLOR</td>
<td>Sets the color of the screen image data being output or queries the current setting.</td>
<td>4-40</td>
</tr>
<tr>
<td>:IMAGE:FORMat</td>
<td>Sets the output format of the screen image data or queries the current setting.</td>
<td>4-40</td>
</tr>
<tr>
<td>:IMAGE:SEND?</td>
<td>Queries the screen image data.</td>
<td>4-40</td>
</tr>
<tr>
<td>:INPut?</td>
<td>Queries all settings related to all input modules.</td>
<td>4-44</td>
</tr>
</tbody>
</table>
### 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:INPut]:MODULE?</td>
<td>Queries the model name of each input module.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR?</td>
<td>Queries all settings related to the motor module.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:FILTER?</td>
<td>Sets the line filter for the motor module or queries the current setting.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:FILTER[:LINE]</td>
<td>Sets the line filter for the motor module or queries the current setting.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:FILTER[:ZCROSS]</td>
<td>Sets the zero crossing filter for the motor module or queries the current setting.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:PM?</td>
<td>Queries all settings related to the motor output of the motor module.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:PM:SCALing</td>
<td>Sets the scaling factor used during motor output computation on the motor module or queries the current setting.</td>
<td>4-44</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:POLE</td>
<td>Sets the motor's number of poles for the motor module or queries the current setting.</td>
<td>4-45</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED?</td>
<td>Queries all settings related to the revolution sensor signal input for the motor module.</td>
<td>4-45</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:FILTER</td>
<td>Sets the line filter for the revolution sensor signal input (pulse input) for the motor module or queries the current setting.</td>
<td>4-45</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:PULSE</td>
<td>Sets the pulse count of the revolution sensor signal input (pulse input) for the motor module or queries the current setting.</td>
<td>4-45</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:RANGE</td>
<td>Sets the voltage range of the revolution sensor signal input for the motor module or queries the current setting.</td>
<td>4-45</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:SCALING</td>
<td>Sets the scaling factor used during rotating speed computation on the motor module or queries the current setting.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:TYPE</td>
<td>Sets the input type of the revolution sensor signal input for the motor module or queries the current setting.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:UNIT</td>
<td>Sets the unit to add to the rotating speed computation result or queries the current setting.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:SPEED:SYNC</td>
<td>Sets the frequency measurement source for the motor module or queries the current setting.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:TORQUE?</td>
<td>Queries all settings related to the torque meter signal input for the motor module.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:TORQUE:RANGE</td>
<td>Sets the voltage range of the torque meter signal input for the motor module or queries the current setting.</td>
<td>4-46</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:TORQUE:SCALING</td>
<td>Sets the scaling factor used during torque computation on the motor module or queries the current setting.</td>
<td>4-47</td>
</tr>
<tr>
<td>[:INPut]:MOTOR:TORQUE:UNIT</td>
<td>Sets the unit to add to the torque computation result or queries the current setting.</td>
<td>4-47</td>
</tr>
<tr>
<td>[:INPut]:POWER?</td>
<td>Queries all settings related to the power measurement module.</td>
<td>4-47</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT?</td>
<td>Queries all settings related to the current measurement on the power measurement module.</td>
<td>4-47</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:AUTO?</td>
<td>Queries the ON/OFF state of the current auto range function of all elements with the power measurement modules.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:AUTO[:ALL]</td>
<td>Turns ON/OFF the current auto range function of all elements with the power measurement modules.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:AUTO[:ELEMENT&lt;&gt;&lt;&gt;</td>
<td>Turns ON/OFF the current auto range function of each element with power measurement module or queries the current setting.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:RANGE?</td>
<td>Queries the current range of all elements with the power measurement modules.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:RANGE[:ALL]</td>
<td>Sets the current range of all elements with the power measurement modules.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:RANGE[:ELEMENT&lt;&gt;&lt;&gt;</td>
<td>Sets the current range of each element with the power measurement module or queries the current setting.</td>
<td>4-48</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:SRATIO?</td>
<td>Queries the current sensor transformation ratio of all elements with the power measurement modules.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:SRATIO[:ALL]</td>
<td>Sets the current sensor transformation ratio of all elements with the power measurement modules.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut]:POWERS:CURRENT:SRATIO[:ELEMENT&lt;&gt;&lt;&gt;</td>
<td>Sets the current sensor transformation ratio of each element with the power measurement module or queries the current setting.</td>
<td>4-49</td>
</tr>
</tbody>
</table>
### Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:INPut][:POWer]:CURRent:TERMinal?</td>
<td>Queries the current measurement terminals of all elements with the power measurement modules.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut][:POWer]:CURRent:TERMinal[:ALL]</td>
<td>Sets the current measurement terminals of all elements with the power measurement modules.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut][:POWer]:CURRent:TERMinal:ELEMent&lt;x&gt;</td>
<td>Sets the current measurement terminals of each element with the power measurement module or queries the current setting.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER?</td>
<td>Queries all settings related to the filter for the power measurement module.</td>
<td>4-49</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE?</td>
<td>Queries the line filter setting of all elements with the power measurement modules.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE[:ALL]</td>
<td>Sets the line filter setting of all elements with the power measurement modules.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE:ELEMent&lt;x&gt;</td>
<td>Sets the line filter setting of each element with the power measurement module or queries the current setting.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:ZCRoss?</td>
<td>Queries the zero crossing filter of all elements with the power measurement modules.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:ZCRoss[:ALL]</td>
<td>Sets the zero crossing filter of all elements with the power measurement modules.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:ZCRoss:ELEMent&lt;x&gt;</td>
<td>Sets the zero crossing filter of each element with the power measurement module or queries the current setting.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing?</td>
<td>Queries all settings related to scaling for the power measurement module.</td>
<td>4-50</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:{PT</td>
<td>CT</td>
<td>SFACtor}?</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:{PT</td>
<td>CT</td>
<td>SFACtor}[:ALL]</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:{PT</td>
<td>CT</td>
<td>SFACtor}:ELEMent&lt;x&gt;</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:STATe?</td>
<td>Queries the ON/OFF state of the scaling function of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:STATe[:ALL]</td>
<td>Turns ON/OFF the scaling function of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:STATe:ELEMent&lt;x&gt;</td>
<td>Turns ON/OFF the scaling function of each element with the power measurement module or queries the current setting.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage?</td>
<td>Queries all settings related to the voltage measurement for power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:AUTO?</td>
<td>Queries the ON/OFF state of the voltage auto range function of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:AUTO[:ALL]</td>
<td>Turns ON/OFF the voltage auto range function of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:AUTO:ELEMent&lt;x&gt;</td>
<td>Turns ON/OFF the voltage auto range function of each element with the power measurement module or queries the current setting.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:RANGE?</td>
<td>Queries the voltage range of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:RANGE[:ALL]</td>
<td>Sets the voltage range of all elements with the power measurement modules.</td>
<td>4-51</td>
</tr>
<tr>
<td>[:INPut][:POWer]:VOLTage:RANGE:ELEMent&lt;x&gt;</td>
<td>Sets the voltage range of each element with the power measurement module or queries the current setting.</td>
<td>4-52</td>
</tr>
</tbody>
</table>
## 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MATH Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:MATH&lt;&gt;?</td>
<td>Queries all settings related to computations.</td>
<td>4-53</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:EXECute</td>
<td>Executes computation.</td>
<td>4-53</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:EXPRESSION</td>
<td>Sets the equation or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:FFT?</td>
<td>Queries all settings related to the FFT.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:FFT:POINT</td>
<td>Sets the number of points for the FFT or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:FFT:WINDOW</td>
<td>Sets the window function for the FFT or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:FUNCTION</td>
<td>Enables/disables the computation function or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:[:MODE]</td>
<td>Turns ON/OFF the computation or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:POINT</td>
<td>Sets the start and end points of the computation or queries the current setting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:SCALing?</td>
<td>Queries all settings related to scale converting.</td>
<td>4-54</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:SCALing:MODE</td>
<td>Sets the scale converting or queries the current setting.</td>
<td>4-55</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:SCALing:VALUE</td>
<td>Sets the upper and lower limits for manual scaling or queries the current setting.</td>
<td>4-55</td>
</tr>
<tr>
<td>:MATH&lt;&gt;:UNIT</td>
<td>Sets the unit to attach to the computed result or queries the current setting.</td>
<td>4-55</td>
</tr>
<tr>
<td><strong>MEASure Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:MEASURE?</td>
<td>Queries all settings related to measurements.</td>
<td>4-57</td>
</tr>
<tr>
<td>:MEASURE:AVERaging?</td>
<td>Queries all settings related to averaging.</td>
<td>4-57</td>
</tr>
<tr>
<td>:MEASURE:AVERaging:COUNT</td>
<td>Sets the number of averaging counts or queries the current setting.</td>
<td>4-57</td>
</tr>
<tr>
<td>:MEASURE:AVERaging[:STATE]</td>
<td>Turns ON/OFF the averaging function or queries the current setting.</td>
<td>4-57</td>
</tr>
<tr>
<td>:MEASURE:DMeasure</td>
<td>Sets the delta computation or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:FUNCTION&lt;&gt;?</td>
<td>Queries all settings related to the user-defined function.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:FUNCTION&lt;&gt;:EXPRESSION</td>
<td>Sets the equation for the user-defined function or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:FUNCTION&lt;&gt;:[:STATE]</td>
<td>Enable/disable the user-defined function or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:FUNCTION&lt;&gt;:UNIT</td>
<td>Sets the unit to attach to the computed result of the user-defined function or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:HARMonics?</td>
<td>Queries all settings related to the measurement during harmonic measurement.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:HARMonics:ORDER</td>
<td>Sets the minimum and maximum harmonic orders to be analyzed during harmonic measurement or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:HARMonics:THD</td>
<td>Sets the equation used to determine the THD (total harmonic distortion) during harmonic measurement or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE[:MODE]</td>
<td>Turns ON/OFF the measurement computation or queries the current setting.</td>
<td>4-58</td>
</tr>
<tr>
<td>:MEASURE:PC?</td>
<td>Queries all settings related to determination of Pc (Corrected Power).</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PC:IEC</td>
<td>Sets the equation used to determine the Pc (Corrected Power) or queries the current setting.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PC:P&lt;&gt;x</td>
<td>Sets the parameters used to determine the Pc (Corrected Power) or queries the current setting.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod?</td>
<td>Queries all settings related to the computation period.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod:CURSor?</td>
<td>Queries all settings when specifying the computation period with the cursors.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod:CURSor[:POSITION]</td>
<td>Sets the computation period when specifying the period with the cursors or queries the current setting.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod:ETRigger?</td>
<td>Queries all settings when using the external trigger signal to determine the computation period.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod:ETRigger[:PATTERN]</td>
<td>Sets the pattern that is used when determining the computation period with the external trigger signal or queries the current setting.</td>
<td>4-59</td>
</tr>
<tr>
<td>:MEASURE:PERiod:EXECute</td>
<td>Executes the computation.</td>
<td>4-60</td>
</tr>
<tr>
<td>:MEASURE:PERiod[:MODE]</td>
<td>Sets the method used to specify the computation period or queries the current setting.</td>
<td>4-60</td>
</tr>
<tr>
<td>:MEASURE:PERiod:ZCRoss?</td>
<td>Queries all settings when using the zero crossing detection to determine the computation period.</td>
<td>4-60</td>
</tr>
<tr>
<td>:MEASURE:PERiod:ZCRoss:SYNChronize?</td>
<td>Sets the synchronizing source for all elements when using the zero crossing detection to determine the computation period.</td>
<td>4-60</td>
</tr>
</tbody>
</table>
4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:MEASure:PERiod:ZCRoss[:SYNChronize]:ELEMent&lt;&lt;</td>
<td>Sets the synchronizing source for each element when using the zero crossing detection to determine the computation period.</td>
<td>4-60</td>
</tr>
<tr>
<td>:MEASure:PHASe</td>
<td>Sets the display format of the phase difference or queries the current setting.</td>
<td>4-60</td>
</tr>
<tr>
<td>:MEASure:SFORmula</td>
<td>Sets the equation used to determine S (apparent power) or queries the current setting.</td>
<td>4-60</td>
</tr>
<tr>
<td><strong>NULL Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:NULL</td>
<td>Turns ON/OFF the NULL function or queries the current setting.</td>
<td>4-61</td>
</tr>
<tr>
<td><strong>NUMeric Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:NUMeric?</td>
<td>Queries all settings related to the numerical data output.</td>
<td>4-63</td>
</tr>
<tr>
<td>:NUMeric:FORMat</td>
<td>Sets the format of the numerical data that are sent using the &quot;NUMeric:{NORMal</td>
<td>HARMonics</td>
</tr>
<tr>
<td>:NUMeric:HARMonics?</td>
<td>Queries all settings related to the numerical data output during harmonic measurement.</td>
<td>4-63</td>
</tr>
<tr>
<td>:NUMeric:HARMonics:CLEar</td>
<td>Clears the numerical data output items during harmonic measurement.</td>
<td>4-63</td>
</tr>
<tr>
<td>:NUMeric:HARMonics:ITEM&lt;&lt;</td>
<td>Sets the numerical data output items during harmonic measurement or queries the current setting.</td>
<td>4-63</td>
</tr>
<tr>
<td>:NUMeric:HARMonics:NUMBER</td>
<td>Sets the number of numerical data that are sent using the &quot;NUMeric:HARMonics:VALue?&quot; command or queries the current setting.</td>
<td>4-63</td>
</tr>
<tr>
<td>:NUMeric:HARMonics:PRESet</td>
<td>Sets the numerical data output items to a preset pattern during harmonic measurement.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:HARMonics:VALue?</td>
<td>Queries the numerical data during harmonic measurement.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:LIST?</td>
<td>Queries all settings related to the output of the numerical list data during harmonic measurement.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:LIST:ITEM</td>
<td>Sets the output items of the numerical list data during harmonic measurement or queries the current setting.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:LIST:ORDER</td>
<td>Sets the maximum harmonic order of the numerical list data to output during harmonic measurement or queries the current setting.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:LIST:SELECT</td>
<td>Sets the output components of the numerical list data during harmonic measurement or queries the current setting.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric:LIST:VALue?</td>
<td>Queries the numerical list data during harmonic measurement.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]?</td>
<td>Queries all settings related to the numerical data output during normal measurement.</td>
<td>4-64</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]:CLEar</td>
<td>Clears the numerical data output items during normal measurement.</td>
<td>4-65</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]:ITEM&lt;&lt;</td>
<td>Sets the numerical data output items during normal measurement or queries the current setting.</td>
<td>4-65</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]:NUMBER</td>
<td>Sets the number of numerical data during normal measurement or queries the current setting.</td>
<td>4-65</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]:PRESet</td>
<td>Sets the numerical data output items to a preset pattern during normal measurement.</td>
<td>4-65</td>
</tr>
<tr>
<td>:NUMeric[:NORMal]:VALue?</td>
<td>Queries the numerical data during normal measurement.</td>
<td>4-65</td>
</tr>
<tr>
<td><strong>SETup Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:SETup?</td>
<td>Queries all settings related to the measurement mode.</td>
<td>4-69</td>
</tr>
<tr>
<td>:SETup[:INITialize]</td>
<td>Initializes the settings.</td>
<td>4-69</td>
</tr>
<tr>
<td>:SETup[:MODE]</td>
<td>Sets the measurement mode or queries the current setting.</td>
<td>4-69</td>
</tr>
<tr>
<td>:SETup:PLLSource</td>
<td>Sets the PLL source during harmonic measurement or queries the current setting.</td>
<td>4-69</td>
</tr>
<tr>
<td>:SETup:RESolution</td>
<td>Sets the number of displayed digits for numerical data or queries the current setting.</td>
<td>4-69</td>
</tr>
<tr>
<td>:SETup:WIRing</td>
<td>Sets the wiring method or queries the current setting.</td>
<td>4-70</td>
</tr>
<tr>
<td><strong>SSTart Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:SSTart</td>
<td>Executes single start.</td>
<td>4-70</td>
</tr>
<tr>
<td><strong>START Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:START</td>
<td>Starts data acquisition.</td>
<td>4-70</td>
</tr>
<tr>
<td><strong>STATus Group</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:STATus?</td>
<td>Queries all settings related to the communication status function.</td>
<td>4-71</td>
</tr>
<tr>
<td>:STATus:CONDITION?</td>
<td>Queries the status register.</td>
<td>4-71</td>
</tr>
<tr>
<td>:STATus:EESE(Extended Event Status Enable register)</td>
<td>Sets the extended event enable register or queries the current setting.</td>
<td>4-71</td>
</tr>
</tbody>
</table>
### 4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:STATus:ESR? (Extended Event Status Register)</td>
<td>Queries and clears the extended event register.</td>
<td>4-71</td>
</tr>
<tr>
<td>:STATus:ERRor?</td>
<td>Queries the code and information of the error.</td>
<td>4-72</td>
</tr>
<tr>
<td>:STATus:FILTER&lt;x&gt;</td>
<td>Sets the transition filter or queries the current setting.</td>
<td>4-72</td>
</tr>
<tr>
<td>:STATus:QENable</td>
<td>Sets whether or not to store messages other than errors in the error queue or queries the current setting.</td>
<td>4-72</td>
</tr>
<tr>
<td>:STATus:MESSage</td>
<td>Sets whether or not to attach a message to the &quot;STATus:ERRor?&quot; response or queries the current setting.</td>
<td>4-72</td>
</tr>
<tr>
<td>:STATus:SPOLL? (Serial Poll)</td>
<td>Executes serial polling.</td>
<td>4-72</td>
</tr>
</tbody>
</table>

**STOP Group**

:STOP

Stops data acquisition. | 4-72 |

**SYSTem Group**

:SYSTem?

Queries all settings related to the system. | 4-74 |

:SYSTem:DATE

Sets the date or queries the current setting. | 4-74 |

:SYSTem:LANguage

Sets the message language or queries the current setting. | 4-74 |

:SYSTem:LCD?

Queries all settings related to the LCD monitor. | 4-74 |

:SYSTem:LCD:BRIGHTness

Sets the brightness of the LCD monitor or queries the current setting. | 4-74 |

:SYSTem:LCD:COLOR?<x>

Queries all settings related to the display colors of the LCD monitor. | 4-74 |

:SYSTem:LCD:COLOR:GRAPH?<x>

Queries all settings related to the display color of graphic items. | 4-74 |

:SYSTem:LCD:COLOR:GRAPH:MODE

Sets the display color mode of graphic items or queries the current setting. | 4-74 |

:SYSTem:LCD:COLOR:TEXT?<x>

Queries all settings related to the display color of text items. | 4-74 |

:SYSTem:LCD:COLOR:TEXT:MODE

Sets the display color mode of text items or queries the current setting. | 4-74 |

:SYSTem:SCSI?

Queries all settings related to the SCSI-ID. | 4-75 |

:SYSTem:SCSI:INITialize

Initializes SCSI related settings. | 4-75 |

:SYSTem:SCSI:OWNid

Sets the SCSI ID of this instrument or queries the current setting. | 4-75 |

:SYSTem:TIME

Sets the time or queries the current setting. | 4-75 |

**TIMebase Group**

:TIMebase?

Queries all settings related to the time base (horizontal axis). | 4-76 |

:TIMebase:OBServe

Sets the observation time of the waveform or queries the current setting. | 4-76 |

:TIMebase:SRATE

Sets the sampling rate or queries the current setting. | 4-76 |

**TRIGger Group**

:TRIGger?

Queries all settings related to the trigger. | 4-78 |

:TRIGger:ACTION?<x>

Queries all settings related to action-on-trigger. | 4-78 |

:TRIGger:ACTION:ACQCount

Sets the action count of action-on-trigger or queries the current setting. | 4-78 |

:TRIGger:ACTION:HCOPY

Sets whether or not to output screen image data (ON/OFF) when an action is activated, or queries the current setting. | 4-78 |

:TRIGger:ACTION:SAVE

Sets whether or not to save the waveform data to the storage medium (ON/OFF) when an action is activated, or queries the current setting. | 4-78 |

:TRIGger:DELAY

Sets the trigger delay or queries the current setting. | 4-78 |

:TRIGger:DREFERence

Sets the trigger position or queries the current setting. | 4-78 |

:TRIGger:EDGE?<x>

Queries all settings related to the edge trigger. | 4-78 |

:TRIGger:EDGE:LEVel

Sets the trigger level for the edge trigger or queries the current setting. | 4-78 |

:TRIGger:EDGE:SLOPe

Sets the trigger slope for the edge trigger or queries the current setting. | 4-78 |

:TRIGger:MODE

Sets the trigger mode or queries the current setting. | 4-79 |

:TRIGger:SOURCE

Sets the trigger source or queries the current setting. | 4-79 |

:TRIGger:TYPE

Sets the trigger type or queries the current setting. | 4-79 |

:TRIGger:WINDOW?<x>

Queries all settings related to the window trigger. | 4-79 |
4.1 Command Listing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TRIGger:WINDOW:CENTer</td>
<td>Sets the center level for the window trigger or queries the current setting.</td>
<td>4-79</td>
</tr>
<tr>
<td>:TRIGger:WINDOW:CONDition</td>
<td>Sets the trigger condition for the window trigger or queries the current setting.</td>
<td>4-79</td>
</tr>
<tr>
<td>:TRIGger:WINDOW:WIDTh</td>
<td>Sets the window width for the window trigger or queries the current setting.</td>
<td>4-79</td>
</tr>
</tbody>
</table>

**WAVeform Group**

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVeform?</td>
<td>Queries all settings related to the waveform data.</td>
<td>4-80</td>
</tr>
<tr>
<td>:WAVeform:BYTeorder</td>
<td>Sets the byte order of the waveform data or queries the current setting.</td>
<td>4-80</td>
</tr>
<tr>
<td>:WAVeform:END</td>
<td>Sets the end point of the output of the waveform data or queries the current setting.</td>
<td>4-80</td>
</tr>
<tr>
<td>:WAVeform:FORMAT</td>
<td>Sets the format of the waveform data or queries the current setting?</td>
<td>4-81</td>
</tr>
<tr>
<td>:WAVeform:LENGTH?</td>
<td>Queries the total number of data points of the waveform.</td>
<td>4-81</td>
</tr>
<tr>
<td>:WAVeform:RANGE?</td>
<td>Queries the range value that is used to convert the waveform to physical data.</td>
<td>4-81</td>
</tr>
<tr>
<td>:WAVeform:SEND?</td>
<td>Queries the waveform data.</td>
<td>4-81</td>
</tr>
<tr>
<td>:WAVeform:SRATe?</td>
<td>Queries the sampling rate of the acquired data.</td>
<td>4-81</td>
</tr>
<tr>
<td>:WAVeform:START</td>
<td>Sets the start point of the output of the waveform data or queries the current setting.</td>
<td>4-82</td>
</tr>
<tr>
<td>:WAVeform:TDATe?</td>
<td>Queries the string containing the trigger date and time when the waveform was acquired.</td>
<td>4-82</td>
</tr>
<tr>
<td>:WAVeform:TRACe</td>
<td>Sets the waveform or queries the current setting.</td>
<td>4-82</td>
</tr>
<tr>
<td>:WAVeform:TRIGger?</td>
<td>Queries the trigger position of the acquired data.</td>
<td>4-82</td>
</tr>
<tr>
<td>:WAVeform:ZCRoss?</td>
<td>Queries zero crossing data of all channels.</td>
<td>4-82</td>
</tr>
</tbody>
</table>

**ZOOM Group**

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ZOOM?</td>
<td>Queries all settings related to the zooming of the waveform.</td>
<td>4-83</td>
</tr>
<tr>
<td>:ZOOM:ALLOCation?</td>
<td>Queries all settings related to the zoomed waveform.</td>
<td>4-83</td>
</tr>
<tr>
<td>:ZOOM:ALLOCation:{CHANnel&lt;x&gt;</td>
<td>MATH&lt;x&gt;}}</td>
<td>Sets whether or not to select the waveform to be zoomed or queries the current setting.</td>
</tr>
<tr>
<td>:ZOOM:FORMAT</td>
<td>Sets the display format of the zoomed waveform or queries the current setting.</td>
<td>4-84</td>
</tr>
<tr>
<td>:ZOOM:MAG&lt;x&gt;</td>
<td>Sets the zoom factor or queries the current setting.</td>
<td>4-84</td>
</tr>
<tr>
<td>:ZOOM:[MODE]</td>
<td>Sets the display mode of the zoomed waveform or queries the current setting.</td>
<td>4-84</td>
</tr>
<tr>
<td>:ZOOM:POSITION&lt;x&gt;</td>
<td>Sets the position of the zoom box or queries the current setting.</td>
<td>4-84</td>
</tr>
</tbody>
</table>

**Common Command Group**

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?(CALibrate)</td>
<td>Performs calibration (zero level compensation) and queries the result.</td>
<td>4-85</td>
</tr>
<tr>
<td>*CLS(Clear Status)</td>
<td>Clears the standard event register, extended event register, and error queue.</td>
<td>4-85</td>
</tr>
<tr>
<td>*ESE(standard Event Status Enable register)</td>
<td>Sets the standard event enable register or queries the current setting.</td>
<td>4-85</td>
</tr>
<tr>
<td>*ESR?(standard Event Status Register)</td>
<td>Queries the standard event register and clears the register.</td>
<td>4-86</td>
</tr>
<tr>
<td>*IDN?(Identify)</td>
<td>Queries the instrument model.</td>
<td>4-86</td>
</tr>
<tr>
<td>*OPC( Operation Complete)</td>
<td>After the completion of the specified overlap command, sets the OPC event.</td>
<td>4-86</td>
</tr>
<tr>
<td>*OPC?( Operation Complete)</td>
<td>Creates a response, after the completion of the specified overlap command.</td>
<td>4-86</td>
</tr>
<tr>
<td>*OPT?(Option)</td>
<td>Queries installed options.</td>
<td>4-86</td>
</tr>
<tr>
<td>*PSC(Power-on Status Clear)</td>
<td>Sets whether or not to clear each register on power up or queries the current setting.</td>
<td>4-86</td>
</tr>
<tr>
<td>*RST(ReSeT)</td>
<td>Initializes the command group settings.</td>
<td>4-86</td>
</tr>
<tr>
<td>*SRE(Service Request Enable register)</td>
<td>Sets the service request enable register or queries the current setting.</td>
<td>4-87</td>
</tr>
<tr>
<td>*STB?(Status Byte)</td>
<td>Queries the status byte register.</td>
<td>4-87</td>
</tr>
<tr>
<td>*TRG(TRIGger)</td>
<td>Executes single start.</td>
<td>4-87</td>
</tr>
<tr>
<td>*TST( TeST)</td>
<td>Executes the self-test and queries the result.</td>
<td>4-87</td>
</tr>
<tr>
<td>*WAI(WAIT)</td>
<td>Waits until the execution of the specified overlap command completes before executing the commands that are specified after this command.</td>
<td>4-87</td>
</tr>
</tbody>
</table>
4.2 **ABORt Group**

The commands in the ABORt group are used to abort the data acquisition operation. These commands can be used to make the same settings and inquiries as when the ABORT (SHIFT + SINGLE START) key on the front panel is pressed.

:ABORt

- **Function**: Aborts data acquisition.
- **Syntax**: :ABORt
- **Example**: :ABORT
- **Description**: For the details regarding the difference between the ":ABORt" and "STOP" commands, see the PZ4000 User's Manual.

4.3 **ACQuire Group**

The commands in the ACQuire Group deal with data acquisitions. These commands can be used to make the same settings and inquiries as when the ACQ (SHIFT + TRIGGER) key on the front panel is pressed.

:ACQuire?

- **Function**: Queries all settings related to data acquisition.
- **Syntax**: :ACQuire?
- **Example**: :ACQUIRE?

:ACQuire:DIVision

- **Function**: Sets whether or not to divide the record length or queries the current setting.
- **Syntax**: :ACQuire:DIVision {<Boolean>}
- **Example**: :ACQUIRE:DIVISION OFF

:ACQuire:RLENgth

- **Function**: Sets the record length or queries the current setting.
- **Syntax**: :ACQuire:RLENgth {<NRf>}
- **Example**: :ACQUIRE:RLENGTH 100000

:ACQuire:TBASE

- **Function**: Sets the time base or queries the current setting.
- **Syntax**: :ACQuire:TBASE {INTERNAL|EXTERNAL}
- **Example**: :ACQUIRE:TBASE INTERNAL
4.4 CHANnel Group

The commands in the CHANnel Group deal with the vertical axis of each channel. These commands can be used to make the same settings and inquiries as when the CH1 to CH8 keys on the front panel are pressed.
4.4  :CHANnel<x>?

Function Queries all settings related to the vertical axis of each channel.
Syntax :CHANnel<x>?
<x> = 1 to 8
Example :CHANNEL1?
→ :CHANNEL1:DISPLAY 1;VOLTAGE: RANGE 2.00E+03; :CHANNEL1:VZOOM 1.00; POSITION 0.000; LABEL "CH1"

:CHANnel<x>:CURRent?

Function Queries all settings related to the current input channel.
Syntax :CHANnel<x>:CURRent?
<x> = 1 to 8
Example :CHANNEL2:CURRENT?
→ :CHANNEL2:CURRENT: TERMINAL 5.0E+00; RANGE 10.0E+00
Description If you specify a channel that does not have the 253751/253752 power measurement module installed, an error will occur.

:CHANnel<x>:CURRent:RANGE

Function Sets the current range of the current input channel or queries the current setting.
Syntax :CHANnel<x>:CURRent:RANGE {<current>|<voltage>|AUTO}
:CHANnel<x>:CURRent:RANGE?
<x> = 1 to 8
<current> = 0.1, 0.2, 0.4, 1, 2, 4, 10(A) (when TERMINal = 5(A))
<current> = 1, 2, 4, 10, 20, 40, 100(A) (when TERMINal = 20(A))
<voltage> = 0.1, 0.2, 0.4, 1(V) (when TERMINal = SENSor)
AUTO = Auto range
Example :CHANnel2:CURRENT:RANGE 10A
:CHANnel2:CURRENT:RANGE?→:CHANnel2:CURRENT: RANGe 10.0E+00
Description • The selectable range is determined by the setting of the current input terminal (:CHANnel<x>:CURRent:TERMinal).
• If you specify a channel that does not have the 253751/253752 power measurement module installed, an error will occur.
• The ":INPut:POWer:CURRent:SRATio: ELEMent<x> (where <x> is the element number)" command can be used to make the same settings and inquiries.

:CHANnel<x>:CURRent:TERMinal

Function Sets the current input terminal of the current input channel or queries the current setting.
Syntax :CHANnel<x>:CURRent:TERMinal {<current>|SENSor}
:CHANnel<x>:CURRent:TERMinal?
<x> = 1 to 8
<current> = 5(A) (for the 253751 power measurement module)
<current> = 5, 20(A) (for the 253752 power measurement module)
SENSor = current sensor
Example :CHANnel2:CURRENT:TERMINAL 5A
:CHANnel2:CURRENT:TERMINAL?→:CHANnel2:CURRENT: TERMINAL 5.0E+00
Description • If you specify a channel that does not have the 253752 power measurement module installed, an error will occur.
• The ":INPut:POWer:CURRent:SRATio: ELEMent<x> (where <x> is the element number)" command can be used to make the same settings and inquiries.

:CHANnel<x>:DISPlay

Function Turns ON/OFF the waveform display of each channel or queries the current setting.
Syntax :CHANnel<x>:DISPlay {<Boolean>}
:CHANnel<x>:DISPlay?
<x> = 1 to 8
Example :CHANnel1:DISPLAY ON
:CHANnel1:DISPLAY?→:CHANnel1:DISPLAY 1
Description The ":DISPLAY:WAVE:CHANnel<x>" command can be used to make the same settings and inquiries.
### 4.4 CHANnel Group

#### :CHANnel<x>:LABel

**Function**: Sets the waveform label of each channel or queries the current setting.

**Syntax**:
- [:CHANnel<x>:LABel {<string>}
- [:CHANnel<x>:LABel?
  
  - **<x>** = 1 to 8
  - **<string>** = 8 characters or less

**Example**: 
- CHANNEL1:LABEL "CH1"
- CHANNEL1:LABEL?

**Description**: Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.

 SPEed = Revolution sensor signal input
 TORQue = Torque meter signal input

#### :CHANnel<x>:POSition

**Function**: Sets the vertical position (the GND position) of each channel or queries the current setting.

**Syntax**:
- [:CHANnel<x>:POSition {<NRf>}
- [:CHANnel<x>:POSition?
  
  - **<x>** = 1 to 8
  - **<NRf>** = -130.000 to 130.000(%)

**Example**: 
- CHANNEL1:POSition 0
- CHANNEL1:POSITION?

**Description**: If the 253771 motor module is not installed, an error will occur.

#### :CHANnel<x>:SPEed?

**Function**: Queries all settings related to the revolution sensor signal input channel.

**Syntax**:
- [:CHANnel<x>:SPEed?
  
  - **<x>** = 7 (fixed)

**Example**: 
- CHANNEL7:SPEED?

**Description**: If the 253771 motor module is not installed, an error will occur.

#### :CHANnel<x>:SPEed:FRANge

**Function**: Sets the frequency range of the revolution sensor signal input channel (pulse input) or queries the current setting.

**Syntax**:
- [:CHANnel<x>:SPEed:FRANge {<frequency>|AUTO}
- [:CHANnel<x>:SPEed:FRANge?
  
  - **<x>** = 7 (fixed)
  - **<frequency>** = 40(Hz): 1 to 40 Hz
    - 800(Hz): 16 to 800 Hz
    - 8k(Hz): 250 to 8 kHz
    - 200k(Hz): 2 k to 200 kHz
  - **AUTO** = Auto range

**Example**: 
- CHANNEL7:SPEED:FRANGE 200KHZ
- CHANNEL7:SPEED:FRANGE?→CHANNEL7:SPEED:FRANGE 200.00E+03

**Description**:
- Set the `<frequency>` to the maximum value within the frequency range.
- This command is valid when the input format of the revolution sensor signal (:CHANnel<x>:SPEed:TYPE) is set to "PULSe (pulse input)."
- If the 253771 motor module is not installed, an error will occur.
- The "INPut:MOtor:SPEed:FRANGe" command can be used to make the same settings and inquiries.

#### :CHANnel<x>:SPEed:RANGe

**Function**: Sets the input range of the revolution sensor signal input channel or queries the current setting.

**Syntax**:
- [:CHANnel<x>:SPEed:RANGe {<voltage>|AUTO}
- [:CHANnel<x>:SPEed:RANGe?
  
  - **<x>** = 7 (fixed)
  - **<voltage>** = 1, 2, 5, 10, 20, and 50(V)
  - **AUTO** = Auto range

**Example**: 
- CHANNEL7:SPEED:RANGE 50V
- CHANNEL7:SPEED:RANGE?

**Description**:
- When the input format of the revolution sensor signal (:CHANnel<x>:SPEed:TYPE) is set to "PULSe (pulse input)," it is fixed to 5 (V).
- If the 253771 motor module is not installed, an error will occur.
- The "INPut:MOtor:SPEed:RANGe" command can be used to make the same settings and inquiries.
:CHANnel<x>:SPEED:TYPE
Function: Sets the signal type of the revolution sensor signal input channel or queries the current setting.
Syntax: :CHANnel<x>:SPEED:TYPE {ANALog|PULSe}  
Example: :CHANnel7:SPEED:TYPE ANALOG
Description: • If the 253771 motor module is not installed, an error will occur.  
• The ":INPut:MOTor:SPEed:TYPE" command can be used to make the same settings and inquiries.

:CHANnel<x>:TORque?
Function: Queries all settings related to the torque meter signal input channel.
Syntax: :CHANnel<x>:TORque?
Example: :CHANnel8:TORQUE?
Description: If the 253771 motor module is not installed, an error will occur.

:CHANnel<x>:TORque:RANGE
Function: Sets the input range of the torque meter signal input channel or queries the current setting.
Syntax: :CHANnel<x>:TORque:RANGE {<voltage>|AUTO}
Example: :CHANnel8:TORQUE:RANGE 50.0E+00
Description: • If the 253771 motor module is not installed, an error will occur.  
• The ":INPut:MOTor:TORque:RANGE" command can be used to make the same settings and inquiries.

:CHANnel<x>:VOLTage?
Function: Queries all settings related to the voltage input channel.
Syntax: :CHANnel<x>:VOLTage?
Example: :CHANnel1:VOLTage? → :CHANnel1:VOLTage:RANGE 2.00E+03

:CHANnel<x>:VOLTage:RANGE
Function: Sets the voltage range of the voltage input channel or queries the current setting.
Syntax: :CHANnel<x>:VOLTage:RANGE {<voltage>|AUTO}
Example: :CHANnel1:VOLTage:RANGE 2000V
Description: The ":INPut:POWer:VOLTage:RANGE:ELement<x> (where <x> is the element number)" command can be used to make the same settings and inquiries.

:CHANnel<x>:VZoom
Function: Sets the vertical zoom factor or queries the current setting.
Syntax: :CHANnel<x>:VZoom {<NRf>}
Example: :CHANnel1:VZoom 1
Description: • If the 253771 motor module is not installed, an error will occur.  
• The ":INPut:MOTor:TORque:RANGE" command can be used to make the same settings and inquiries.
4.5 COMMunicate Group

The commands in the COMMunicate 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 or not to attach headers to response data or queries the current setting.

(Example of a response with a header: SETUP:MODE NORMAL, example of a response without a header: NORMAL)

**Syntax**
:COMMunicate:HEADER {<Boolean>}

**Example**
:COMMUNICATE:HEADER ON
:COMMUNICATE:HEADER?
→ :COMMUNICATE:HEADER 1

**:COMMunicate:LOCKout**

**Function**
Sets/releases local lockout.

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

**Example**
:COMMUNICATE:LOCKOUT ON
:COMMUNICATE:LOCKOUT?
→ :COMMUNICATE:LOCKOUT 1

**Description**
This is a dedicated command for the serial interface. An interface message is available for the GPIB interface.
4.5 COMMunicate Group

**:COMMunicate:OPSE**

(Operation Pending Status Enable register)

**Function**
Sets the overlap commands for *OPC, *OPC?, and *WAI or queries the current setting.

**Syntax**
**:COMMunicate:OPSE <Register>
**:COMMunicate:OPSE?

<Register> = 0 to 65535, See the diagram for the :COMMunicate:WAIT? command.

**Example**
**:COMMUNICATE:OPSE 65535
**:COMMunicate:OPSE? = 96

**Description**
All bits are set to 1 in the above example to set all commands to overlap. However, bits that are fixed to 0 do not change, and therefore, only bits 5 and 6 are set to 1.

**:COMMunicate:OPSR?**

(Operation Pending Status Register)

**Function**
Queries the operation pending status register.

**Syntax**
**:COMMunicate:OPSR?

**Example**
**:COMMUNICATE:OPSR?

→ 0

**Description**
For the operation pending registers, see the diagram for the :COMMunicate:WAIT? command.

**:COMMunicate:OVERlap**

Function sets the commands to permit overlap operation or queries the current setting.

**Syntax**
**:COMMunicate:OVERlap <Register>
**:COMMunicate:OVERlap?

<Register> = 0 to 65535, See the diagram for the :COMMunicate:WAIT? command.

**Example**
**:COMMUNICATE:OVERLAP 65535
**:COMMunicate:OVERLAP? = 96

**Description**
- All bits are set to 1 in the above example to set all commands to overlap. However, bits that are fixed to 0 do not change, and therefore, only bits 5 and 6 are set to 1.
- For the description regarding how to synchronize the program using the COMMunicate:OVERlap command, see page 3-7.
- Bits 5 and 6 are set to 1 in the above example to set all overlap commands (See the diagram for the :COMMunicate:WAIT? command.)

**:COMMunicate:REMote**

Function switches between remote and local. ON is remote.

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

**Example**
**:COMMUNICATE:REMOTE ON
**:COMMunicate:REMote?

→ 1

**Description**
This is a dedicated command for the serial interface. An interface message is available for the GP-IB interface.

**:COMMunicate:STATus?**

Function queries the line-specific status.

**Syntax**
**:COMMunicate:STATus?

**Example**
**:COMMUNICATE:STATUS?

→ 0

**Description**
The meaning of each status bit is as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>GP-IB</th>
<th>Serial</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unrecoverable Parity error transmission error</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Framing error</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Break character detected</td>
<td></td>
</tr>
<tr>
<td>3-6</td>
<td>Always 0</td>
<td></td>
</tr>
</tbody>
</table>

The status bit is set when the causing event occurs and cleared when it is read.

**:COMMunicate:VERBose**

Function sets whether to use the full (example: SETUP:MODE NORMAL) or abbreviated (example: SET NORM) form for response data or queries the current setting.

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

**Example**
**:COMMUNICATE:VERBOSE ON
**:COMMunicate:VERBOSE?

→ 1

**:COMMunicate:WAIT**

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

**Syntax**
**:COMMunicate:WAIT <Register>
**:COMMunicate:WAIT? <Register>

<Register> = 0 to 65535 (extended event register, see page 5-4.)

**Example**
**:COMMUNICATE:WAIT 1
**:COMMunicate:WAIT? 65535

→ 1

**Description**
For the description regarding how to synchronize the program using COMMunicate:WAIT, see page 3-9.

**:COMMunicate:WAIT?**

Function generates a response when one of the specified extended events occurs.

**Syntax**
**:COMMunicate:WAIT? <Register>

<Register> = 0 to 65535 (extended event register, see page 5-4.)

**Example**
**:COMMUNICATE:WAIT? 65535

Operation pending status register/overlap enable register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation pending status register</td>
</tr>
<tr>
<td>1</td>
<td>Operation pending overlap enable</td>
</tr>
<tr>
<td>2</td>
<td>Operation pending command</td>
</tr>
<tr>
<td>3</td>
<td>Operation pending error</td>
</tr>
<tr>
<td>4</td>
<td>Operation pending screen</td>
</tr>
<tr>
<td>5</td>
<td>Operation pending display</td>
</tr>
<tr>
<td>6</td>
<td>Operation pending printer</td>
</tr>
<tr>
<td>7</td>
<td>Operation pending remote</td>
</tr>
</tbody>
</table>

When bit 5 (PRN) = 1:
- Printer operation is not complete.
When bit 6 (ACS) = 1:
- Medium access is not complete.
4.6 CURSor Group

The commands in the CURSor Group deal with cursor measurements. These commands can be used to make the same settings and inquiries as when the CURSOR key on the front panel is pressed.
:CURSor?
Function Queries all settings related to cursor measurements.
Syntax :CURSor?
Example :CURSOR?

:CURsor:HORizontal?
Function Queries all settings related to the H cursor.
Syntax :CURSor:HORizontal?
Example :CURSOR:HORIZONTAL?

:CURsor:HORizontal:POSition<x>
Function Sets the H cursor position or queries the current setting.
Syntax :CURSor:HORizontal:POSition<x> {<NRf>}
Example :CURSOR:HORIZONTAL:POSITION1 25.0;POSITION2 -25.0

:CURsor:HORizontal:TRACe
Function Sets the waveform on which to place the H cursor or queries the current setting.
Syntax :CURSor:HORizontal:TRACe {<NRf>|MATH<x>}
Example :CURSOR:HORIZONTAL:TRACE 1

:CURsor:HORizontal:DY?
Function Queries the Y-axis value (physical value) between the H cursors.
Syntax :CURSor:HORizontal:DY?
Example :CURSOR:HORIZONTAL:DY?

:CURsor:HORizontal:DY?
Function Returns NaN (Not A Number) if the "CURsor:TYPE" is not set to Horizontal or HAVerical.
Description "NAN (Not A Number)" will also be returned, if the "DISPLAY:FORMat" setting does not include the waveform display.
4.6 CURSor Group

:CURSor:HORizontal:Y<x>?
Function Queries the Y-axis value (physical value) of the H cursor.
Syntax :CURSor:HORizontal:Y<x>?
Example :CURSOR:HORIZONTAL:Y1? → 50.000E+00
Description • "NAN (Not A Number)" will be returned, if the ":CURSor:TYPE" is not set to Horizontal or HAVertical.
• "NAN (Not A Number)" will also be returned, if the ":DISPLAY:FORMat" setting does not include the waveform display.

:CURSor:MARKer?
Function Queries all settings related to the marker.
Syntax :CURSor:MARKer?
Example :CURSOR:MARKER? → :CURSOR:MARKER:
TRACE1 1;TRACE2 1;POSITION1 20.000E-03; POSITION2 80.000E-03

:CURSor:MARKer:DX?
Function Queries the X-axis value (physical value) between the markers.
Syntax :CURSor:MARKer:DX?
Example :CURSOR:MARKER:DX? → 60.000E-03

:CURSor:MARKer:DY?
Function Queries the Y-axis value (physical value) between the markers.
Syntax :CURSor:MARKer:DY?
Example :CURSOR:MARKER:DY? → 157.26E+00
Description • "NAN (Not A Number)" will be returned, if the ":CURSor:TYPE" is not set to MARKer.
• "NAN (Not A Number)" will also be returned, if the ":DISPLAY:FORMat" setting does not include the waveform display.

:CURSor:MARKer:FFT<x>
Function Sets the X-axis value of the marker position for the FFT result or queries the current setting.
Syntax :CURSor:MARKer:FFT<x> {<frequency>|<NRf>}
Example :CURSOR:MARKER:FFT1 200kHz
:CURSOR:MARKER:FFT1? → :CURSOR:MARKER:
FFT1 200.00E+03
Description • This command is valid when ":CURSor:MARKer:TRACe<x>" is set to MATH<x> and the equation of MATH<x> is set to FFT.
• The range and resolution of <frequency> is determined from the sampling rate and the number of FFT points.
• <NRf> is set in terms of harmonic order. The range depends on the number of FFT points as follows. For the procedure to set the number of FFT points, see the ":MATH<x>:FFT:POINt" command.
For 1000 points : 0 to 500
For 2000 points : 0 to 1000
For 10000 points : 0 to 5000

:CURSor:MARKer:JUMP
Function Jumps to a waveform of the marker.
Syntax :CURSor:MARKer:JUMP {M1_MA|M1_Z1|M1_Z2|M2_MA|M2_Z1|M2_Z2}
Example :CURSOR:MARKER:JUMP M1_L1
Description The parameters "M1" and "M2" represent markers 1 and 2, respectively. "MA," "Z1," and "Z2" represent the main waveform, zoomed waveforms 1 and 2, respectively.

:CURSor:MARKer:PERDt?
Function Queries the 1/Δ value of the horizontal axis between the markers.
Syntax :CURSor:MARKer:PERDt?
Example :CURSOR:MARKER:PERDt? → 16.667E+00
4.6 CURSor Group

:CURSor:MARKer:POSition<x>
Function Sets the X-axis value (physical value) of the marker position or queries the current setting.
Syntax :CURSor:MARKer:POSition<x> {<time>|<NRf>}
  :CURSor:MARKer:POSition<x>?
  <time> = 0 to (OBSERVATION TIME) (during the normal measurement mode, when Time Base = Internal)
  <NRf> = 0 to Record length (when Time Base = Internal, or during the harmonic measurement mode)
Example :CURSOR:MARKER:POSITION1 20MS
 :CURSOR:MARKER:POSITION1? → CURSOR:MARKER:POSITION1 20.000E-03
  Description • The range and resolution of <time> depends on the observation time.
  • Specify <NRf> in terms of sampled data points. The range is from 0 to the record length.

:CURSor:MARKer:TRACe<x>
Function Sets the waveform on which to place the marker or queries the current setting.
Syntax :CURSor:MARKer:TRACe<x> {<NRf>|MATH<x>}
  :CURSor:MARKer:TRACe<x>?
  TRACe<x>'s = 1, 2
  <NRf> = 1 to 8 (channel)
  <x> = 1, 2 (MATH)
Example :CURSOR:MARKER:TRACE1 1
   :CURSOR:MARKER:TRACE1? → :CURSOR:MARKER:TRACE1 1

:CURSor:MARKer:X<x>?
Function Queries the X-axis value (physical value) of the marker position.
Syntax :CURSor:MARKer:X<x>?
Example :CURSOR:MARKER:X1? → 20.000E-03
  Description • The command can be used to make the same inquiry.
  • "NAN (Not A Number)" will also be returned, if the "DISPLAY:FORMat" setting does not include the waveform display.

:CURSor:MARKer:Y<x>?
Function Queries the Y-axis value (physical value) of the marker position.
Syntax :CURSor:MARKer:Y<x>?
Example :CURSOR:MARKER:Y1? → 78.628E+00
  Description "NAN (Not A Number)" will be returned, if the "%CURSor:TYPE" is not set to MARKer.

:CURSor[:TYPE]
Function Sets the marker/cursor type or queries the current setting.
Syntax :CURSor[:TYPE] {OFF|MARKer|HORizontal|VERTical|HAVertical}
Example :CURSOR:TYPE HORIZONTAL

:CURSor:VERTical?
Function Queries all settings related to the V cursor.
Syntax :CURSOR:VERTical?
Example :CURSOR:VERTICAL?
  → :CURSOR:VERTICAL:TRACE 1;POSITION1 20.000E-03;
   POSITION2 80.000E-03

:CURSor:VERTical:DX?
Function Queries the X-axis value (physical value) between the V cursors.
Syntax :CURSOR:VERTical:DX?
Example :CURSOR:VERTICAL:DX?
  → 60.000E-03

:CURSor:VERTical:FFT<x>
Function Sets the V cursor position with respect to the FFT result or queries the current setting.
Syntax :CURSor:VERTical:FFT<x> {<frequency>|<NRf>}
  :CURSor:VERTical:FFT<x>?
  <x> = 1 to 2
  <frequency> = 0 to 2.5MHz (during the normal measurement mode, when Time Base = Internal)
  <NRf> = 0 to 5000 (when Time Base = External or during the harmonic measurement mode)
Example :CURSOR:VERTICAL:FFT1 200kHz
   :CURSOR:VERTICAL:FFT1? → :CURSOR:MARKER:FFT1 200.0E+03
  Description • This command is valid when :CURSOR:VERTical:TRACe<x> is set to MATH<x> and the equation of MATH<x> is set to FFT.
  • The range and resolution of <frequency> is determined from the sampling rate and the number of FFT points.
  • <NRf> is set in terms of harmonic rate. The range depends on the number of FFT points as follows. For the procedure to set the number of FFT points, see the "MATH<x>:FFT:POINt" command.
    For 1000 points : 0 to 500
    For 2000 points : 0 to 1000
    For 10000 points : 0 to 5000
4.6 CURSor Group

**:CURSor:VERTical:PERDi?**
Function Queries the 1/Δ value (physical value) of the horizontal axis between the V cursors.
Syntax :CURSor:VERTical:PERDi?
Example :CURSOR:VERTICAL:PERDi? \rightarrow 16.667E+00

**:CURSor:VERTical:POSition<x>**
Function Sets the V cursor position or queries the current setting.
Syntax :CURSor:VERTical:POSition<x> {<time>|<NRf>}
 :CURSor:VERTical:POSition<x>?
<time> = 0 to (OBSERVATION TIME) (during the normal measurement mode, when Time Base = Internal)
<NRf> = 0 to Record length (when Time Base = Internal, or during the harmonic measurement mode)
Example :CURSOR:VERTICAL:POSITION1 20MS
 :CURSOR:VERTICAL:POSITION1? \rightarrow CURSOR: VERTICAL: POSITION1 20.000E-03

Description • The range and resolution of <time> depends on the observation time.
• Specify <NRf> in terms of sampled data points. The range is from 0 to the record length.

**:CURSor:VERTical:TRACe**
Function Sets the waveform on which to place the V cursor or queries the current setting.
Syntax :CURSor:VERTical:TRACe {<NRf>|MATH<x>}
 :CURSor:VERTical:TRACe?
<NRf> = 1 to 8(channel)
<x> = 1, 2(MATH)
Example :CURSOR:VERTICAL:TRACE 1
 :CURSOR:VERTICAL:TRACE? \rightarrow CURSOR: VERTICAL: TRACE 1

**:CURSor:XY:X<x>?**
Function Queries the X-axis value (physical value) of the XY cursor position.
Syntax :CURSor:XY:X<x>?
Example :CURSOR:XY:X1? \rightarrow -75.000E+00

Description The “:DISPlay:XY:XTRace?” command can be used to make the same inquiry.

**:CURSor:XY:DX?**
Function Queries the X-axis value (physical value) between the XY cursors.
Syntax :CURSor:XY:DX7
Example :CURSOR:XY:DX7 \rightarrow 150.00E+00

**:CURSor:XY:POSition<x>**
Function Sets the XY cursor position or queries the current setting.
Syntax :CURSor:XY:POSition<x> {<NRf>}
 :CURSor:XY:POSition<x>?<NRf> = -100.0 to 100.0(%) (The resolution is 0.1(%) )
Example :CURSOR:XY:POSITION1 -25
 :CURSOR:XY:POSITION1? \rightarrow CURSOR:XY: POSITION1 -25.0

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

**:CURSor:XY:TRACe?**
Function Queries the waveform on which the XY cursor is placed.
Syntax :CURSor:XY:TRACe?
Example :CURSOR:XY:TRACE? \rightarrow CURSOR:XY:TRACE 1

Description The “:DISPlay:XY:XTRace?” command can be used to make the same inquiry.

**:CURSor:XY:X<x>?**
Function Queries the X-axis value (physical value) of the XY cursor position.
Syntax :CURSor:XY:X<x>?
Example :CURSOR:XY:X1? \rightarrow 75.000E+00
4.7 DISPlay Group

The commands in the DISPlay Group deal with the screen display. These commands can be used to make the same settings and inquiries as when the DISPLAY key on the front panel is pressed.
4.7 Display Group

:DISPLAY?
Function Queries all settings related to the screen display.
Syntax :DISPLAY?
Example When the display format (:DISPLAY:FORMat) is set to "NWAVe"
:DISPLAY?→:DISPLAY:FORMAT BOTH;(the response to "DISPLAY:NUMerical?" without the "DISPLAY:" section);(the same response to "DISPLAY:WAVE?");
:DISPLAY:DATE 1

:DISPLAY:BAR?
Function Queries all settings related to the bar graph display.
Syntax :DISPLAY:BAR?
Example :DISPLAY:BAR?
→ :DISPLAY:BAR:ITEM1 U,1;ITEM2 I,1;CURSOR1 1;CURSOR2 13;ORDER 1,100

:DISPLAY:BAR:CURSOR<x>
Function Sets the marker position (harmonic order) on the bar graph display or queries the current setting.
Syntax :DISPLAY:BAR:CURSOR<x> {<NRf>}
:DISPLAY:BAR:CURSOR<x>?
<x> = 1, 2
<NRf> = 0 to 500 (To the end harmonic order of the bar graph display)
Example :DISPLAY:BAR:CURSOR 1
:DISPLAY:BAR:CURSOR?→:DISPLAY:BAR:CURSOR1 1

:DISPLAY:BAR:ITEM<x>
Function Sets the bar graph display items (function, element) or queries the current setting.
Syntax :DISPLAY:BAR:ITEM<x> {<Function>,<Element>}
:DISPLAY:BAR:ITEM<x>?
<x> = 1 to 2 (item number)
<Function> = {U|I|P|S|Q|LAMBda|...} (See the function selection list on page 4-32 (3).)
Example :DISPLAY:BAR:ITEM1 U,1
:DISPLAY:BAR:ITEM1?
→ :DISPLAY:BAR:ITEM1 U,1

:DISPLAY:BAR:ORDER
Function Sets the start and end harmonic orders of the bar graph display or queries the current setting.
Syntax :DISPLAY:BAR:ORDER {<NRf>,<NRf>}
:DISPLAY:BAR:ORDER?
First <NRf> = 0 to 490 (start harmonic order of the bar graph display)
Second <NRf> = 10 to 500 (end harmonic order of the bar graph display)
Example :DISPLAY:BAR:ORDER 1,100
:DISPLAY:BAR:ORDER?→:DISPLAY:BAR:ORDER 1,100

Description • Set the start harmonic order, then the end harmonic order.
• Set the end harmonic order so that it is greater than or equal to (start harmonic order + 10).
4.7 **DISPlay Group**

### :DISPlay:DATE

**Function**
Turns ON/OFF the date and time displays or queries the current setting.

**Syntax**
```
:DISPlay:DATE {<Boolean>}
```

**Example**
:DISPlay:DATE ON

### :DISPlay:FORMat

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

**Syntax**
```
:DISPlay:FORMat {NUMeric|WAVE|XY|BAR|VECTor|NWAVe|NXY|NBAR|WXY|WBAR}
:DISPlay:FORMat?
```

**NUMeric** = Displays only the numerical values.

**WAVE** = Displays only the waveforms.

**XY** = Displays the X-Y display.

**BAR** = Displays the bar graph.

**VECTor** = Displays the vector graph.

**NWAVe** = Displays both the numerical values and the waveforms.

**NXY** = Displays both the numerical values and the X-Y display.

**NBAR** = Displays both the numerical values and the bar graph.

**WXY** = Displays both the waveforms and the X-Y display.

**WBAR** = Displays both the waveforms and the bar graph.

**Example**
:DISPlay:FORMAT NUMERIC

### :DISPlay:NUMeric?

**Function**
Queries all settings related to the numerical display.

**Syntax**
```
:DISPlay:NUMeric?
```

**Example**
• For normal measurement mode (when :SETup[:MODE] is set to “NORMal”)
  :DISPlay:NUMeric? → Same as the response for the “:DISPlay[NUMeric]:NORMal?” command.

• For harmonic analysis mode (when :SETup[:MODE] is set to “HARMonics”)
  :DISPlay:NUMeric? → Same as the response for the “:DISPlay[NUMeric]:HARMonics?” command.

### :DISPlay:[NUMeric]:HARMonics?

**Function**
Queries all settings related to the numerical display during harmonic measurement.

**Syntax**
```
:DISPlay[NUMeric]:HARMonics?
```

**Example**
• Example when the numerical display format is set to {8|16}
  :DISPlay[NUMeric]:HARMonics? →
  :DISPlay:NUMERIC:HARMONICS:IAMOUNT 8;
  ITEM1 U,1,1;ITEM2 I,1,1;ITEM3 P,1,1;
  ... (abbreviated)...;ITEM255 NONE;
  ICURSOR 1

• Example when the numerical display format is set to [SINGLE|DUAL] list display
  :DISPlay[NUMeric]:HARMonics? →
  :DISPlay:NUMERIC:HARMONICS:IAMOUNT SINGLE;LIST1 U,1;LIST2 I,1;
  LCURSOR 1

### :DISPlay:[NUMeric]:HARMonics:IAMount

**Function**
Queries all settings related to the numerical display during harmonic measurement.

**Syntax**
```
:DISPlay[NUMeric]:HARMonics:IAMount {<NRf>|SINGle|DUAL|SIGMa}
:DISPlay[NUMeric]:HARMonics:IAMount?
```

**<NRf>** = 8, 16

**Example**
:DISPlay[NUMeric]:HARMonics:IAMount 8:
:DISPlay:NUMERIC:HARMONICS:IAMOUNT? →
:DISPlay:NUMERIC:HARMONICS:IAMOUNT 8

**Description**
The harmonic measurement data information that is displayed depends on the selected numerical display format as follows.

- **<NRf>** = Displays the numerical display items in the order of item numbers. (<NRf> denotes the number of displayed items on one screen.)

  **SINGLE** = Displays a list of display items in EVEN and ODD columns.

  **DUAL** = Displays two lists of display items in the order of harmonic columns.

- **SIGMa** = Displays the numeric data of the main functions (U, I, P, S, Q, and λ...) and the phase difference (φ) between U and I for each element.
4.7 Display Group

:DISPLAY[:NUMeric]:HARMonics:ICURsor
Function
Sets the cursor position of the numerical display during harmonic measurement or queries the current setting.
Syntax
:DISPLAY[:NUMeric]:HARMonics:ICURsor {<NRf>}
:DISPLAY[:NUMeric]:HARMonics:ICURsor? <NRf> = 1 to 300
Example
:DISPLAY:NUMERIC:HARMONICS:ICURSOR 1
:DISPLAY:NUMERIC:HARMONICS:ICURSOR?
→
:DISPLAY:NUMERIC:HARMONICS:ICURSOR 1
Description
• The cursor position is specified using the item number.
• This command is valid when the numerical display format (:DISPLAY[:NUMeric]:HARMonics:IAMount) is set to {8|16}.

:DISPLAY[:NUMeric]:HARMonics:ITEM<x>
Function
Sets the numerical displayed items during harmonic measurement or queries the current setting.
Syntax
:DISPLAY[:NUMeric]:HARMonics:ITEM<x> {NONE|<Function>,<Element>,<Order>}
:DISPLAY[:NUMeric]:HARMonics:ITEM<x>? <x> = 1 to 255 (item number)
NONE = no display item
<Function> = {U|I|P|S|Q|...} (See the function selection list on page 4-32 (2).)
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)
<Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 500)
Example
:DISPLAY:NUMERIC:HARMONICS:ITEM1 U,1,1
:DISPLAY:NUMERIC:HARMONICS:ITEM17->
:DISPLAY:NUMERIC:HARMONICS:ITEM U,1,1
Description
This command is valid when the numerical display format (:DISPLAY[:NUMeric]:HARMonics:IAMount) is set to {SINGle|DUAL} list display.

:DISPLAY[:NUMeric]:HARMonics:LCURsor
Function
Sets the cursor position on the list display during harmonic measurement or queries the current setting.
Syntax
:DISPLAY[:NUMeric]:HARMonics:LCURsor {<Order>}
:DISPLAY[:NUMeric]:HARMonics:LCURsor? <Order> = {TOTal|DC|<NRf>} (<NRf> = 1 to 500)
Example
:DISPLAY:NUMERIC:HARMONICS:LCURSOR TOTAL
:DISPLAY:NUMERIC:HARMONICS:LCURSOR?
→
:DISPLAY:NUMERIC:HARMONICS:LCURSOR TOTAL
Description
• The cursor position is specified using the harmonic order.
• This command is valid when the numerical display format (:DISPLAY[:NUMeric]:HARMonics:IAMount) is set to {SINGle|DUAL|SIGMA} list display.

:DISPLAY[:NUMeric]:HARMonics:LIST<x>
Function
Sets the list display items during harmonic measurement or queries the current setting.
Syntax
:DISPLAY[:NUMeric]:HARMonics:LIST<x> {<Function>,<Element>}
:DISPLAY[:NUMeric]:HARMonics:LIST<x>? <x> = 1, 2 (item number)
<Function> = {U|I|P|S|Q|LAMBda|...} (See the function selection list on page 4-32 (3).)
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf> = 1 to 4)
Example
:DISPLAY:NUMERIC:HARMONICS:LIST1 U,1
:DISPLAY:NUMERIC:HARMONICS:LIST2 U,1
Description
This command is valid when the numerical display format (:DISPLAY[:NUMeric]:HARMonics:IAMount) is set to {SINGle|DUAL} list display.

:DISPLAY[:NUMeric]:HARMonics:PRESet
Function
Sets the numerical display items to a preset pattern during harmonic measurement.
Syntax
:DISPLAY[:NUMeric]:HARMonics:PRESet {<NRf>}
<NRf> = 1 to 4
Example
:DISPLAY:NUMERIC:HARMONICS:PRESET 1
Description
Regardless of what value (1 to 4) is specified for <NRf>, the display pattern (order) of the numerical display items will be the same as the display order when Reset Exec of the Display setting menu, which is displayed on the PZ4000 screen, is executed. For details related to the order of displayed items when reset is executed, see the PZ4000 User’s Manual.

:DISPLAY[:NUMeric]:NORMal?
Function
Queries all settings related to the numerical display during normal measurement.
Syntax
:DISPLAY[:NUMeric]:NORMal?
Example
• Example when the numerical display format (:DISPLAY[:NUMeric]:NORMal:IAMount) is set to "<NRf>" (split display)
:DISPLAY:NUMERIC:NORMAL? <NRf>
→
:DISPLAY:NUMERIC:NORMAL:IAMOUNT 8;ITEM1 URMS,1;ITEM2 UMN,1;ITEMS UDC,1,... (abbreviated)...ITEM255 NONE; ICURSOR 1
• Example when the numerical display format (:DISPLAY[:NUMeric]:NORMal:IAMount) is set to "ALL"
:DISPLAY:NUMERIC:NORMAL?->:DISPLAY:
:DISPLAY:NUMERIC:NORMAL:IAMOUNT ALL; FCURSOR URMS
4.7 Display Group

:DISPLAY[:NUMERIC]:NORMal:FCURsor
Function Sets the cursor position of the numerical display (All display) during normal measurement or queries the current setting.
Syntax :DISPLAY[:NUMERIC]:NORMall:FCURsor {<Function>}
:DISPLAY[:NUMERIC]:NORMall:FCURsor?
<Function> = {URMS|UMN|UDC|UAC|IRMS|...}
(See the function selection list on page 4-31 (1).)
Example :DISPLAY:NUMERIC:NORMAL:FCURSOR URMS
:DISPLAY:NUMERIC:NORMAL:FCURSOR?
→ :DISPLAY:NUMERIC:NORMAL:FCURSOR URMS
Description • The cursor position is specified using the function.
• This command is valid when the numerical display format (:DISPLAY[:NUMERIC]:HARMonics:IAMount)
  is set to ”ALL.”

:DISPLAY[:NUMERIC]:NORMal:IAMount
Function Sets the numerical display format during normal measurement or queries the current setting.
Syntax :DISPLAY[:NUMERIC]:NORMal:IAMount {<NRf>|ALL}
:DISPLAY[:NUMERIC]:NORMal:IAMount?
<NRf> = 8, 16, 42, 78
Example :DISPLAY:NUMERIC:NORMAL:IAMOUNT 8
:DISPLAY:NUMERIC:NORMAL:IAMOUNT?
→ :DISPLAY:NUMERIC:NORMAL:IAMOUNT 8
Description The displayed measurement data depend on the selected numerical display format as follows.
<NRf> = Displays the numerical display items in the order of item numbers. (<NRf> denotes the number of displayed items on one screen.)
ALL = Displays all functions in order for each element.

:DISPLAY[:NUMERIC]:NORMal:ICURsor
Function Sets the cursor position of the numerical display (split display) during normal measurement or queries the current setting.
Syntax :DISPLAY[:NUMERIC]:NORMal:ICURsor {<NRf>}
:DISPLAY[:NUMERIC]:NORMal:ICURsor?
<NRf> = 1 to 300
Example :DISPLAY:NUMERIC:NORMAL:ICURSOR 1
:DISPLAY:NUMERIC:NORMAL:ICURSOR?
→ :DISPLAY:NUMERIC:NORMAL:ICURSOR 1
Description • The cursor position is specified using the item number.
• This command is valid when the numerical display format (:DISPLAY[:NUMERIC]:HARMonics:IAMount)
  is set to <NRf> (split display).

:DISPLAY[:NUMERIC]:NORMal:ITEM<x>
Function Sets the numerical displayed item during normal measurement or queries the current setting.
Syntax :DISPLAY[:NUMERIC]:NORMall:ITEM<x> {NONE|<Function>,<Element>}
:DISPLAY[:NUMERIC]:NORMall:ITEM<x>?
<x> = 1 to 255(item number)
NONE = no display item
<Function> = {URMS|UMN|UDC|UAC|IRMS|...}
(See the function selection list on page 4-31 (1).)
Example :DISPLAY:NUMERIC:NORMAL:ITEM1 URMS,1
:DISPLAY:NUMERIC:NORMAL:ITEM1?
→ :DISPLAY:NUMERIC:NORMAL:ITEM1 URMS,1
Description This command is valid when the numerical display format (:DISPLAY[:NUMERIC]:HARMonics:IAMount)
  is set to <NRF> (split display).

:DISPLAY[:NUMERIC]:NORMal:PRESet
Function Sets the numerical display items to a preset pattern during normal measurement.
Syntax :DISPLAY[:NUMERIC]:NORMal:PRESet {<NRf>}
<NRf> = 1 to 4
Example :DISPLAY:NUMERIC:NORMAL:PRESET 1
Description Regardless of what value (1 to 4) is specified for <NRf>, the display pattern (order) of the numerical display items will be the same as the display order when Reset Exec of the Display setting menu, which is displayed on the PZ4000 screen, is executed. For details related to the order of displayed items when reset is executed, see the PZ4000 User’s Manual.

:DISPLAY:VECTor?
Function Queries all settings related to the vector display.
Syntax :DISPLAY:VECTor?
Example :DISPLAY:VECTOR?
→ :DISPLAY:VECTOR:
NUMERIC 1;UMAG 1.000;IMAG 1.000

:DISPLAY:VECTor:IMAG
Function Sets the zoom factor of the current display during vector display or queries the current setting.
Syntax :DISPLAY:VECTor:IMAG {<NRf>}
<NRf> = 0.100 to 100,000
Example :DISPLAY:VECTOR:IMAG 1
:DISPLAY:VECTOR:IMAG?
→ :DISPLAY:VECTOR:IMAG 1.000

:DISPLAY:VECTor:IMAG
Function Sets the zoom factor of the current display during vector display or queries the current setting.
Syntax :DISPLAY:VECTor:IMAG {<NRf>}
<NRf> = 0.100 to 100,000
Example :DISPLAY:VECTOR:IMAG 1
:DISPLAY:VECTOR:IMAG?
→ :DISPLAY:VECTOR:IMAG 1.000
4.7 Display Group

**:DISPlay:VECTor:NUMeric**

Function: Turns on/off the numerical data display during vector display or queries the current setting.

Syntax: :

Example:

**:DISPlay:VECTor:UMAG**

Function: Sets the zoom factor of the voltage display during vector display or queries the current setting.

Syntax: :

Example:

**:DISPlay:WAVE?**

Function: Queries all settings related to the waveform display.

Syntax: :

Example:

**:DISPlay:WAVE:{CHANnel<x>|MATH<x>}**

Function: Turns on/off the channel/computed waveform display or queries the current setting.

Syntax: :

Example:

**:DISPlay:WAVE:GRATicule**

Function: Sets the graticule type (grid) or queries the current setting.

Syntax: :

Example:

**:DISPlay:WAVE:INTERpolate**

Function: Sets the interpolation method of the waveform or queries the current setting.

Syntax: :

Example:

**:DISPlay:WAVE:MAPPing?**

Function: Queries all settings related to the waveform mapping to the split screen.

Syntax: :

Example:

**:DISPlay:WAVE:MAPPing:{CHANnel<x>|MATH<x>}**

Function: Sets the channel waveform/MATH waveform mapping to the split screen or queries the current setting.

Syntax: :

Example:

Description: This command is valid when the waveform mapping method (:DISPlay:WAVE:MAPPing[:MODE]) is set to "USER."

**:DISPlay:WAVE:FORMat**

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

Syntax: :

Example:
<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay:WAVE:SVALue</td>
<td>Turns ON/OFF the scale value display or queries the current setting.</td>
<td>Syntax: :DISPlay:WAVE:SVALue {&lt;Boolean&gt;}</td>
<td>Example: :DISPlay:WAVE:SVALue OFF</td>
</tr>
<tr>
<td>:DISPlay:WAVE:TLABel</td>
<td>Turns ON/OFF the waveform label display or queries the current setting.</td>
<td>Syntax: :DISPlay:WAVE:TLABel {&lt;Boolean&gt;}</td>
<td>Example: :DISPlay:WAVE:TLABel ON</td>
</tr>
</tbody>
</table>

Description:
- The waveform labels can be set using the ":CHANnel<x>:LABel" command.
- The range and resolution of <Frequency> is determined from the sampling rate and the number of FFT points.
- <NRf> is set in terms of harmonic order. The range depends on the number of FFT points as follows. For the procedure to set the number of FFT points, see the ":MATH<x>:FFT:POINt" command. For 1000 points: 0 to 500 For 2000 points: 0 to 1000 For 10000 points: 0 to 5000
- The range of FFT waveform is set by the "<frequency>" parameter and the equation of "MATH<x>" is set to FFT.
- The command is valid when ":DISPlay:XY:XTRace" is set to "MATH<x>" and the equation of "MATH<x>" is set to FFT.
### :DISPLAY:XY:POSITION

**Function**
Sets the range of the T-Y waveform to be displayed on the X-Y display or queries the current setting.

**Syntax**
:DISPLAY:XY:POSITION {<time>,
<time>|<NRf>,<NRf>}:DISPLAY:XY:POSITION?

- `<time>` = 0 to (OBSERVATION TIME) (during the normal measurement mode, when Time Base = Internal)
- `<NRf>` = 0 to (Record Length) (when Time Base = External or during the harmonic measurement mode)

**Example**
:DISPLAY:XY:POSITION 0.0,80MS
:DISPLAY:XY:POSITION? -> :DISPLAY:XY:POSITION 0.000E-03,80.000E-03

**Description**
- Set the start point first and then the end point.
- The range and resolution of `<time>` depend on the observation time.
- When using `<NRf>`, specify using the number of sampling data points. The range is from 0 to (record length).

### :DISPLAY:XY:XTRACE

**Function**
Sets the channel to assign to the X-axis of the X-Y display or queries the current setting.

**Syntax**
:DISPLAY:XY:XTrace {<NRf>|MATH<x>}

- `<NRf>` = 1 to 8 (channel)
- `<x>` = 1, 2 (MATH)

**Example**
:DISPLAY:XY:XTRACE 1

---

* Function selection (<Function>) list

(1) Functions in the normal measurement mode

**Applicable commands**
:DISPLAY[:NUMerical]:NORMAL:FCURsor
:DISPLAY[:NUMerical]:NORMAL:ITEM<x>

**Selection used in the communications menu (numerical display header name)**

- **URMS**: Urms
- **UMN**: Umean
- **UDC**: Udc
- **UAC**: Uac
- **IRMS**: Irms
- **IMN**: Imean
- **IDC**: Idc
- **IAC**: Iac
- **P**: P
- **S**: S
- **Q**: Q
- **LAMBda**: λ
- **PHI**: φ
- **FU**: FreqU (fU)
- **FI**: FreqI (fI)
- **UPPeak**: U+peak (U+pk)
- **UMPeak**: U-peak (U-pk)
- **IPPeak**: I+peak (I+pk)
- **IMPeak**: I-peak (I-pk)
- **CFU**: CfU
- **CFI**: CfI
- **FFU**: FfU
- **FFI**: FfI
- **Z**: Z
- **RS**: Rs
- **XS**: Xs
- **RP**: Rp
- **XP**: Xp
- **PC**: Pc
- **ETA**: η
- **SETA**: 1/η
- **F1**: F1
- **F2**: F2
- **F3**: F3
- **F4**: F4
- **DURMS**: ΔUrms
- **DUMN**: ΔUmean
- **DUDC**: ΔUdc
- **DUAC**: ΔUac
- **DIRMS**: ΔIrms
- **DIMN**: ΔImean
- **DIDC**: ΔIdc
- **DIAC**: ΔIac
- **SPEed**: Speed
- **TORQue**: Torque
- **SYNC**: SyncSpd
- **SLIP**: Slip
- **PM**: Pm
(2) Functions in the harmonic measurement mode
Applicable commands
:DISPLAY[:NUMerics]:HARMonics:ITEM<x>
Selection used in: Function name used in the communications menu (numerical display header name)

U : U
I : I
P : P
S : S
Q : Q
LAMBda : \lambda
PHI : \phi
PHIU : \phi U
PHII : \phi I
FU : FreqU (fU)
FI : FreqI (fI)
Z : Z
RS : Rs
XS : Xs
RP : Rp
XP : Xp
UHDF : Uhdf
IHDF : Ihdf
PHDF : Phdf
SHDF : Shdf
QHDF : Qhdf
UTHD : Uthd
ITHD : Ithd
PTHD : Pthd
STHD : Sthd
QTHD : Qthd
UTHF : Uthf
ITHF : Ithf
UTIF : Utif
ITIF : Itif
HVF : hvf
HCF : hcf
ETA : \eta
SETA : \frac{1}{\eta}
PHI_U1U2 : \eta U1-U2
PHI_U1U3 : \eta U1-U3
PHI_U1I1 : \eta U1-I1
PHI_U1I2 : \eta U1-I2
PHI_U1I3 : \eta U1-I3
F1 : F1
F2 : F2
F3 : F3
F4 : F4
SPEed : Speed

* SPEed, TORQue, SYNC, SLIP, PM, MAETa, and MBETa are applicable when the motor module is installed.

(3) Functions in the harmonic measurement mode (list display)
Applicable commands
:DISPLAY[:NUMerics]:HARMonics:LIST<x>
:DISPLAY:BAR:ITEM<x>
:FILE:SAVE:NUMerics:LIST
Selection used in: Function name used in the communications menu (numerical display header name)

U : U
I : I
P : P
S : S
Q : Q
LAMBda : \lambda
PHI : \phi
PHIU : \phi U
PHII : \phi I
Z : Z
RS : Rs
XS : Xs
RP : Rp
XP : Xp
TORQue : Torque

* TORQue is applicable when the motor module is installed.
4.8 FILE Group

The commands in the FILE Group deal with file operations. These commands can be used to make the same settings and inquiries as when the FILE key on the front panel is pressed.
4.8 FILE Group

:FILE?
Function Queries all settings related to file operations.
Syntax :FILE?
Example :FILE?
→

:FILE:CDIRectory
Function Changes the current directory.
Syntax :FILE:CDIRectory {<Filename>}
<Filename> = directory name
Example :FILE:CDIRECOTRY "IMAGE"
Description Specify ".." to move to a higher directory.

:FILE:DELETE:IMAGE:{TIFF|BMP|PSCRipt}
Function Deletes a screen image data file.
Syntax :FILE:DELETE:IMAGE:{TIFF|BMP|PSCRipt} {<Filename>}
Example :FILE:DELETE:IMAGE:TIFF "IMAGE1"

:FILE:DELETE:NUMeric:{ASCii|FLOat}
Function Deletes a numerical data file.
Syntax :FILE:DELETE:NUMeric:{ASCii|FLOat} {<Filename>}
Example :FILE:DELETE:NUMERIC:ASCii "NUM1"

:FILE:DELETE:SETup
Function Deletes a setup parameter file.
Syntax :FILE:DELETE:SETup {<Filename>}
Example :FILE:DELETE:SETup "SETUP1"

:FILE:DELETE:WAVE:{BINary|ASCii|FLOat}
Function Deletes a waveform data file.
Syntax :FILE:DELETE:WAVE:{BINary|ASCii|FLOat} {<Filename>}
Example :FILE:DELETE:WAVE:BINARY "WAVE1"

:FILE:DRIVE
Function Sets the drive (medium) setting.
Syntax :FILE:DRIVE {FD0|SCSI,<NRf>[,<NRf>]} First <NRf> = SCSI address (0 to 7) Second <NRf> = partition (0 to 5)
Example :FILE:DRIVE FD0
Description If you are using a drive that has no partitions set, omit the second <NRf>.

:FILE:FORMAT
Function Formats the floppy disk.
Syntax :FILE:FORMAT {DD64|DD72|HD12|HD14}
Example :FILE:FORMAT HD14

:FILE:FREE?
Function Queries the free space (bytes) on the drive.
Syntax :FILE:FREE?
Example :FILE:FREE? → 163840

:FILE:LOAD:ABORT
Function Aborts loading a file.
Syntax :FILE:LOAD:ABORT
Example :FILE:LOAD:ABORT
4.8 FILE Group

**:FILE:LOAD:SETup**
Function Loads a setup parameter file.
Syntax :FILE:LOAD:SETup {<Filename>}
Example :FILE:LOAD:SETUP "SETUP1"

**:FILE:LOAD:WAVE**
Function Loads a waveform data file.
Syntax :FILE:LOAD:WAVE {<Filename>}
Example :FILE:LOAD:WAVE "WAVE1"
Description Only waveform data in binary format can be loaded.

**:FILE:MDIRectory**
Function Creates a directory.
Syntax :FILE:MDIRectory {<Filename>}
Example :FILE:MDIRECTORY "TEST"

**:FILE:PATH?**
Function Queries the absolute path of the current directory.
Syntax :FILE:PATH?
Example :FILE:PATH? → "FD0\IMAGE"

**:FILE:SAVE?**
Function Queries all settings related to saving a file.
Syntax :FILE:SAVE?

**:FILE:SAVE:ABORT**
Function Aborts saving the file.
Syntax :FILE:SAVE:ABORT
Example :FILE:SAVE:ABORT

**:FILE:SAVE:ANAMing**
Function Sets whether or not to automatically assign file names or queries the current setting.
Syntax :FILE:SAVE:ANAMing {<Boolean>}
Example :FILE:SAVE:ANAMING ON

**:FILE:SAVE:COMMent**
Function Sets the comment that is attached to the file being saved or queries the current setting.
Syntax :FILE:SAVE:COMMent {<string>}
Example :FILE:SAVE:COMMENT "CASE1"

**:FILE:SAVE:NUMeric?**
Function Queries all settings related to saving the numerical data to a file.
Syntax :FILE:SAVE:NUMeric?

**:FILE:SAVE:NUMeric[EXECute]**
Function Saves the numerical data to a file.
Syntax :FILE:SAVE:NUMeric[] EXECute {<Filename>}
Example :FILE:SAVE:NUMERIC:EXECUTE "NUM1"

**:FILE:SAVE:NUMeric:LIST?**
Function Queries all settings related to saving the numerical data list to a file during harmonic measurement.
Syntax :FILE:SAVE:NUMeric:LIST?
Example :FILE:SAVE:NUMERIC:LIST? → :FILE:SAVE:NUMERIC:LIST:ELEMENT1 1; ELEMENT2 0; ELEMENT3 0; ELEMENT4 0; U 1; I 0; P 0; S 0; Q 0; LAMBDa 0; PHI 0; PHIU 0; PHII 0; Z 0; RS 0; XS 0; RP 0; XP 0; SIGMA 0

**:FILE:SAVE:NUMeric:LIST:ELEMent<x>**
Function Turns ON/OFF the output of each element when saving numerical list data to a file during harmonic measurement or queries the current setting.
Syntax :FILE:SAVE:NUMeric:LIST:ELEMent<x> {<Boolean>}
Example :FILE:SAVE:NUMERIC:LIST:ELEMENT1 ON

**:FILE:SAVE:NUMeric:LIST:{<List-Function>|SIGma}**
Function Turns ON/OFF the output of each function when saving numerical list data to a file during harmonic measurement or queries the current setting.
Syntax :FILE:SAVE:NUMERIC:LIST:{<List-Function>|SIGma} {<Boolean>}
Example :FILE:SAVE:NUMERIC:LIST:{<List-Function>|SIGma} {<Boolean>}
Example :FILE:SAVE:NUMERIC:LIST:{<List-Function>|SIGma}? List-Function = [U|I|P|S|Q|LAMBDa|... ] (See the function selection list on page 4-32 (3).)
Example :FILE:SAVE:NUMERIC:LIST:U ON
4.8 FILE Group

**:FILE:SAVE:NUMeric:TYPE**

Function  Sets the format of the numerical data being saved or queries the current setting.

Syntax  **:FILE:SAVE:NUMeric:TYPE** {ASCii|FLOat}

Example  **:FILE:SAVE:NUMeric:TYPE** FLOAT

**:FILE:SAVE:WAVE?**

Function  Queries all settings related to saving the waveform data to a file.

Syntax  **:FILE:SAVE:WAVE?**

Example  **:FILE:SAVE:WAVE?**

Description  This command is valid when the format of the waveform data being saved (**:FILE:SAVE:WAVE:TYPE**) is set to {FLoat}. If it is set to {BINary|ASciii}, then all waveforms that are turned ON will be selected.
4.9 HCOPy Group

The commands in the HCOPy Group deal with the output of screen data to the built-in printer (option) or other devices. These commands can be used to make the same settings and inquiries as when the COPY or MENU (SHIFT+COPY) key on the front panel is pressed.
### 4.9 HCOPy Group

#### :HCOPy?
**Function**
Queries all settings related to screen data output.

**Syntax**
:HCOPy?

**Example**
:HCOPY? → :HCOPY:DIRECTION PRINTER; COMMENT "THIS IS TEST."

#### :HCOPY:ABORT
**Function**
Aborts data output and paper feeding.

**Syntax**
:HCOPY:ABORT

**Example**
:HCOPY:ABORT

#### :HCOPY:CENTronics?
**Function**
Queries all settings related to the external printer output.

**Syntax**
:HCOPY:CENTronics?

**Example**
:HCOPY:CENTRONICS?

```
:HCOPY:CENTRONICS:FORMAT ESCP2;COLOR 0
```

#### :HCOPY:CENTronics:COLOR
**Function**
Sets the color (ON/OFF) of the external printer output or queries the current setting.

**Syntax**
:HCOPY:CENTronics:COLOR {<Boolean>}

**Example**
:HCOPY:CENTRONICS:COLOR OFF

### Description
Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.

#### :HCOPY:CENTronics:FORMAT
**Function**
Sets the command format that is output to the printer or queries the current setting.

**Syntax**
:HCOPY:CENTronics:FORMAT {ESCP|ESCP2|LIPS3|PR201|PCL5|BJ}

**Example**
:HCOPY:CENTRONICS:FORMAT ESCP2

### Description
This command is void when the data output destination (:HCOPY:DIRection) is set to {PRINter|CENTronics}.

#### :HCOPY:PRINTER:DLIST
**Function**
Outputs the numerical data list to the built-in printer. This is an overlap command.

**Syntax**
:HCOPY:PRINTER:DLIST

**Example**
:HCOPY:PRINTER:DLIST

#### :HCOPY:PRINTER:FEED
**Function**
Feeds the paper (built-in printer). This is an overlap command.

**Syntax**
:HCOPY:PRINTER:FEED

**Example**
:HCOPY:PRINTER FEED

#### :HCOPY:SAVE?
**Function**
Queries all settings related to saving the file.

**Syntax**
:HCOPY:SAVE?

**Example**
:HCOPY:SAVE? → :HCOPY:SAVE:ANAMING 1; NAME "DATA1"; COMMENT "CASE1"

#### :HCOPY:SAVE:ANAMing
**Function**
Sets whether or not to automatically assign file names or queries the current setting.

**Syntax**
:HCOPY:SAVE:ANAMing {<Boolean>}

**Example**
:HCOPY:SAVE:ANAMING ON

### Description
Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.
:HCOPY:SAVE:COMMENT
Function
Sets the comment that is attached to the file being saved or queries the current setting.
Syntax
:HCOPY:SAVE:COMMENT {<string>}
:HCOPY:SAVE:COMMENT?
Example
:HCOPY:SAVE:COMMENT "CASE1"
:HCOPY:SAVE:COMMENT?
→ :HCOPY:SAVE:COMMENT "CASE1"
Description
Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.

:HCOPY:SAVE:NAME
Function
Sets the file name or queries the current setting.
Syntax
:HCOPY:SAVE:NAME {<Filename>}
:HCOPY:SAVE:NAME?
Example
:HCOPY:SAVE:NAME "DATA1"
:HCOPY:SAVE:NAME?
→ :HCOPY:SAVE:NAME "DATA1"
Description
The save destination of the screen data is specified using:
• the ":FILE:DRIVE" command for the drive.
• the ":FILE:DIRECTORY" command for the directory.
The save destination path can be queried using the ":FILE:PATH?" command.

:HCOPY:{TIFF|BMP}?
Function
Queries all settings related to the TIFF/BMP format.
Syntax
:HCOPY:{TIFF|BMP}?
Example
:HCOPY:TIFF?
→ :HCOPY:TIFF:COLOR COLOR; COMPRESSION 0

:HCOPY:{TIFF|BMP}:COLOR
Function
Sets the color for the TIFF/BMP format or queries the current setting.
Syntax
:HCOPY:{TIFF|BMP}:COLOR {OFF|COLOR|REVerse}
:HCOPY:{TIFF|BMP}:COLOR?
Example
:HCOPY:TIFF:COLOR COLOR
:HCOPY:TIFF:COLOR?
→ :HCOPY:TIFF:COLOR COLOR
4.10 IMAGe Group

The commands in the IMAGe Group deal with the output of screen image data. There are no front-panel keys that correspond to the commands in this group.

:IMAGe?
Function Queries all settings related to the output of the screen image data.
Syntax :IMAGe?
Example :IMAGE? → :IMAGE:FORMAT TIFF; COLOR OFF

:IMAGe:COLOR
Function Sets the color of the screen image data being output or queries the current setting.
Syntax :IMAGE:COLOR {OFF|COLOR|REVerse}
Example :IMAGE:COLOR OFF
:IMAGE:COLOR?
Example :IMAGE:COLOR OFF

:IMAGe:FORMAT
Function Sets the output format of the screen image data or queries the current setting.
Syntax :IMAGE:FORMAT {TIFF|BMP}
Example :IMAGE:FORMAT TIFF
:IMAGE:FORMAT?
Example :IMAGE:FORMAT? → :IMAGE:FORMAT TIFF

:IMAGe:SEND?
Function Queries the screen image data.
Syntax :IMAGE:SEND?
Example :IMAGE:SEND? → #6 (Number of bytes, 6 digits) (Series of data bytes)
Description • The number of bytes in <block data> is \(2 + 6 + \text{number of data} + 1\) (delimiter)).
• For information about block data, see page 3-6.
4.11 INPut Group

The commands in the INPut Group deal with the measurement conditions of each input module. These commands can be used to make the same settings and inquiries as when the INPUT key on the front panel is pressed.

```
:INPut MODUle?

POWer: VOLT age: RANGe: ALL

AUTO ELEMent <NRf>

AUTO?

AUTO: ALL OFF ON <NRf>

ELEMent <x> OFF ON <NRf>

CURRent: TERMinal: ALL

SENSor ELEMent <x> SENSor?

RANGe: ALL

AUTO ELEMent <x> AUTO?

AUTO: ALL OFF ON <NRf>

ELEMent <x> OFF ON <NRf>

SRATio: ALL

ELEMent <x> SENSor <NRf>
```

4.11 INPut Group

4-43 Commands
4.11 INPut Group

**[:INPut]?**

Function Queries all settings related to all input modules.
Syntax [:INPut]?
Example • When the motor module is not installed
:INPut? → (Same as the response to "[:INPut]:POWer?")
When the motor module is installed
:INPut? → (Same as the response to "[:INPut]:MOTor?")

**[:INPut]:MODUle?**

Function Queries the model name of each input module.
Syntax [:INPut]:MODUle? |<NRf>|
Example :INPut:MODUle? 1 → 253751
:INPut:MODUle? 2 → 253751, 253752, 253752, 253771

Description • The following responses are possible.
253751 = Power measurement module (1000 V/5 A)
253752 = Power measurement module (1000 V/20&5 A)
253771 = Motor module (Speed/Torque)
0 = no module
• If the parameter is omitted, the model name of each input module is returned for all elements in order starting with element 1.

**[:INPut]:MOTor?**

Function Queries all settings related to the motor module.
Syntax [:INPut]:MOTor? |
Example :INPut:MOTor? → :INPut:MOTor:SPEED:
RANGE 50.0E+00;TYPE ANALOG;
FRANGE 200.00E+03;PULSE 60;
SCALING 1.0000;UNIT "rpm";:INPut:MOTor:TORQUE:
RANGE 50.0E+00;SCALING 1.0000;
UNIT "Nm";:INPut:MOTor:PM:
SCALING 1.0000;UNIT "W";:INPut:MOTor:FILTER:
LINE OFF;ZCROSS OFF;:INPut:MOTor:POLE 2;
SYNCHRONIZE 2

Description • If the 253771 motor module is not installed, an error will occur.

**[:INPut]:MOTor:FILTer:[LINE]**

Function Sets the line filter for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:FILTer:[LINE] {OFF|<frequency>}
Example :INPut:MOTor:FILTer:LINE OFF

Description If the 253771 motor module is not installed, an error will occur.

**[:INPut]:MOTor:FILTer:ZCRoss**

Function Sets the zero crossing filter for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:FILTer:ZCRoss {OFF|<frequency>}
Example :INPut:MOTor:FILTer:ZCRoss OFF

Description If the 253771 motor module is not installed, an error will occur.

**[:INPut]:MOTor:PM?**

Function Queries all settings related to the motor output for the motor module.
Syntax [:INPut]:MOTor:PM? |
Example :INPut:MOTor:PM? → :INPut:MOTor:PM:
SCALING 1.0000;UNIT "W"

Description If the 253771 motor module is not installed, an error will occur.

**[:INPut]:MOTor:PM:SCALing**

Function Sets the scaling factor used during motor output computation on the motor module or queries the current setting.
Syntax [:INPut]:MOTor:PM:SCALing {<NRf>}
Example :INPut:MOTor:PM:SCALing 1
:INPut:MOTor:PM:SCALing? → :INPut:MOTor:PM:SCALing 1.0000

Description If the 253771 motor module is not installed, an error will occur.
[:INPut]:MOTor:PM:UNIT
Function Sets the unit to add to the motor output computation result or queries the current setting.
Syntax [:INPut]:MOTor:PM:UNIT {<string>}
[:INPut]:MOTor:PM:UNIT?
Example :INPUT:MOTOR:PM:UNIT "W"
[:INPut]:MOTor:PM:UNIT?
Description • Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.
• This command never affects the computation result.
• If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:POLE
Function Sets the motor’s number of poles for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:POLE {<NRf>}
[:INPut]:MOTor:POLE?
Example :INPUT:MOTOR:POLE 2
[:INPut]:MOTor:POLE?→:INPUT:MOTOR:POLE 2
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:SPEed?
Function Queries all settings related to the revolution sensor signal input for the motor module.
Syntax [:INPut]:MOTor:SPEed?
Example :INPUT:MOTOR:SPEED?→:INPUT:MOTOR:SPEED: RANGE 50.0E+00;TYPE ANALOG;
FRANGE 200.00E+03;PULSE 60;
SCALING 1.0000;UNIT "Nm"
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:SPEed:FRANge
Function Sets the frequency range of the revolution sensor signal input (pulse input) for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SPEed:FRANge {<frequency>|AUTO}
[:INPut]:MOTor:SPEed:FRANge?
Example :INPUT:MOTOR:SPEED:FRANGE 200KHZ
[:INPut]:MOTor:SPEed:FRANge?→:INPUT:MOTOR:SPEED:FRANGE 200.00E+03
Description • Set the <Frequency> to the maximum value within the frequency range.
• This command is valid when the input format of the revolution sensor signal ([::INPut]:MOTor:SPEed:TYPE) is set to "PULSe (pulse input)."
• If the 253771 motor module is not installed, an error will occur.
• The "::CHANnel7:SPEed:FRANge" command can be used to make the same settings and inquiries.

[:INPut]:MOTor:SPEed:PULSe
Function Sets the pulse count of the revolution sensor signal input (pulse input) for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SPEed:PULSe {<NRf>}
[:INPut]:MOTor:SPEed:PULSe?
Example :INPUT:MOTOR:SPEED:PULSE 60
[:INPut]:MOTor:SPEed:PULSe?→:INPUT:MOTOR:SPEED:PULSE 60
Description • This command is valid when the input format of the revolution sensor signal ([::INPut]:MOTor:SPEed:TYPE) is set to "PULSe (pulse input)."
• If the 253771 motor module is not installed, an error will occur.
4.11 INPut Group

[:INPut]:MOTor:SPEed:RANGe
Function Sets the voltage range of the revolution sensor signal input for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SPEed:RANGe {<voltage>|AUTO}
[:INPut]:MOTor:SPEed:RANGe?
<voltage> = 1, 2, 5, 10, 20, and 50(V)
AUTO = Auto range
Example :INPUT:MOTOR:SPEED:RANGE 50V
:INPUT:MOTOR:SPEED:RANGE?
→ :INPUT:MOTOR:SPEED:RANGE 50.0E+00
Description • When the input format of the revolution sensor signal ([:INPut]:MOTor:SPEed:TYPE) is set to "PULSe (pulse input)," it is fixed to 5 (V).
• If the 253771 motor module is not installed, an error will occur.
• The ":CHANne17:SPEed:RANGe" command can be used to make the same settings and inquiries.

[:INPut]:MOTor:SPEed:SCALing
Function Sets the scaling factor used during rotating speed computation on the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SPEed:SCALing {<NRf>}
[:INPut]:MOTor:SPEed:SCALing?
<NRf> = 0.0001 to 99999.9999
Example :INPUT:MOTOR:SPEED:SCALING 1
:INPUT:MOTOR:SPEED:SCALING?
→ :INPUT:MOTOR:SPEED:SCALING 1.0000
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:SPEed:TYPE
Function Sets the input type of the revolution sensor signal input for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SPEed:TYPE {ANALog|PULSe}
[:INPut]:MOTor:SPEed:TYPE?
Example :INPUT:MOTOR:SPEED:TYPE ANALOG
:INPUT:MOTOR:SPEED:TYPE?
→ :INPUT:MOTOR:SPEED:TYPE ANALOG
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:SYNChronize
Function Sets the frequency measurement source for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:SYNChronize {<NRf>}
[:INPut]:MOTor:SYNChronize?
<NRf> = 1 to 8
Example :INPUT:MOTOR:SYNCHRONIZE 2
:INPUT:MOTOR:SYNCHRONIZE?
→ :INPUT:MOTOR:SYNCHRONIZE 2
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:TORQue:
Function Queries all settings related to the torque meter signal input for the motor module.
Syntax [:INPut]:MOTor:TORQue?
Example :INPUT:MOTOR:TORQUE?
→ :INPUT:MOTOR:TORQUE:RANGe 50.0E+00;SCALING 1.0000;UNIT "Nm"
Description If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:TORQue:RANGe
Function Sets the voltage range of the torque meter signal input for the motor module or queries the current setting.
Syntax [:INPut]:MOTor:TORQue:RANGe {<voltage>|AUTO}
[:INPut]:MOTor:TORQue:RANGe?
<voltage> = 1, 2, 5, 10, 20, and 50(V)
AUTO = Auto range
Example :INPUT:MOTOR:TORQUE:RANGe 50V
:INPUT:MOTOR:TORQUE:RANGe?
→ :INPUT:MOTOR:TORQUE:RANGe 50.0E+00
Description • If the 253771 motor module is not installed, an error will occur.
• The ":CHANne18:TORQue:RANGe" command can be used to make the same settings and inquiries.
[:INPut]:MOTor:TORQue:SCALing

Function  Sets the scaling factor used during torque computation on the motor module or queries the current setting.

Syntax  [:INPut]:MOTor:TORQue:SCALing {<NRf>}
        [:INPut]:MOTor:TORQue:SCALing?
        <NRf> = 0.0001 to 99999.9999

Example  :INPUT:MOTOR:TORQUE:SCALING 1
         :INPUT:MOTOR:TORQUE:SCALING?
         → :INPUT:MOTOR:TORQUE:SCALING 1.0000

Description  If the 253771 motor module is not installed, an error will occur.

[:INPut]:MOTor:TORQue:UNIT

Function  Sets the unit to add to the torque computation result or queries the current setting.

Syntax  [:INPut]:MOTor:TORQue:UNIT {<string>}
        [:INPut]:MOTor:TORQue:UNIT?
        <string> = 8 characters or less

Example  :INPUT:MOTOR:TORQUE:UNIT "Nm"
         :INPUT:MOTOR:TORQue:UNIT?
         → :INPUT:MOTOR:TORQUE:UNIT "Nm"

Description  • Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.
• This command never affects the computation result.
• If the 253771 motor module is not installed, an error will occur.

[:INPut]:POWer?

Function  Queries all settings related to the power measurement module.

Syntax  [:INPut]:POWer?

Example  :INPUT:POWER?→:INPUT:POWER:VOLTAGE:
          RANGE:ELEMENT1 2.00E+03;
          ELEMENT2 2.00E+03;ELEMENT3 2.00E+03;
          INPUT:POWER:CURRENT:TERMINAL:
          ELEMENT1 5.0E+00;ELEMENT2 5.0E+00;
          ELEMENT3 5.0E+00;ELEMENT4 5.0E+00;
          INPUT:POWER:CURRENT:RANGE:
          ELEMENT1 10.0E+00;ELEMENT2 10.0E+00;
          ELEMENT3 10.0E+00;ELEMENT4 10.0E+00;
          INPUT:POWER:CURRENT:SRATIO:
          ELEMENT1 10.0000;ELEMENT2 10.0000;
          ELEMENT3 10.0000;ELEMENT4 10.0000;
          INPUT:POWER:FILTER:LINE:ELEMENT1 OFF;
          ELEMENT2 OFF;ELEMENT3 OFF;ELEMENT4 OFF;
          INPUT:POWER:FILTER:ZCROSS:ELEMENT1 OFF;
          ELEMENT2 OFF;ELEMENT3 OFF;ELEMENT4 OFF;
          INPUT:POWER:SCALING:STATE:ELEMENT1 0;
          ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;
          INPUT:POWER:SCALING:PT:ELEMENT1 1.0000;
          ELEMENT2 1.0000;ELEMENT3 1.0000;
          ELEMENT4 1.0000; INPUT:POWER:SCALING:CT:ELEMENT1 1.0000;
          ELEMENT2 1.0000;ELEMENT3 1.0000;
          ELEMENT4 1.0000; INPUT:POWER:SCALING:SFACTOR:ELEMENT1 1.0000;
          ELEMENT2 1.0000;ELEMENT3 1.0000;
          ELEMENT4 1.0000

[:INPut]:POWer:CURRent?

Function  Queries all settings related to the current measurement on the power measurement module.

Syntax  [:INPut]:POWer:CURRent?

Example  :INPUT:POWER:CURRENT?→:INPUT:POWER:
          CURRENT:TERMINAL:ELEMENT1 5.0E+00;
          ELEMENT2 5.0E+00;ELEMENT3 5.0E+00;
          ELEMENT4 5.0E+00; INPUT:POWER:CURRENT:
          RANGE:ELEMENT1 10.0E+00;
          ELEMENT2 10.0E+00;ELEMENT3 10.0E+00;
          ELEMENT4 10.0E+00; INPUT:POWER:CURRENT:
          SRATIO:ELEMENT1 10.0000;
          ELEMENT2 10.0000;ELEMENT3 10.0000;
          ELEMENT4 10.0000
4.11 INPut Group

[:INPut][:POWer]:CURRent:AUTO?
Function Queries the ON/OFF state of the current auto range function of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:AUTO?
Example :INPUT:POWER:CURRENT:AUTO?→:INPUT:POWER:CURRENT:AUTO:ELEMENT1 0;ELEMENT2 0;ELEMENT3 0;ELEMENT4 0

[:INPut][:POWer]:CURRent:AUTO[:ALL]
Function Turns ON/OFF the current auto range function of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:AUTO[:ALL] {<Boolean>}
Example :INPUT:POWER:CURRENT:AUTO:ALL ON

[:INPut][:POWer]:CURRent:AUTO:ELEMent<x>
Function Turns ON/OFF the current auto range function of each element with power measurement module or queries the current setting.
Syntax [:INPut][:POWer]:CURRent:AUTO:ELEMent<x> {<current>|<voltage>|AUTO}
Example :INPUT:POWER:CURRENT:AUTO:ELEMENT1 10A

[:INPut][:POWer]:CURRent:RANGe?
Function Queries the current range of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:RANGe?
Example :INPUT:POWER:CURRENT:RANGE?

[:INPut][:POWer]:CURRent:RANGe[:ALL]
Function Sets the current range of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:RANGe[:ALL] {<current>|<voltage>|AUTO}
Example :INPUT:POWER:CURRENT:RANGE:ALL 10A

[:INPut][:POWer]:CURRent:RANGe:ELEMent<x>
Function Sets the current range of each element with the power measurement module or queries the current setting.
Syntax [:INPut][:POWer]:CURRent:RANGe:ELEMent<x> {<current>|<voltage>|AUTO}
Example :INPUT:POWER:CURRENT:RANGE:ELEMENT1 10A

Description
- The selectable range is determined by the current input terminal setting of element 1 ([:INPut][:POWer]:CURRent:TERMinal:ELEMent1). Therefore, only elements that have the same current measurement terminal setting as element 1 are set.

[:INPut][:POWer]:CURRent:RANGe:ELEMent<x>
Function Sets the current range of each element with the power measurement module or queries the current setting.
Syntax [:INPut][:POWer]:CURRent:RANGe:ELEMent<x> {<current>|<voltage>|AUTO}
Example :INPUT:POWER:CURRENT:RANGE:ELEMENT1 10A

Description
- The selectable range is determined by the current input terminal setting of the specified element.
- The "-CHANnel<x>:CURRent:RANGe (where <x> is the channel number)" command can be used to make the same settings and inquiries.
- Setting "AUTO" using this command is equivalent to specifying "ON" using the "[:INPut][:POWer]:CURRent:AUTO: ELEMent<x>" command.
[:INPut][:POWer]:CURRent:SRATio?
Function Queries the current sensor's transformation ratio of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:SRATio?
Example :INPUT:POWER:CURRENT:SRATIO?
→ :INPUT:

[:INPut][:POWer]:CURRent:SRATio[:ALL]
Function Sets the current sensor transformation ratio of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:SRATio[:ALL] {<NRf>}
Example :INPUT:POWER:CURRENT:SRATIO:ALL 10

[:INPut][:POWer]:CURRent:SRATio:ELEMent<x>
Function Sets the current sensor transformation ratio of each element with the power measurement module or queries the current setting.
Syntax [:INPut][:POWer]:CURRent:SRATio:ELEMent<x> {<current>|SENSor}
Example :INPUT:POWER:CURRENT:SRATIO:ELEMENT1 10

[:INPut][:POWer]:CURRent:TERMinal?
Function Queries the current input terminals of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:TERMinal?
Example :INPUT:POWER:CURRENT:TERMINAL?
→ :INPUT:POWER:CURRENT:TERMINAL:ELEMENT1 5.0E+00; ELEMENT2 5.0E+00; ELEMENT3 5.0E+00; ELEMENT4 5.0E+00

[:INPut][:POWer]:CURRent:TERMinal[:ALL]
Function Sets the current input terminals of all elements with the power measurement modules.
Syntax [:INPut][:POWer]:CURRent:TERMinal[:ALL] {<current>|SENSor}
Example :INPUT:POWER:CURRENT:TERMINAL:ALL 5A
Description • For elements that have 253751 power measurement modules (1000V/5A) installed, 20(A) setting will not be carried out.
• For elements that do not have 253751/253752 power measurement modules installed, current measurement terminal settings will not be carried out.

[:INPut][:POWer]:CURRent:TERMinal:ELEMent<x>
Function Sets the current input terminals of each element with the power measurement module or queries the current setting.
Syntax [:INPut][:POWer]:CURRent:TERMinal:ELEMent<x> {<current>|SENSor}
Example :INPUT:POWER:CURRENT:TERMINAL:ELEMENT1 5A
Description • If the 253752/253752 power measurement module is not installed, an error will occur.
• The ":CHANnel<x>:CURRent:TERMinal (where <x> is the channel number)" command can be used to make the same settings and inquiries.

[:INPut][:POWer]:FILTer?
Function Queries all settings related to the filter for the power measurement module.
Syntax [:INPut][:POWer]:FILTer?
Example :INPUT:POWER:FILTER?
→ :INPUT:POWER:FILTER:LINE:ELEMENT1 OFF; ELEMENT2 OFF; ELEMENT3 OFF; ELEMENT4 OFF; :INPUT:POWER:FILTER:ZCROSS:ELEMENT1 OFF; ELEMENT2 OFF; ELEMENT3 OFF; ELEMENT4 OFF
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE?</td>
<td>Queries the line filter setting of all elements with the power measurement modules.</td>
<td>[:INPut][:POWer]:FILTER:LINE?</td>
<td>:INPUT:POWER:FILTER:LINE?=:INPUT:POWER:FILTER:LINE:ELEMENT1 OFF;ELEMENT2 OFF;ELEMENT3 OFF;ELEMENT4 OFF</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE[:ALL]</td>
<td>Sets the line filter setting of all elements with the power measurement modules.</td>
<td>[:INPut][:POWer]:FILTER:LINE[:ALL] {OFF</td>
<td>&lt;frequency&gt;}</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:LINE:ELEMent&lt;x&gt;</td>
<td>Sets the line filter setting of each element with the power measurement module or queries the current setting.</td>
<td>[:INPut][:POWer]:FILTER:LINE:ELEMent&lt;x&gt; {OFF</td>
<td>&lt;frequency&gt;}</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:ZCROSS?</td>
<td>Queries the zero crossing filter of all elements with the power measurement modules.</td>
<td>[:INPut][:POWer]:FILTER:ZCROSS?</td>
<td>:INPUT:POWER:FILTER:ZCROSS:ELEMENT1 OFF</td>
</tr>
<tr>
<td>[:INPut][:POWer]:FILTER:ZCROSS[:ALL]</td>
<td>Sets the zero crossing filter of all elements with the power measurement modules.</td>
<td>[:INPut][:POWer]:FILTER:ZCROSS[:ALL] {OFF</td>
<td>&lt;frequency&gt;}</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing?</td>
<td>Queries all settings related to scaling for the power measurement module.</td>
<td>[:INPut][:POWer]:SCALing?</td>
<td>:INPUT:POWER:SCALING:STATE:ELEMENT1 0;ELEMENT2 0;ELEMENT3 0;ELEMENT4 0;:INPUT:POWER:SCALING:PT:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000;:INPUT:POWER:SCALING:CT:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000;:INPUT:POWER:SCALING:SFACTOR:ELEMENT1 1.0000;ELEMENT2 1.0000;ELEMENT3 1.0000;ELEMENT4 1.0000</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:{PT</td>
<td>CT</td>
<td>SFACtor}?</td>
<td>Queries the PT ratio/CT ratio/power coefficient of all elements with the power measurement modules.</td>
</tr>
<tr>
<td>[:INPut][:POWer]:SCALing:{PT</td>
<td>CT</td>
<td>SFACtor}[:ALL]</td>
<td>Sets the PT ratio/CT ratio/power coefficient of all elements with the power measurement modules.</td>
</tr>
</tbody>
</table>
### [:INPut][:POWer]:SCALing[:PT|CT|SFACtor]:ELEMent<x>

**Function**
Sets the PT ratio/CT ratio/power coefficient of each element with the power measurement module or queries the current setting.

**Syntax**
```
[:INPut][:POWer]:SCALing:{PT|CT|SFACtor}:ELEMent<x> {<NRf>}
[:INPut][:POWer]:SCALing:{PT|CT|SFACtor}:ELEMent<x>?
```

**Example**
`:INPUT:POWER:SCALING:PT:ELEMENT1 1
`:INPUT:POWER:SCALING:PT:ELEMENT1?
→ :INPUT:POWER:SCALING:PT:ELEMENT1 1.0000

**[:INPut][:POWer]:SCALing:STATe?**

**Function**
Queries the ON/OFF state of the scaling function of all elements with the power measurement modules.

**Syntax**
```
[:INPut][:POWer]:SCALing:STATe?
```

**Example**
`:INPUT:POWER:SCALING:STATE?
→ :INPUT:POWER:SCALING:STATE:ELEMENT1 0; ELEMENT2 0; ELEMENT3 0; ELEMENT4 0

**[:INPut][:POWer]:SCALing:STATe[:ALL]**

**Function**
Turns ON/OFF the scaling function of all elements with the power measurement modules.

**Syntax**
```
[:INPut][:POWer]:SCALing:STATe[:ALL] {<Boolean>}
```

**Example**
`:INPUT:POWER:SCALING:STATE:ALL OFF

### [:INPut][:POWer]:VOLTage?**

**Function**
Queries all settings related to the voltage measurement for power measurement modules.

**Syntax**
```
[:INPut][:POWer]:VOLTage?
```

**Example**
`:INPUT:POWER:VOLTAGE?
→ :INPUT:POWER:VOLTAGE:RANGE:ELEMENT1 2.00E+03; ELEMENT2 2.00E+03; ELEMENT3 2.00E+03; ELEMENT4 2.00E+03

### [:INPut][:POWer]:VOLTage:STATe?**

**Function**
Queries the ON/OFF state of the voltage auto range function of all elements with the power measurement modules.

**Syntax**
```
[:INPut][:POWer]:VOLTage:AUTO?
```

**Example**
`:INPUT:POWER:VOLTAGE:AUTO?
→ :INPUT:POWER:VOLTAGE:AUTO:ELEMENT1 0; ELEMENT2 0; ELEMENT3 0; ELEMENT4 0

### [:INPut][:POWer]:VOLTage:AUTO[:ALL]**

**Function**
Turns ON/OFF the voltage auto range function of all elements with the power measurement modules.

**Syntax**
```
[:INPut][:POWer]:VOLTage:AUTO[:ALL] {<Boolean>}
```

**Example**
`:INPUT:POWER:VOLTAGE:AUTO:ALL ON

### [:INPut][:POWer]:VOLTage:AUTO:ELEMent<x>**

**Function**
Turns ON/OFF the voltage auto range function of each element with the power measurement module or queries the current setting.

**Syntax**
```
[:INPut][:POWer]:VOLTage:AUTO:ELEMent<x> {<Boolean>}
[:INPut][:POWer]:VOLTage:AUTO:ELEMent<x>?
```

**Example**
`:INPUT:POWER:VOLTAGE:AUTO:ELEMENT1 ON
`:INPUT:POWER:VOLTAGE:AUTO:ELEMENT1?
→ :INPUT:POWER:VOLTAGE:AUTO:ELEMENT1 1

### [:INPut][:POWer]:VOLTage:RANGe**

**Function**
Sets the voltage range of all elements with the power measurement modules.

**Syntax**
```
[:INPut][:POWer]:VOLTage:RANGe[:ALL] {<voltage>|AUTO}
```

**Example**

**Description**
Setting "AUTO" using this command is equivalent to executing 
```
[:INPut][:POWer]:VOLTage:AUTO[:ALL] ON.
```

---

IM 253710-11E

4-51

4 Commands
4.11 INPut Group

[:INPut][:POWer]:VOLTage:RANGe:
ELEMent<x>

Function  Sets the voltage range of each element with the power measurement module or queries the current setting.

Syntax  [:INPut][:POWer]:VOLTage:RANGe:
ELEMent<x> {<voltage>|AUTO}
[:INPut][:POWer]:VOLTage:RANGe:
ELEMent<x>?
<x> = 1 to 4
<voltage> = 30, 60, 120, 200, 300, 600, 1200, 2000(V)
AUTO = AUTO RANGE

Example  :INPUT:POWER:VOLTAGE:RANGE:
ELEMent1 2000V
:INPUT:POWER:VOLTAGE:RANGE:ELEMent1?
INPUT:POWER:VOLTAGE:RANGE:
ELEMent1 2.00E+03

Description  • The "[:CHANnel<x>:VOLTage:RANGe (where <x> is the channel number)" command can be used to make the same setting and inquiries.
• Setting "AUTO" using this command is equivalent to specifying "ON" using the "[:INPut][:POWer]:VOLTage:AUTO: ELEMent<x>" command.
4.12 MATH Group

The commands in the MATH Group deal with computations. These commands can be used to make the same settings and inquiries as when the MATH key on the front panel is pressed.

:Math<x>?
Function Queries all settings related to computations.
Syntax :Math<x>?
<x> = 1, 2
Example :Math1?:Math:Mode 1;Function 0;
Expression "C1";Unit "";Scaling:
Mode AUTO;Value 1.0000E+02,
-1.0000E+02.;:Math1:Point 10.0000E-03,
90.0000E-03;FFT:Point 1000;
Window Rectangle

:Math<x>:Execute
Function Executes computation.
Syntax :Math<x>:Execute
<x> = 1, 2
Example :Math1:Execute
Description This command is applicable to both MATH1 and MATH2. Specifying <x> has no meaning.
4.12 MATH Group

**:MATH<x>:EXPRession**

Function:
Sets the equation or queries the current setting.

Syntax:
```plaintext
:MATH<x>:EXPRession {<string>}
```

- `<x>` = 1, 2
- `<string>` = 50 characters or less

Example:
```plaintext
:MATH1:EXPRession "C1"
```

Description:
Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.

**:MATH<x>:FFT?**

Function:
Queries all settings related to the FFT.

Syntax:
```plaintext
:MATH<x>:FFT?
```

- `<x>` = 1, 2

Example:
```plaintext
:MATH1:FFT? -> :MATH1:FFT:POINT 1000; WINDOW RECTANGLE
```

Description:
This command is applicable to both MATH1 and MATH2. Specifying `<x>` has no meaning.

**:MATH<x>:FFT:POINT**

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

Syntax:
```plaintext
:MATH<x>:FFT:POINT {<NRf>}
```

- `<x>` = 1, 2
- `<NRf>` = 1000, 2000, 10000

Example:
```plaintext
:MATH1:FFT:POINT 1000
```

Description:
This command is applicable to both MATH1 and MATH2. Specifying `<x>` has no meaning.

**:MATH<x>:FFT:WINDOW**

Function:
Sets the window function for the FFT or queries the current setting.

Syntax:
```plaintext
:MATH<x>:FFT:WINDOW {RECTangle|HANNing}
```

- `<x>` = 1, 2

Example:
```plaintext
:MATH1:FFT:WINDOW RECTANGLE
```

Description:
This command is applicable to both MATH1 and MATH2. Specifying `<x>` has no meaning.

**:MATH<x>:SCALEing?**

Function:
Queries all settings related to converting the scale.

Syntax:
```plaintext
:MATH<x>:SCALEing?
```

- `<x>` = 1, 2

Example:
```plaintext
:MATH1:SCALEing? -> :MATH1:SCALEing: MODE AUTO; VALUE 0.1000, 0.0000
```
:MATH<x>:SCALing:MODE
Function: Sets the converting the scale or queries the current setting.
Syntax: :MATH<x>:SCALing:MODE {AUTO|MANual}
         :MATH<x>:SCALing:MODE?
         <x> = 1, 2
Example:
: MATH1:SCALING:MODE AUTO
: MATH1:SCALING:MODE?
→ :MATH1:SCALING:MODE AUTO

:MATH<x>:SCALing:VALue
Function: Sets the upper and lower limits for manual scaling or queries the current setting.
Syntax: :MATH<x>:SCALing:VALue {<NRf>,<NRf>}
         :MATH<x>:SCALing:VALue?
         <x> = 1, 2
         <NRf> = -9.9999E+30 to 9.9999E+30
Example:
: MATH1:SCALING:VALUE 100,-100
: MATH1:SCALING:VALUE?
→ :MATH1:SCALING:VALUE 1.0000E+02,-1.0000E+02
Description: Set the upper limit, then the lower limit.

:MATH<x>:UNIT
Function: Sets the unit to attach to the computed result or queries the current setting.
Syntax: :MATH<x>:UNIT {<string>}
         :MATH<x>:UNIT?
         <x> = 1, 2
         <string> = 8 characters or less
Example: : MATH1:UNIT ""
: MATH1:UNIT?
→ :MATH1:UNIT ""
Description:
• Characters and symbols other than the ones displayed on the keyboard on the screen cannot be used.
• This command does not affect the computation results in any way.
4.13 MEASure Group

The commands in the MEASure Group deal with measurements. These commands can be used to make the same settings and inquiries as when the MEASURE key on the front panel is pressed.
### 4.13 MEASure Group

**:MEASure?**

**Function**: Queries all settings related to measurements.

**Syntax**: :MEASURE?

**Example**: :MEASURE?

→ :MEASURE:MODE 1;PERIOD:
MODE ZCROSS;ZCROSS:SYNCHRONIZE:
ELEMENT1 2;ELEMENT2 4;ELEMENT3 6;
ELEMENT4 8; :MEASURE:DMEASURE OFF;
FUNCTION1:STATE 0;EXPRESSION "URMS(E1)"
UNIT ";"; :MEASURE:FUNCTION2:STATE 0;
EXPRESSION "URMS(E2)";UNIT ";"; :MEASURE:
FUNCTION3:STATE 0;EXPRESSION "URMS(E3)"
UNIT ";"; :MEASURE:FUNCTION4:STATE 0;
EXPRESSION "URMS(E4)";UNIT ";"; :MEASURE:
SFORMULA RMS;AVERAGING:STATE 1;COUNT 4;
:MEASURE:PHASE 180;PC:IEC 1976;P1 0.5000;
P2 0.5000

---

**:MEASure:AVERaging?**

**Function**: Queries all settings related to averaging.

**Syntax**: :MEASURE:AVERaging?

**Example**: :MEASURE:AVERaging?

→ :MEASURE:AVERaging:
STATE 1;COUNT 4

---

**:MEASure:AVERaging:COUNt**

**Function**: Sets the number of averaging counts or queries the current setting.

**Syntax**: :MEASURE:AVERaging:COUNt {<NRf>}

:MEASURE:AVERaging:COUNt?

Example

:MEASURE:AVERaging:COUNt 4

:MEASURE:AVERaging:COUNt?

→ :MEASURE:AVERaging:COUNt 4

---

**:MEASure:AVERaging[:STATe]**

**Function**: Turns ON/OFF the averaging function 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
4.13 MEASURE Group

:MEASURE:DMeasure
Function Sets the delta computation or queries the current setting.
Syntax :MEASURE:DMeasure {OFF|U1_U2|I1_I2|P3V3|P3W3|V3A3|DT_ST|ST_DT}
Example :MEASURE:DMEASURE OFF
:MEASURE:DMeasure?
Description The following selection are available.
OFF = Does not perform delta computation.
U1_U2 = u1-u2
I1_I2 = i1-i2
P3V3 = 3P3V-to-3V transformation
P3W3 = 3P3W-to-3V transformation
V3A3 = 3V3A transformation
DT_ST = Delta-to-Star transformation
ST_DT = Star to Delta transformation

:MEASURE:FUNCTION<x>?
Function Queries all settings related to the user-defined function.
Syntax :MEASURE:FUNCTION<x>?
<x> = 1 to 4
Example :MEASURE:FUNCTION1?

:MEASURE:FUNCTION<x>:EXPRession
Function Sets the equation for the user-defined function or queries the current setting.
Syntax :MEASURE:FUNCTION<x>:EXPRession {<string>}
Example :MEASURE:FUNCTION1:EXPRESSION "URMS(E1)"

:MEASURE:FUNCTION<x>[:STATE]
Function Enable/disable the user-defined function or queries the current setting.
Syntax :MEASURE:FUNCTION<x>[:STATE] {<Boolean>}
Example :MEASURE:FUNCTION1:STATE ON

:MEASURE:FUNCTION<x>:UNIT
Function Sets the unit to attach to the computed result of the user-defined function or queries the current setting.
Syntax :MEASURE:FUNCTION<x>:UNIT {<string>}
Example :MEASURE:FUNCTION1:UNIT ""
4.13 MEASure Group

**:MEASURE:**PC?

Function Queries all settings related to determination 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 determine the Pc (Corrected Power) or queries the current setting.

Syntax :MEASURE:**PC:**IEC**{<NRf>}**

Example :MEASURE:**PC:**IEC 1976

**:MEASURE:**PC:**IEC**? <NRf> = 1976, 1993

Example :MEASURE:**PC:**IEC**?** → :MEASURE:**PC:**IEC 1976

Description Specifies the year of the issue of the IEC76-1 in which the equation used to determine the Pc is given.

**:MEASURE:**PC:**P<x>**

Function Pc(Corrected Power) Sets the parameters used to determine the Pc (Corrected Power) or queries the current setting.

Syntax :MEASURE:**PC:**P<x>{<NRf>}**

Example :MEASURE:**PC:**P1 0.5

**:MEASURE:**PC:**P<x>**? <x> = 1, 2

Example :MEASURE:**PC:**P1? → :MEASURE:**PC:**P1 0.5000

Description This parameter is used when ":**MEASURE:**PC:**IEC" is set to "1976(IEC76-1(1976), IEEE C57.12.90-1993)."

**:MEASURE:**PERiod?

Function Queries all settings related to the computation period.

Syntax :MEASURE:**PERiod?**

Example :MEASURE:**PERiod? → :MEASURE:**PERiod:**MODE **ZCROSS;ZCROSS:**SYNCHRONIZE: ELEMENT1 2; ELEMENT2 4; ELEMENT3 6; ELEMENT4 8

**:MEASURE:**PERiod:**ETRigger?**

Function Queries all settings when using the external trigger signal to determine the computation period.

Syntax :MEASURE:**PERiod:**ETRigger?**

Example :MEASURE:**PERiod:**ETRigger? → :MEASURE:**PERiod:**ETRIGGER:**PATTERN **LOW

**:MEASURE:**PERiod:**ETRigger{[:PATTern]}**

Function Sets the pattern that is used when determining the computation period with the external trigger signal or queries the current setting.

Syntax :MEASURE:**PERiod:**ETRigger{[:PATTern]}**{LOW|HIGH}

Example :MEASURE:**PERiod:**ETRigger{[:PATTern]}{LOW|HIGH} → :MEASURE:**PERiod:**ETRigger{[:PATTern]}{LOW|HIGH}
### 4.13 MEASure Group

#### :MEASure:PERiod:EXECute

**Function**
Executes the computation.

**Syntax**
:MEASure:PERiod:EXECute

**Example**
:MEASURE:PERIOD:EXECUTE

#### :MEASure:PERiod[:MODE]

**Function**
Sets the method used to specify the computation period or queries the current setting.

**Syntax**
:MEASure:PERiod[:MODE] {ZCRoss|CURSor|ETRigger}

**Example**
:MEASURE:PERIOD:MODE?

**Example**
:MEASURE:PERIOD:MODE ZCROSS

**Description**
This command is valid during the normal measurement mode. It is fixed to CURSor during the harmonic measurement mode.

#### :MEASure:PERiod:ZCRoss?

**Function**
Queries all settings when using the zero crossing detection to determine the computation period.

**Syntax**
:MEASure:PERiod:ZCRoss?

**Example**
:MEASURE:PERIOD:ZCROSS?

**Example**
:MEASURE:PERIOD:ZCROSS:SYNCHRONIZE:ELEMENT1 2; ELEMENT2 4; ELEMENT3 6; ELEMENT4 8

#### :MEASure:PERiod:ZCRoss:SYNChronize?

**Function**
Sets the synchronizing source for all elements when using the zero crossing detection to determine the computation period.

**Syntax**
:MEASure:PERiod:ZCRoss:SYNChronize?

**Example**
:MEASURE:PERIOD:ZCROSS:SYNCHRONIZE?

**Example**
:MEASURE:PERIOD:ZCROSS:SYNCHRONIZE:ELEMENT1 2; ELEMENT2 4; ELEMENT3 6; ELEMENT4 8

#### :MEASure:PERiod:ZCRoss[:SYNChronize]:ELEMent<x>

**Function**
Sets the synchronizing source for each element when using the zero crossing detection to determine the computation period.

**Syntax**
:MEASure:PERiod:ZCRoss[:SYNChronize]:ELEMent<x> {<NRf>|EXTernal}

**Example**
:MEASURE:PERIOD:ZCROSS:SYNCHRONIZE:ELEMENT1 2

**Example**

#### :MEASure:PHASe

**Function**
Sets the display format of the phase difference or queries the current setting.

**Syntax**
:MEASure:PHASe {<NRf>}

**Example**
:MEASURE:PHASE 180

**Example**

**Description**
“180” and “360” denote 0 to ±180° (Lead/Lag) and 0 to 360°, respectively.

#### :MEASure:SFORmula

**Function**
Sets the equation used to determine S (apparent power) or queries the current setting.

**Syntax**
:MEASure:SFORmula {RMS|MEAN|DC}

**Example**
:MEASURE:SFORMULA RMS

**Example**
:MEASURE:SFORMULA? → :MEASURE:SFORMULA RMS

**Description**
The equation corresponding to each selection is as follows:
RMS : S = Urms * Irms
MEAN : S = Umean * Imean
DC : S = Udc * Idc
4.14 NULL Group

The commands in the NULL Group deal with the NULL function. These commands can be used to make the same settings and inquiries as when the NULL key on the front panel is pressed.

:NULL
Function               Turns ON/OFF the NULL function or queries the current setting.
Syntax                 :NULL {<Boolean>}
Example                :NULL ON
                        :NULL? → :NULL 1
Description            When turn ON, the applied voltage/current is set as the reference (0) and all succeeding measured values will be based on this reference.
The commands in the NUMeric Group deal with the output of numerical data. There are no front-panel keys that correspond to the commands in this group.
4.15 NUMeric Group

**:NUMeric?**

**Function**
Queries all settings related to the numerical data output.

**Syntax**
```
:NUMeric?
```

**Example**
```
:NUMERIC?
```

**:NUMeric:FORMat**

**Function**
Sets the format of the numerical data that are sent using the ":NUMeric:(NORMAL|HARMonics|LIST):VALue?" command or queries the current setting.

**Syntax**
```
:NUMeric:FORMat {ASCii|FLOat}
```

**Example**
```
:NUMERIC:FORMAT ASCII
```

**Description**
The format of the numerical data that is output depends on the ":NUMeric:FORMat" setting.
1. When set to "ASCii"
   - The physical values are output in <NR3> format. Each item of data is separated by a comma.
2. When set to "FLOat"
   - A 6-byte header ("#40060" for example) is added to beginning of numeric data block.
   - The header is followed by physical values in IEEE single precision floating point format (4 bytes).
   - The byte order of each item of data is MSB First.

**:NUMeric:HARMonics?**

**Function**
Queries all settings related to the numerical data output during harmonic measurement.

**Syntax**
```
:NUMeric:HARMonics?
```

**Example**
```
:NUMERIC:HARMONICS?
```

**:NUMeric:HARMonics:CLEar**

**Function**
Clears the numerical data output items during harmonic measurement (sets them to NONE).

**Syntax**
```
:NUMeric:HARMonics:CLEar {ALL|[<NRf>,<NRf>]}<NRf> = 1 to 255 (First item number to clear)
```

**Example**
```
:NUMERIC:HARMONICS:CLEAR ALL
```

**Description**
If the second <NRf> is omitted, output items from the first item number to the end item (255) are cleared.

**:NUMeric:HARMonics:ITEM<x>**

**Function**
Sets the numerical data output items during harmonic measurement or queries the current setting.

**Syntax**
```
:NUMeric:HARMonics:ITEM<x> {NONE|[<Function>,<Element>,<Order>]}<x> = 1 to 255 (item number)
```

**Example**
```
:NUMERIC:HARMONICS:ITEM1 U,1,1
```

**:NUMeric:HARMonics:NUMber**

**Function**
Sets the number of numerical data that are sent using the ":NUMeric:HARMonics:VALue?" command or queries the current setting.

**Syntax**
```
:NUMeric:HARMonics:NUMber {<NRf>|ALL}<NRf> = 1 to 255 (ALL)
```

**Example**
```
:NUMERIC:HARMONICS:NUMBER 8
```

**Description**
If the parameter is omitted in the ":NUMeric:HARMonics:VALue?" command, 1 to (the specified value) of numerical data are output in order.
**:NUMeric:HARMonics:PRESet**

**Function**
Sets the numerical data output items to a preset pattern during harmonic measurement.

**Syntax**
`:NUMeric:HARMonics:PRESet {<NRf>}
<NRf> = 1 to 4`

**Example**
`:NUMERIC:HARMONICS:PRESET 1`

**Description**
For information related to the output items that are set to preset values, see "A list of numerical data output items that are preset" on page 4-67.

**:NUMeric:HARMonics:VALue?**

**Function**
Queries the numerical data during harmonic measurement.

**Syntax**
`:NUMeric:HARMonics:VALue? {<NRf>}
<NRf> = 1 to 255 (item number)
Example when <NRf> is specified
`:NUMERIC:HARMONICS:VALUE? 1` → `104.75E+00`
Example when <NRf> is omitted
`:NUMERIC:HARMONICS:VALUE?` → `104.75E+00, 0.9584E+00, 72.01E+00, (omit), 50.086E+00`

**Description**
- When <NRf> is specified, only the numerical data of that item number are output.
- When <NRf> is omitted, the numerical data from 1 to the item number specified using the ":NUMeric:HARMonics:NUMber" command are output in order.
- If the item of the specified number is set to "NONE" or if no numerical data exist, the item will output error data. "NAN" (Not A Number) is returned when ":NUMeric:FORMat" is set to "ASCii." 9.91E+37 is returned if it is set to "FLOat."
- In addition, if the numerical data are erroneous (the display is "Error" or "--OF--"), "INF" (infinity) is returned when ":NUMeric:FORMat" is set to "ASCii." 9.9E+37 is returned if it is set to "FLOat."

**:NUMeric:LIST?**

**Function**
Queries all settings related to the output of the numerical list data during harmonic measurement.

**Syntax**
`:NUMeric:LIST?`

**Example**
`:NUMERIC:LIST?` → `ORDER 100; SELECT ALL; ITEM U,1`

**:NUMeric:LIST:ITEM**

**Function**
Sets the output items of the numerical list data during harmonic measurement or queries the current setting.

**Syntax**
`:NUMeric:LIST:ITEM {<Function>, <Element>}
<Function> = {U|I|P|S|Q|LAMBda|PHI|PHIU|PHII} (See the function selection list on page 4-32 (3))
<Element> = {<NRf>|SIGMA|SIGMB} (<NRf>=1 to 4)

**Example**
`:NUMERIC:LIST:ITEM U,1
`:NUMERIC:LIST:ITEM?` → `U,1`

**:NUMeric:LIST:ORDer**

**Function**
Sets the maximum harmonic order of the numerical list data to output during harmonic measurement or queries the current setting.

**Syntax**
`:NUMeric:LIST:ORDer {<NRf>|ALL}
Example when <NRf> is specified
`:NUMERIC:LIST:ORDER 100`
`:NUMERIC:LIST:ORDER?` → `ORDER 100`

**:NUMeric:LIST:SElect**

**Function**
Sets the output components of the numerical list data during harmonic measurement or queries the current setting.

**Syntax**
`:NUMeric:LIST:SElect {EVEN|ODD|ALL}
Example when <NRf> is specified
`:NUMERIC:LIST:SELECT ALL`
`:NUMERIC:LIST:SELECT?` → `SELECT ALL`

**:NUMeric:LIST:VALue?**

**Function**
Queries the numerical list data during harmonic measurement.

**Syntax**
`:NUMeric:LIST:VALue?`

**Example**
`:NUMERIC:LIST:VALUE?` → `103.58E+00, 0.00E+00, 103.53E+00, 0.09E+00, 2.07E+00, 0.04E+00, (omit), 0.01E+00, 0.01E+00`

**Description**
The numerical data of TOTal, DC, and 1st order to ":NUMeric:LIST:ORDer" are output.

**:NUMeric:NORMal?**

**Function**
Queries all settings related to the numerical data output during normal measurement.

**Syntax**
`:NUMeric:NORMal?`

**Example**
`:NUMERIC:NORMAL?` → `NUMBER 8; ITEM1 URMS,1; ITEM2 UMN,1; ITEM3 UDC,1; ITEM4 UAC,1; ITEM5 IMN,1; ITEM6 IMD,1; ITEM7 IDC,1; ITEM8 IAC,1`
4.15 NUMerical Group

**:NUMeric[:NORMal]:CLEar**

**Function**
Clears the numerical data output items during normal measurement (Sets them to "NONE").

**Syntax**
```
:NUMeric[:NORMal]:CLEar {ALL | <NRf>[, <NRf>]}<br>
```
- **First <NRf>** = 1 to 255  
  (First item number to clear)
- **Second <NRf>** = 1 to 255  
  (Last item number to clear)

**Example**
```
:NUMERIC:NORMAL:CLEAR ALL
```

**Description**
If the second <NRf> is omitted, output items from the first item number to the end item (255) are cleared.

**:NUMeric[:NORMal]:ITEM<x>**

**Function**
Sets the numerical data output items during normal measurement or queries the current setting.

**Syntax**
```
:NUMeric[:NORMal]:ITEM<x> {NONE | <Function>,<Element>}
:NUMeric[:NORMal]:ITEM<x>?
```
- **<x>** = 1 to 255 (item number)
- **NONE** = No output items
- **<Function>** = {URMS|UMN|UDC|UAC|IRMS|...}  
  (See the function selection list on page 4-31 (1))
- **<Element>** = {<NRf>|SIGMA|SIGMB}(<NRf> = 1 to 4)

**Example**
```
:NUMERIC:NORMAL:ITEM1 URMS,1
:NUMERIC:NORMAL:ITEM1?
```

**:NUMeric[:NORMal]:NUMber**

**Function**
Sets the number of numerical data that are sent using the ":NUMeric:NORMal:VALue?” command or queries the current setting.

**Syntax**
```
:NUMeric[:NORMal]:NUMber {<NRf>|ALL}
:NUMeric[:NORMal]:NUMber?
```
- **<NRf>** = 1 to 255  
  (ALL)

**Example**
```
:NUMERIC:NORMAL:NUMBER 8
:NUMERIC:NORMAL:NUMBER?
```

**Description**
If the parameter is omitted in the ":NUMeric:HARMonics:VALue?” command, 1 to (the specified value) of numerical data are output in order.

**:NUMeric[:NORMal]:PRESet**

**Function**
Sets the numerical data output items to a preset pattern during normal measurement.

**Syntax**
```
:NUMeric[:NORMal]:PRESet {<NRf>}
```
- **<NRf>** = 1 to 4

**Example**
```
:NUMERIC:NORMAL:PRESET 1
```

**Description**
For information related to the output items that are set to preset values, see "A list of numerical data output items that are preset" on next page.

**:NUMeric[:NORMal]:VALue?**

**Function**
Queries the numerical data during normal measurement.

**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, (omit), 49.868E+00
```

**Description**
- When <NRf> is specified, only the numerical data of that item number are output.
- When <NRf> is omitted, the numerical data from 1 to the item number specified using the ":NUMeric:HARMonics:NUMber” command are output in order.
- If the item of the specified number is set to "NONE" or if no numerical data exist, the item will output error data. "NAN" (Not A Number) is returned when ":NUMeric:FORMat” is set to "ASCii.”  
  9.91E+37 is returned if it is set to "FLOat.”
- If the output item is PHI (φ), the result is returned in the range from 0 to 360° regardless of the display format of the phase difference specified by MEASure:PHASe.
### 4.15 NUMeric Group

<table>
<thead>
<tr>
<th>Pattern 1</th>
<th>ITEM&lt;x&gt;</th>
<th>Function</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>URMS,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IRMS,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>P,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>S,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Q,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>LAMBda,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PHI,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FU,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>FI,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 to 19</td>
<td>URMS to FI,</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 to 29</td>
<td>URMS to FI,</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 to 39</td>
<td>URMS to FI,</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41 to 49</td>
<td>URMS to FI,</td>
<td>SIGMA</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51 to 59</td>
<td>URMS to FI,</td>
<td>SIGMB</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61 to 255</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern 2</th>
<th>ITEM&lt;x&gt;</th>
<th>Function</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>URMS,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>UMN,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>UDC,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>UAC,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IRMS,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>IMN,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IDC,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>IAC,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>P,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>S,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Q,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>LAMBda,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PHI,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>FU,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>FI,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>16 to 30</td>
<td>URMS to FI,</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>31 to 45</td>
<td>URMS to FI,</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>46 to 60</td>
<td>URMS to FI,</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>61 to 75</td>
<td>URMS to FI,</td>
<td>SIGMA</td>
<td></td>
</tr>
<tr>
<td>76 to 90</td>
<td>URMS to FI,</td>
<td>SIGMB</td>
<td></td>
</tr>
<tr>
<td>91 to 255</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern 3</th>
<th>ITEM&lt;x&gt;</th>
<th>Function</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>URMS,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>UMN,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>UDC,</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Pattern 4

<table>
<thead>
<tr>
<th>ITEM&lt;x&gt;</th>
<th>Function</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>URMS,</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>UMN,</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>UDC,</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>UAC,</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>IRMS,</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>IMN,</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>IDC,</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>IAC,</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>P,</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>S,</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Q,</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>LAMBda,</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>PHI,</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>FU,</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>FI,</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>URMS to FI,</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>URMS to FI,</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>URMS to FI,</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>URMS to FI,</td>
<td>SIGMA</td>
</tr>
<tr>
<td>20</td>
<td>URMS to FI,</td>
<td>SIGMB</td>
</tr>
<tr>
<td>21</td>
<td>CFU,</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>CFI,</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>FFU,</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>FFI,</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>RS,</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>XS,</td>
<td>1</td>
</tr>
</tbody>
</table>

**A list of numerical data output items that are preset**

(1) Preset pattern of normal measurement numerical data output items

Applicable command "NUMer:PRESet"

**Pattern 1**

**Pattern 2**

**Pattern 3**

**Pattern 4**
4.15 NUMeric Group

Commands

- 4-67IM 253710-11E

4

4.15 NUMeric Group

27 RP, 1
28 XP, 1
29 PC, 1
30 ETA, 1
31 to 60 URMS to ETA, 2
61 to 90 URMS to ETA, 3
91 to 120 URMS to ETA, 4
121 to 150 URMS to ETA, SIGMA
151 to 180 URMS to ETA, SIGMB
181 to 255 NONE

(2) Preset pattern of harmonic measurement numerical data output items

Applicable command " :NUM: :HARMonics:PRESet"

181 to 255 NONE

Pattern 1

ITEM<x> <Function>, <Element>, <Order>
1 U, 1, TOTal
2 I, 1, TOTal
3 P, 1, TOTal
4 Q, 1, TOTal
5 U, 1, 1
6 I, 1, 1
7 P, 1, 1
8 Q, 1, 1
9 FU, 1, (1)
10 FI, 1, (1)
11 to 20 U to FI, 2, TOTal to 1
21 to 30 U to FI, 3, TOTal to 1
31 to 40 U to FI, 4, TOTal to 1
41 to 50 U to FI, SIGMA, TOTal to 1
51 to 60 U to FI, SIGMB, TOTal to 1
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 U, 1, DC(0)
8 I, 1, DC(0)
9 P, 1, DC(0)
10 S, 1, DC(0)
11 Q, 1, DC(0)
12 U, 1, 1
13 I, 1, 1
14 P, 1, 1
15 S, 1, 1
16 Q, 1, 1
17 LAMBda, 1, 1
18 PHI, 1, 1
19 FU, 1, (1)
20 FI, 1, (1)
21 to 40 U to FI, 2, TOTal to 1
41 to 60 U to FI, 3, TOTal to 1
61 to 80 U to FI, 4, TOTal to 1
81 to 100 U to FI, SIGMA, TOTal to 1
101 to 120 U to FI, SIGMB, TOTal to 1
121 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 LAMBda, 1, TOTal
7 U, 1, DC(0)
8 I, 1, DC(0)
9 P, 1, DC(0)
10 S, 1, DC(0)
11 Q, 1, DC(0)
12 U, 1, 1
13 I, 1, 1
14 P, 1, 1
15 S, 1, 1
16 Q, 1, 1
17 LAMBda, 1, 1
18 PHI, 1, 1
19 FU, 1, (1)
20 FI, 1, (1)
21 UTHD, 1, (1)

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 U, 1, DC(0)
8 I, 1, DC(0)
9 P, 1, DC(0)
10 S, 1, DC(0)
11 Q, 1, DC(0)
12 U, 1, 1
13 I, 1, 1
14 P, 1, 1
15 S, 1, 1
16 Q, 1, 1
17 LAMBda, 1, 1
18 PHI, 1, 1
19 FU, 1, (1)
20 FI, 1, (1)
21 Z, 1, 1
22 RS, 1, 1
23 XS, 1, 1
24 RP, 1, 1
25 XP, 1, 1
26 UTHD, 1, (1)
### 4.15 NUMeric Group

<table>
<thead>
<tr>
<th>Range</th>
<th>Description</th>
<th>Value</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>ITHD</td>
<td>1</td>
<td>(1)</td>
</tr>
<tr>
<td>28</td>
<td>PTHD</td>
<td>1</td>
<td>(1)</td>
</tr>
<tr>
<td>29</td>
<td>STHD</td>
<td>1</td>
<td>(1)</td>
</tr>
<tr>
<td>30</td>
<td>QTHD</td>
<td>1</td>
<td>(1)</td>
</tr>
<tr>
<td>31 to 60</td>
<td>U to QTHD</td>
<td>2</td>
<td>TOTal to 1</td>
</tr>
<tr>
<td>61 to 90</td>
<td>U to QTHD</td>
<td>3</td>
<td>TOTal to 1</td>
</tr>
<tr>
<td>91 to 120</td>
<td>U to QTHD</td>
<td>4</td>
<td>TOTal to 1</td>
</tr>
<tr>
<td>121 to 150</td>
<td>U to QTHD</td>
<td>SIGMA</td>
<td>TOTal to 1</td>
</tr>
<tr>
<td>151 to 180</td>
<td>U to QTHD</td>
<td>SIGMB</td>
<td>TOTal to 1</td>
</tr>
<tr>
<td>181 to 255</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.16 SETup Group

The commands in the SETup Group deal with setting the measurement mode. These commands can be used to make the same settings and inquiries as when the SETUP key on the front panel is pressed.

:SETup?
Function Queries all settings related to the measurement mode.
Syntax :SETup?
Example :SETUP?→:SETUP:MODE NORMAL; WIRING P1W2,P1W2;RESOLUTION 5

:SETup:INITialize
Function Initializes the settings.
Syntax :SETup:INITialize
Example :SETUP:INITIALIZE
Description Resets all setup parameters except communication settings to factory default values.

:SETup[:MODE]
Function Sets the measurement mode or queries the current setting.
Syntax :SETup[:MODE] {NORMal|HARMonics}
Example :SETUP:MODE NORMAL

:SETup:PLLSource
Function Sets the PLL source during harmonic measurement or queries the current setting.
Syntax :SETup:PLLSource {<NRF>|EXTernal}
Example :SETUP:PLLSOURCE 1

:SETup:RESolution
Function Sets the number of displayed digits for numerical data or queries the current setting.
Syntax :SETup:RESolution {<NRF>}<NRf> = 5, 6
Example :SETUP:RESOLUTION 5

4.16 SetUp Group/4.17 SSTart Group/4.18 STARt Group

:SETup:WIRing

Function: Sets the wiring method or queries the current setting.

Syntax: :SETup:WIRing {(P1W2|P1W3|P3W3|P3W4| V3A3|NONE)}

:SETup:WIRing?

P1W2 = single-phase two-wire system
P1W3 = single-phase three-wire system
P3W3 = three-phase three-wire system
P3W4 = three-phase four-wire system
V3A3 = three-voltage three-current system
NONE = No wiring

Example: :SETUP:WIRING P1W2,P1W2

Description:
- Set Wiring-A, then Wiring-B.
- Wiring-B can be omitted for combinations in which Wiring-B cannot be set.
- Depending on the model, some combinations of wiring methods cannot be selected.
- For a single-phase model, Wiring-A is fixed to P1W2 and Wiring-B cannot be set.

4.17 SSTart Group

The commands in the SSTart Group are used to execute single start measurement. This command can be used to execute the same operation as when the SINGLE START key on the front panel is pressed.

:SSTart

Function: Executes single start.

Syntax: :SSTart

Example: :SSTART

4.18 STARt Group

The commands in the STARt Group are used to start the data acquisition operation. This command can be used to execute the same operation as when the START/STOP key on the front panel is pressed.

:STARt

Function: Starts data acquisition.

Syntax: :STARt

Example: :START

Description: Use the ":STOP" command to stop the data acquisition.
4.19 STATus Group

The commands in the STATus Group are used to set and query the status report. There are no front-panel keys that correspond to the commands in this group. For the status report, see chapter 5.

: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 0;QMESSAGE 1

: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 #B00000000
:STATUS:EESE?=→:STATUS:EESE 0

:STATus:EESR? (Extended Event Status Register)
Function Queries and clears the extended event register.
Syntax :STATus:EESR?
Example :STATUS:EESR?→0

:STATus:CONDition?
Function Queries the status register.
Syntax :STATus:CONDition?
Example :STATUS:CONDition?=→16
Description For the description regarding how to synchronize the program using the :STATus:CONDition command, see page 3-8.
4.19 STATus Group

:STATus:ERRor?
Function Queries the code and information of the error (top of the error queue).
Syntax :STATus:ERRor?
Example :STATus:ERROR?
Description • "0" (No error) is returned, if there is no error.
• The messages cannot be returned in Japanese.
• You can set whether or not to attach the messages to the error 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 Sets how the bits in the status register must change in order to set the event. If it is set to "Rise", an event is set when the value changes from "0" to "1."

:STATus:QENable
Function Sets whether or not to store messages other than errors in the error queue or queries the current setting.
Syntax :STATus:QENable {<Boolean>}
:STATus:QENable?
Example :STATus:QENABLE ON
:STATus:QENABLE?→:STATus:QENABLE 1

:STATus:QMESSage
Function Sets whether or not to attach a message to the "STATus:ERRor?" response or queries the current setting.
Syntax :STATus:QMESSage {<Boolean>}
:STATus:QMESSage?
Example :STATus:QMESSAGE ON
:STATus:QMESSAGE?→:STATus:QMESSAGE 1

:STATus:SPOLI?
Function Executes serial polling.
Syntax :STATus:SPOLI?
Example :STATus:SPOLI?
:STATus:SPOLI?→:STATus:SPOLI 0
Description This is a dedicated command for the serial interface. An interface message is available for the GP-IB interface.

4.20 STOP Group
The commands in the STOP Group are used to stop the data acquisition operation. This command can be used to execute the same operation as when the START/STOP key on the front panel is pressed.

:STOP
Function Stops data acquisition.
Syntax :STOP
Example :STOP
Description Use the ":START" command to start the data acquisition.

STOP
4.21 SYSTem Group

The commands in the SYSTem Group deal with cursor measurements. These commands can be used to make the same settings and inquiries as when the MISC key on the front panel is pressed.
4.21 SYSTem Group

:SYSTem?
Function Queries all settings related to the system.
Syntax :SYSTem?
Example :SYSTEM?
→ :SYSTEM:LANGUAGE JAPANESE;LCD:BRIGHTNESS 2;COLOR:GRAPH:MODE DEFAULT;COLOR:TEXT:MODE PRESET1;SCSI:OWNID 6

:SYSTem:DATE
Function Sets the date or queries the current setting.
Syntax :SYSTem:DATE {<string>}
:SYSTem:DATE?
Example :SYSTEM:DATE "99/01/01"
:SYSTEM:DATE?
→ "99/01/01"
Description The lower two digits are displayed for the year.

:SYSTem:LANGuage
Function Sets the message language or queries the current setting.
Syntax :SYSTem:LANGuage {JAPANese|ENGLish}
:SYSTem:LANGuage?
Example :SYSTEM:LANGUAGE JAPANESE
:SYSTEM:LANGUAGE?
→ :SYSTEM:LANGUAGE JAPANESE

: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;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?
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?
Function Queries all settings related to the display color of 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;CHANNEL1 7,7,0;CHANNEL2 0,7,0;CHANNEL3 7,0,7;CHANNEL4 0,7,7;CHANNEL5 7,0,0;CHANNEL6 7,4,0;CHANNEL7 0,4,7;CHANNEL8 5,5,5;MATH1 0,4,7;MATH2 5,5,5

:SYSTem:LCD:COlor:GRAPh:{BACKground|GRATicule|CURSor|CHANnel<x>|MATH<x>}
Function Queries the display color for the background/graticule/cursor/channel waveform/MATH waveform or queries the current setting.
Syntax :SYSTem:LCD:COlor:GRAPh:{BACKground|GRATicule|CURSor|CHANnel<x>|MATH<x>} {<NRf>,<NRf>,<NRf>}
:SYSTem:LCD:COlor:GRAPh:{BACKground|GRATicule|CURSor|CHANnel<x>|MATH<x>}?
Example :SYSTEM:LCD:COLOR:GRAPH:BACKGROUND 0,0,0
:SYSTEM:LCD:COLOR:GRAPH:BACKGROUND?
→ :SYSTEM:LCD:COLOR:GRAPH:BACKGROUND 0,0,0

:SYSTem:LCD:COlor:GRAPh:MODE
Function Sets the display color mode of 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:TEXT?
Function Queries all settings related to the display color of 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

4-74
:SYStem:LCd:COlor:TEXT:({LETter|BACKground|BOX|SUB|SELected})

**Function**: Sets the display colors for characters (Menu Fore)/menu background (Menu Back)/selected menu (Select Box)/popup 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

**Description**: Set the color in the order R, G, and B.

:SYStem:LCd:COlor:TEXT:MODe

**Function**: Sets the display color mode of 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

**Description**: Set the color in the order R, G, and B.

:SYStem:SCSI?

**Function**: Queries all settings related to the SCSI-ID.

**Syntax**: :SYStem:SCSI?

**Example**: :SYStem:SCSI?

**Description**: If the SCSI (option) is not installed, an error occurs.

:SYStem:SCSI:OWNId

**Function**: Sets the SCSI ID of this instrument or queries the current setting.

**Syntax**: :SYStem:SCSI:OWNId {<NRf>}

:SYStem:SCSI:OWNId?

<NRf> = 0 to 7

**Example**: :SYStem:SCSI:OWNId 6

**Description**: If the SCSI (option) is not installed, an error occurs.

: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"

**Description**: If the SCSI (option) is not installed, an error occurs.

:SYStem:SCSI:INITialize

**Function**: Initializes SCSI related settings.

**Syntax**: :SYStem:SCSI:INITialize

**Example**: :SYStem:SCSI:INITialize

**Description**: If the SCSI (option) is not installed, an error occurs.

- Make sure to execute this command, if the SCSI-ID of this instrument is changed using the ":SYStem:SCSI:OWNId" command.
4.22 TIMebase Group

The commands in the TIMebase Group deal with the time base (horizontal axis). These commands can be used to make the same settings and inquiries as when the OBSERVATION TIME knob on the front panel is pressed.

**:TIMebase?**
Function Queries all settings related to the time base (horizontal axis).
Syntax :TIMebase?
Example :TIMEBASE?→:TIMEBASE:
          OBSERVE 10000E-03;SRATE 1.00000E+06

**:TIMebase:OBServe**
Function Sets the observation time of the waveform or queries the current setting.
Syntax :TIMebase:OBServe {<time>}
          :TIMebase:OBServe?
<time> = 10us to 1ks
          (See the PZ4000 User's Manual.)
Example :TIMEBASE:OBSERVE 100MS
          :TIMEBASE:OBSERVE?→:TIMEBASE:
          OBSERVE 100.00E-03

**:TIMebase:SRATe**
Function Sets the sampling rate or queries the current setting.
Syntax :TIMebase:SRATe {<frequency>}
          :TIMebase:SRATe?
<frequency> = 50Hz to 5MHz
          (See the PZ4000 User's Manual.)
Example :TIMEBASE:SRATE 1MHz
          :TIMEBASE:SRATE?→:TIMEBASE:
          SRATE 1.00000E+06
Description The observation time is set to the optimal setting (longest range possible) depending on the specified sampling rate.
4.23 TRIGger Group

The commands in the TRIGger Group deal with the trigger. These commands can be used to make the same settings and inquiries as when the TRIGGER key on the front panel is pressed.
4.23 TRIGger Group

**.TRIGger?**
Function: Queries all settings related to the trigger.
Syntax: .TRIGger?
Example: .TRIGger? → .TRIGger:MODE AUTO;SOURCE 1; TYPE EDGE;EDGE:SLOPE RISE; LEVEL 1.000E+03; .TRIGger:DREFERENCE 10; DELAY 0.0E+00; ACTION:SAVE 0; HCOPY 0; ACQCOUNT INFINITE

**.TRIGger:ACTion?**
Function: Queries all settings related to action-on-trigger.
Syntax: .TRIGger:ACTion?
Example: .TRIGGER:ACTION?
→ .TRIGGER:ACTion:
SAVE 0; HCOPY 0; ACQCOUNT INFINITE

**.TRIGger:ACTion:ACQCount**
Function: Sets the action count of action-on-trigger or queries the current setting.
Syntax: .TRIGger:ACTion:ACQCount {<NRf> | INFinite}
Example: .TRIGGER:ACTION:ACQCount 10
→ .TRIGGER:ACTion:ACQCount 10

**.TRIGger:ACTion:HCOPY**
Function: Sets whether or not to output screen image data (ON/OFF) when an action is activated, or queries the current setting.
Syntax: .TRIGger:ACTion:HCOPY {<Boolean>}
Example: .TRIGGER:ACTion:HCOPY ON
→ .TRIGGER:ACTion:HCOPY 1

**.TRIGger:ACTion:SAVE**
Function: Sets whether or not to save the waveform data to the storage medium (ON/OFF) when an action is activated, or queries the current setting.
Syntax: .TRIGger:ACTion:SAVE {<Boolean>}
Example: .TRIGGER:ACTion:SAVE ON
→ .TRIGGER:ACTion:SAVE 1

**.TRIGger:DELay**
Function: Sets the trigger delay or queries the current setting.
Syntax: .TRIGger:DELay {<time>}
Example: .TRIGGER:DELAY 0
→ .TRIGGER:DELAY 0

**.TRIGger:DREFerence (Delay REFerence)**
Function: Sets the trigger position or queries the current setting.
Syntax: .TRIGger:DREFerence {<NRf>}
Example: .TRIGGER:DREFERENCE 10
→ .TRIGGER:DREFERENCE 10

**.TRIGger:EDGE?**
Function: Queries all settings related to the edge trigger.
Syntax: .TRIGger:EDGE?
Example: .TRIGGER:EDGE?
→ .TRIGGER:EDGE:
SLOPE RISE; LEVEL 0.0

**.TRIGger:EDGE:LEVel**
Function: Sets the trigger level for the edge trigger or queries the current setting.
Syntax: .TRIGger:EDGE:LEVel {<NRf>}
Example: .TRIGGER:EDGE:LEVEL 1000V
→ .TRIGGER:EDGE:LEVEL 0.0

**.TRIGger:EDGE:SLOPe**
Function: Sets the trigger slope for the edge trigger or queries the current setting.
Syntax: .TRIGger:EDGE:SLOPe {RISE|FALL|BOTH}
Example: .TRIGGER:EDGE:SLOPe RISE
4.23 TRIGger Group

:TRIGger:MODE
Function Sets the trigger mode or queries the current setting.
Syntax :TRIGger:MODE {OFF|AUTO|ALevel|NORMal|HFAuto|HFNORMAL}
Example :TRIGGER:MODE AUTO

:TRIGger:SOURce
Function Sets the trigger source or queries the current setting.
Syntax :TRIGger:SOURce {<NRf>|EXTernal}
Example :TRIGGER:SOURCE 1

:TRIGger:TYPE
Function Sets the trigger type or queries the current setting.
Syntax :TRIGger:TYPE {EDGE|WINDow}
Example :TRIGGER:TYPE EDGE

:TRIGger:WINDow
Function Queries all settings related to the window trigger.
Syntax :TRIGger:WINDow?
Example :TRIGGER:WINDOW?

:TRIGger:WINDow:CENTer
Function Sets the center level for the window trigger or queries the current setting.
Syntax :TRIGger:WINDow:CENTer {<NRf>}
Example :TRIGGER:WINDOW:CENTER 0.0

:TRIGger:WINDow:CONDition
Function Sets the trigger condition for the window trigger or queries the current setting.
Syntax :TRIGger:WINDow:CONDition {IN|OUT}
Example :TRIGGER:WINDOW:CONDITION IN

:TRIGger:WINDow:WIDTH
Function Sets the window width for the window trigger or queries the current setting.
Syntax :TRIGger:WINDow:WIDTH {<NRf>}
Example :TRIGGER:WINDOW:WIDTH 25
4.24 WAVeform Group

The commands in the WAVeform Group deal with the acquired waveform data. There are no front-panel keys that correspond to the commands in this group.

;WAVeform?
Function Queries all settings related to the waveform data.
Syntax ;WAVeform?
Example ;WAVeform?
"WAVEFORM?
→ ;WAVEFORM:TRACE 1;
FORMAT BINARY;BYTEORDER LSBFIRST;
START 0;END 100"

;WAVeform:BYTEorder
Function Sets the byte order of the waveform data that are sent using the ";WAVeform:SEND?" command or queries the current setting.
Syntax ;WAVeform:BYTEorder {LSBFirst|MSBFirst}
;WAVeform:BYTEorder?
Example ;WAVeform:BYTEorder LSBFIRST
;WAVEFORM:BYTEORDER LSBFIRST
;WAVEFORM:BYTEORDER?
→ ;WAVEFORM:BYTEORDER LSBFIRST
Description This setting is valid when ";WAVeform:FORMat" is set to {BINary|FLoat}.

;WAVeform:END
Function Sets the end point of the output of the waveform data that are sent using the ";WAVeform:SEND?" command or queries the current setting.
Syntax ;WAVeform:END {<NRf>}
;WAVeform:END?
Example ;WAVEFORM:END 100
;WAVEFORM:END?→ ;WAVEFORM:END 100
Description (The total number of data points) can be queried using the ";WAVEform:LENGth?" command.
4.24 WAVeform Group

**:WAVeform:FORMat**

**Function**
Sets the format of the waveform data that are sent using the ":WAVeform:SEND?" command or queries the current setting?

**Syntax**
`:WAVeform:FORMat {ASCii|BINary|FLOat}

**:WAVeform:FORMat?**

**Example**
`:WAVEFORM:FORMAT BINARY

**:WAVeform:FORMat?**

**Description**
For the differences in the waveform data output depending on the format setting, refer to the **:WAVeform:SEND?** command.

**:WAVeform:LENGTH?**

**Function**
Queries the total number of data points of the waveform that is specified using the ":WAVeform:TRACe" command.

**Syntax**
`:WAVeform:LENGTH?

**:WAVeform:LENGTH?**→100001

**Description**
- The total number of data points varies depending on the record length (ON/OFF state of dividing the record length) and observation time (sampling rate).
- For details, see the PZ4000 User's Manual.

**:WAVeform:RANGE?**

**Function**
Queries the range value that is used to convert the waveform specified using the ":WAVeform:TRACe" command to physical data.

**Syntax**
`:WAVeform:RANGE?

**:WAVeform:RANGE?**→250.00E+00

**Description**
- This range value is used when converting the waveform to physical values when the ":WAVeform:FORMat" is set to BINary.
- ‘0’ is returned when ":WAVeform:TRACe" is set to MATH<x>.

**:WAVeform:SEND?**

**Function**
Queries the waveform data that are specified using the ":WAVeform:TRACe" command.

**Syntax**
`:WAVeform:SEND?

**Example**
- When ":WAVeform:FORMat" is set to {ASCii}
  
  :WAVEFORM:SEND?→<NR3>,<NR3>,...  
- When ":WAVeform:FORMat" is set to {BINary|FLOat}
  
  :WAVEFORM:SEND?→#8(Number of bytes, 8 digits)(Series of data bytes)

**Description**
The format of the numerical data that is output depends on the ":NUMeric:FORMat" setting.

1. When set to ‘ASCii’
   
   The physical values are output in <NR3> format. Each item of data is separated by a comma.

2. When set to ‘BINary’
   
   The A/D value before it is converted to a physical value is output in WORD format (2 bytes, 0 to FFFH, unsigned).
   
   The output byte order of each data point follows the order that is set using the ":WAVeform:BYTeorder" command.
   
   The equation used to convert to a physical value is
   \[
   \text{Physical value} = \frac{\text{WORD data} - 2048}{2048} \times \text{range value.}
   \]
   
   Binary output is not possible when ":WAVeform:TRACe" is set to MATH<x>. All 0s are returned. Inquire using the FLOat format in this case.

3. When set to ‘FLOat’
   
   The physical values are output in IEEE single precision floating point format (4 bytes).
   
   The output byte order of each data point follows the order that is set using the ":WAVeform:BYTeorder" command.

**:WAVeform:SRATe?**

**Function**
Queries the sampling rate of the acquired data.

**Syntax**
`:WAVeform:SRATe?

**Example**
`:WAVEFORM:SRATE?→1.00000E+06
### :WAVeform::START

**Function**
Sets the start point of the output of the waveform data that are sent using the ":WAVeform:SEND?" command or queries the current setting.

**Syntax**
```
:WAVeform::START {<NRf>}
:WAVeform::START?
```
- `<NRf>` = 0 to (Total number of data points - 1)

**Example**
- `:WAVEFORM::START 0`
- `:WAVEFORM::START?` → `:WAVEFORM::START 0`

**Description**
(The total number of data points) can be queried using the ":WAVeform:LENGTH?" command.

### :WAVeform:TDATe?

**Function**
Queries the string containing the trigger date and time when the waveform was acquired.

**Syntax**
`:WAVeform:TDATe?`

**Example**
- `:WAVEFORM:TDATE?` → "1999/12/23 12:34:56"

**Description**
The date and time is separated by one space character.

### :WAVeform:TRACe

**Function**
Sets the target waveform in the waveform group or queries the current setting.

**Syntax**
```
:WAVeform:TRACe {<NRf>|MATH<x>}
:WAVeform:TRACe?
```
- `<NRf>` = 1 to 8 (channel)
- `<x>` = 1, 2 (MATH)

**Example**
- `:WAVEFORM:TRACE 1`
- `:WAVEFORM:TRACE?` → `:WAVEFORM:TRACE 1`

### :WAVeform:TRIGger?

**Function**
Queries the trigger position of the acquired data.

**Syntax**
`:WAVeform:TRIGger?`

**Example**
- `:WAVEFORM:TRIGGER?` → 10000

**Description**
The number of points from the beginning of the record length to the trigger position.

### :WAVeform:ZCRoss?

**Function**
Queries zero crossing data of all channels.

**Syntax**
`:WAVeform:ZCRoss?`

**Example**
- `:WAVEFORM:ZCROSS?` → #8(Number of bytes, 8 digits)(Series of data bytes)

**Description**
- The output start and end points of zero crossing data are specified using the ":WAVeform:{START|END}" command in the same fashion as for the waveform data.
- The data format of each output point is fixed to WORD (2-byte) format.
- The output byte order follows the order that is set using the ":WAVeform:BYTeorder" command.
4.25 ZOOM Group

The commands in the ZOOM Group deal with the zooming of the waveform. These commands can be used to make the same settings and inquiries as when the ZOOM key on the front panel is pressed.

:ZOOM?
Function Queries all settings related to the zooming of the waveform.
Syntax :ZOOM?
Example :ZOOM?→:ZOOM:MODE MAIN_Z1_Z2;
FORM:_SINGLE;
ALLO:CHANNEL1 1;
CHANNEL2 0;CHANNEL3 0;CHANNEL4 0;
CHANNEL5 0;CHANNEL6 0;CHANNEL7 0;
CHANNEL8 0;MATH1 0;MATH2 0;ZOOM:MA1 2;
MAG2 2;POSITION1 25.000E-03;
POSITION2 75.000E-03

:ZOOM:ALLOcation?
Function Queries all settings related to the zoomed waveform.
Syntax :ZOOM:ALLOcation?
Example :ZOOM:ALLOcation?→:ZOOM:ALLOcation:
ALLO:CHANNEL1 1;
CHANNEL2 0;CHANNEL3 0;
CHANNEL4 0;CHANNEL5 0;CHANNEL6 0;
CHANNEL7 0;CHANNEL8 0;MATH1 0;MATH2 0
4.25 ZOOM Group

:ZOOM:ALLOcation:{CHANnel<x>|MATH<x>}
Function Sets whether or not to select the waveform to be zoomed or queries the current setting.
Syntax :ZOOM:ALLOcation:{CHANnel<x>|MATH<x>} {<Boolean>}
:ZOOM:ALLOcation:{CHANnel<x>|MATH<x>}?
<x> of CHANnel<x> = 1 to 8
<x> of MATH<x> = 1, 2
Example :ZOOM:ALLOCATION:CHANNEL1 ON
:ZOOM:ALLOCATION:CHANNEL1?
→ :ZOOM:ALLOCATION:CHANNEL1 1

:ZOOM:FORMat
Function Sets the display format of the zoomed waveform or queries the current setting.
Syntax :ZOOM:FORMat {MAIN|SINGle|DUAL|TRIad|QUAD}
:ZOOM:FORMat?
Example :ZOOM:FORMAT SINGLE
:ZOOM:FORMAT?
→ :ZOOM:FORMAT SINGLE

:ZOOM:MAG<x>
Function Sets the zoom factor or queries the current setting.
Syntax :ZOOM:MAG<x> {<NRf>}
:ZOOM:MAG<x>?
<x> = 1, 2
<NRf> = 2 to 100000 (See the PZ4000 User’s Manual)
Example :ZOOM:MAG1 2
:ZOOM:MAG1?
→ :ZOOM:MAG1 2
Description The selectable zoom factor varies depending on the measurement mode, the observation time, the record length, and the record length division settings.

:ZOOM[:MODE]
Function Sets the the display mode of the zoomed waveform or queries the current setting.
Syntax :ZOOM[:MODE] {MAIN|MAIN_Z1|Z1|MAIN_Z1_Z2|MAIN_Z2|Z2|Z1_Z2}
:ZOOM:MODE?
Example :ZOOM:MODE MAIN_Z1_Z2
:ZOOM:MODE?
→ :ZOOM:MODE MAIN_Z1_Z2

:ZOOM:POsition<x>
Function Sets the position of the zoom box or queries the current setting.
Syntax :ZOOM:POsition<x> {<time>|<NRf>}
:ZOOM:POsition<x>?
<x> = 1, 2
<time> = 0 to (OBSERVATION TIME) (during the normal measurement mode, when Time Base = Internal)
<NRf> = 0 to Record length (when Time Base = Internal, or during the harmonic measurement mode)
Example :ZOOM:POSITION1 25MS
:ZOOM:POSITION1?
→ :ZOOM:POSITION1 25.000E-03
Description • The range and resolution of <time> depends on the observation time.
• Specify <NRf> in terms of sampled data points. The range is from 0 to the record length.
4.26 Common Command Group

The commands in the common command group are independent of the instrument's functions and are specified in IEEE 488.2-1987. There is no front-panel key that corresponds to this group.

**CAL? (CALibrate)**
- **Function**: Performs calibration (zero level compensation, same operation as pressing the CAL key) and queries the result.
- **Syntax**: *CAL?
- **Example**: *CAL?
- **Description**: "0" is returned when the calibration completes properly. "1" is returned if there is an abnormality.

**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 immediately after the program message terminator, the output queue is also cleared.
  • For details related to the registers and queues, see chapter 5.

**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**: • Set the value using a decimal sum of each bit.
  • For example, if "*ESE 251" is set, the standard event enable register is set to "11111011." This means that bit 2 of the standard event register is disabled so that bit 5 (ESB) of the status register will not be set to "1," even if a query error occurs.
  • The default setting is "*ESE 0" (all bits disabled).
  • The standard event enable register is not cleared even if an inquiry is made using *ESE?.
  • For details related to the standard event enable register, see page 5-3.
4.26 Common Command Group

*ESR? (standard Event Status Register)
Function Queries the standard event register and clears the register.
Syntax *ESR?
Example *ESR? \rightarrow 32
Description • Returns the sum of each bit expressed as a decimal number.
• You can determine what type of event occurred when SRQ occurred.
• For example, if "32" is returned, it indicates that the standard event register is set to "00100000." This means that SRQ occurred because a "command syntax error" error occurred.
• The standard event register is cleared if an inquiry is made using *ESR?.
• For details related to the standard event register, see page 5-3.

*IDN? (IDeNtify)
Function Queries the instrument model.
Syntax *IDN?
Example *IDN? \rightarrow YOKOGAWA,253710,0,F1.10
Description The response is returned in the following format: <Maker>, <Model>, <Serial No.>, <Firmware version>. The <Serial No.> is always set to 0.

*OPC (OPeration Complete)
Function Sets bit 0 of the standard event register (OPC bit) to 1 when the specified overlap command completes.
Syntax *OPC
Example *OPC
Description • For the description regarding how to synchronize the program using the *OPC command, see page 3-7.
• The "COMMunicate:OPSE" command is used to specify the overlap commands.
• If the *OPC command is not placed at the end of the message, the operation is not guaranteed.

*OPC? (OPeration Complete)
Function If the specified overlap command has been completed when *OPC? is sent, ASCII code "1" is returned.
Syntax *OPC?
Example *OPC? \rightarrow 1
Description • For the description regarding how to synchronize the program using the *OPC? command, see page 3-13.
• The "COMMunicate:OPSE" command is used to specify the overlap commands.
• If the *OPC? command is not placed at the end of the message, the operation is not guaranteed.

*OPT? (OPTion)
Function Queries installed options.
Syntax *OPT?
Example *OPT? \rightarrow M1,PRINTER,SCSI
Description • Returns whether or not the following items exist: <Extended memory>, <Built-in printer>, <SCSI>
• The "*OPT?" command must be the last query in a program message. Otherwise, an error occurs.

*PSC (Power-on Status Clear)
Function Sets whether or not to clear the following registers on power up or queries the current setting. The registers are cleared if the value that is rounded to an integer is a non-zero number.
• Standard event enable register
• Extended event enable register
• Transition filter
Syntax *PSC {<NRf>}
Example *PSC \rightarrow 1
Description For details regarding the registers, see chapter 5.

*RST (ReSeT)
Function Initializes the settings.
Syntax *RST
Example *RST
Description • *OPC and *OPC? that were sent earlier are also reset.
• Resets all setup parameters except communication settings 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?

Example
*SRE 239
*SRE?→175 (because bit 6 (MSS) is ignored)

Description
- Set the value using a decimal sum of each bit.
- For example, if "*SRE 239" is set, the service request enable register is set to "11101111." This means that bit 4 of the standard event register is disabled so that bit 4 (MAV) of the status register will not be set to "1," even if the "output queue is not empty."
- However, bit 6 of the status byte is the MSS bit, so it is ignored.
- The default setting is "*SRE 0" (all bits disabled).
- The service request enable register is not cleared even if an inquiry is made using *SRE?.
- For details related to the service request enable register, see page 5-1.

*STB? (Status Byte)

Function
Queries the status byte register.

Syntax
*STB?

Example
*STB?→4

Description
- Returns the sum of each bit expressed as a decimal number.
- Since the register is read without serial polling, bit 6 is the MSS bit, not RQS.
- For example, if "4" is returned, it indicates that the standard event register is set to "00000100." This means that SRQ occurred because the "error queue is not empty."
- The status byte register is not cleared, even if an inquiry is made using *STB?.
- For details related to the status byte register, see page 5-2.

*TRG (Trigger)

Function
Executes single start (the same as pressing the SINGLE START key).

Syntax
*TRG

Example
*TRG

Description
- The multi-line message GET (Group Execute Trigger) operates in the same way as this command.

*TST? (Test)

Function
Executes the self-test and queries the result.

Syntax
*TST?

Example
*TST?→0

Description
- The self-test involves the testing of the internal memories.
- "0" is returned if the self-test is successful.
- "1" is returned otherwise.

*WAI (Wait)

Function
Waits until the execution of the specified overlap command completes before executing the commands that are specified after this command.

Syntax
*WAI

Example
*WAI

Description
- For the description regarding how to synchronize the program using the *WAI command, see page 3-7.
- The "COMMunicate:OPSE" command is used to specify the overlap commands.
Chapter 5 Status Report

5.1 Overview of the Status Report

The figure below shows the status report which is read by a serial poll. This is an extended version of the one specified in IEEE 488.2-1987.
Overview of Registers and Queues

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Writing</th>
<th>Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status byte</td>
<td>Masks status byte.</td>
<td>*SRE?</td>
<td>*SRE?</td>
</tr>
<tr>
<td>Service request enable register</td>
<td>Masks status byte.</td>
<td>*SRE?</td>
<td>*SRE?</td>
</tr>
<tr>
<td>Standard event register</td>
<td>Change in device status</td>
<td>—</td>
<td>*SRE?</td>
</tr>
<tr>
<td>Standard event enable register</td>
<td>Masks standard event register</td>
<td>*ESE?</td>
<td>*ESE?</td>
</tr>
<tr>
<td>Extended event register</td>
<td>Change in device status</td>
<td>—</td>
<td>STATus:EESE?</td>
</tr>
<tr>
<td>Extended event enable register</td>
<td>Masks standard event register</td>
<td>STATus:EESE?</td>
<td>STATus:EESE?</td>
</tr>
<tr>
<td>Condition register</td>
<td>Current instrument status</td>
<td>—</td>
<td>STATus:CONDITION?</td>
</tr>
<tr>
<td>Transit filter</td>
<td>Extended event occurrence conditions</td>
<td>STATus:FILTer</td>
<td>STATus:FILTer&lt;&lt;x&gt;7</td>
</tr>
<tr>
<td>Output queue</td>
<td>Stores response message All executable queues to a query</td>
<td>STATus:ERRor?</td>
<td></td>
</tr>
<tr>
<td>Error queue</td>
<td>Stores error Nos. and messages</td>
<td>—</td>
<td>STATus:ERRor?</td>
</tr>
</tbody>
</table>

Registers and Queues which Affect the Status Byte

Registers which affect each bit of the status byte are shown below.

- **Standard event register**: Sets bit 5 (ESB) of status byte to “1” or “0”.
- **Output queue**: Sets bit 4 (MAV) of status byte to “1” or “0”.
- **Extended event register**: Sets bit 3 (EES) of status byte to “1” or “0”.
- **Error queue**: Sets bit 2 (EAV) of status byte to “1” or “0”.

Enable Registers

Registers which mask a bit so that the bit does not affect the status byte, even if the bit is set to “1”, are shown below.

- **Status byte**: Masks bits using the service request enable register.
- **Standard event register**: Masks bits using the standard event enable register.
- **Extended event register**: Masks bits using the extended event enable register.

Writing/Reading from Registers

The *ESE command is used to set bits in the standard event enable register to “1” or “0”, and the *ESR? query is used to check whether bits in that register are set to “1” or “0”. For details of these commands, refer to Chapter 4.

5.2 Status Byte

Overview of Status Byte

<table>
<thead>
<tr>
<th>RGS</th>
<th>7 6 ESB MAV EES EAV 1 0</th>
</tr>
</thead>
</table>

**Bits 0, 1 and 7**

Not used (always “0”)

**Bit 2 EAV (Error Available)**

Set to “1” when the error queue is not empty, i.e. when an error occurs. For details, refer to page 5-5.

**Bit 3 EES (Extended Event Summary Bit)**

Sets to “1” when the logical “AND” of an Extended Event Register bit and the corresponding Enable Register bit is equal to “1”—that is, when an event takes place in the instrument. Refer to page 5-4.

**Bit 4 MAV (Message Available)**

Set to “1” when the output queue is not empty, i.e. when there is data which is to be output when an query is made. Refer to page 5-5.

**Bit 5 ESB (Event Summary Bit)**

Set to “1” when the logical AND of the standard event register and the corresponding enable register is “1”, i.e. when an event takes place in the instrument. Refer to page 5-3.

**Bit 6 RQS (Request Status)/MSS (Master Summary Status)**

Sets to “1” when the logical “AND” of any one of the Status Byte bits (other than bit 6) and the corresponding Service Request Enable Register bit becomes “1”—that is, when the instrument is requesting service from the controller.

RQS is set to “1” when MSS changes from “0” to “1”, and is cleared when a serial poll is performed or when MSS changes to “0”.

**Bit Masking**

To mask a bit in the status byte so that it does not cause an SRQ, set the corresponding bit of the service request enable register to “0”.

For example, to mask bit 2 (EAV) so that no service will be requested, even if 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 of the *SRE command, refer to Chapter 4.
Operation of the Status Byte
A service request is issued when bit 6 of the status byte becomes “1”. Bit 6 becomes “1” when any of the other bits becomes “1” (or when the corresponding bit in the service request enable register becomes “1”). For example, if an event occurs causing the logical AND of any one bit in the standard event register and the corresponding bit of the enable register to become “1”, bit 5 (ESB) is set to “1.” In this case, if bit 5 of the service request enable register is “1”, bit 6 (MSS) will be set to “1”, thus requesting service from the controller.

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 query 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 poll
  Execution of a serial poll changes bit 6 to RQS. This causes RQS to be read. After completion of the read-out, only RQS is cleared. Using a serial poll, it is not possible to read MSS.

Clearing the Status Byte
No method is provided for forcibly clearing all the bits in the status byte. Bits which are cleared are shown below.

- When an query is made using the *STB? query
  No bit is cleared.

- When a serial poll is performed
  Only the RQS bit is cleared.

- When the *CLS command is received
  When the *CLS command is received, the status byte itself is not cleared, but the contents of the standard event register (which affects the bits in the status byte) are cleared. As a result, the corresponding bits in the status byte are cleared, except bit 4 (MAV), since the output queue cannot be emptied by the *CLS command. However, the output queue will also be cleared if the *CLS command is received just after a program message terminator.

5.3 Standard Event Register
Overview of the Standard Event Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PON</td>
<td>URQ</td>
<td>CME</td>
<td>EXE</td>
<td>DDE</td>
<td>QYE</td>
<td>RQC</td>
<td>OPC</td>
<td></td>
</tr>
</tbody>
</table>

Bit 7 PON (Power ON)
Bit 7 PON (Power ON) Set to “1” when 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.
Examples: Incorrectly spelled command name; received string data that have spelling errors or that are 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.
Examples: Parameters are outside the setting range; received a command that has a parameter that is outside the range or a command that deals with an option that is not installed.

Bit 3 DDE (Device Dependent Error)
Set to “1” when execution of the command is not possible due to an internal problem in the instrument that is not a command error or an execution error.
Example: The circuit breaker is reset.

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.
Examples: 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 has been completed. Refer to Chapter 4.

Bit Masking
To mask a bit in the standard event register so that it does not cause bit 5 (ESB) of the status byte to change, set the corresponding bit in the standard event enable register to “0”.
For example, to mask bit 2 (QYE) so that ESB will not be set to “1”, even if a query error occurs, set bit 2 of the standard event enable register to “0”. This can be done using the *ESE command. To inquire whether each bit of the standard event enable register is “1” or “0”, use the *ESE?. For details of the *ESE command, refer to Chapter 4.
5.3 Standard Event Register/5.4 Extended Event Register

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”).

Examples
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 completion of the read-out, the register will be 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 *ESR?
• When the *CLS command is received
• When power is turned ON again

5.4 Extended Event Register

Reading the extended event register tells you whether changes in the condition register (reflecting internal conditions) have occurred. A filter can be applied which allows you to decide which events are reported to the extended event register.

The meaning of each bit of the condition register is as follows.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RUN (Running)</td>
<td>Set to “1” during acquisition.</td>
</tr>
<tr>
<td>1</td>
<td>DAV (numeric Data Available)</td>
<td>Set to “1” when the numerical data are updated. The update is complete when DAV is set.</td>
</tr>
<tr>
<td>3</td>
<td>CAL (Calibrating)</td>
<td>Set to “1” during calibration.</td>
</tr>
<tr>
<td>4</td>
<td>TST (Testing)</td>
<td>Set to “1” during self-test.</td>
</tr>
<tr>
<td>5</td>
<td>PRN (Printing)</td>
<td>Set to “1” while the built-in printer is in operation.</td>
</tr>
<tr>
<td>6</td>
<td>ACS (Accessing)</td>
<td>Sets to “1” while floppy drive, or external SCSI device is being accessed.</td>
</tr>
<tr>
<td>7</td>
<td>PLLE (PLL source input Error)</td>
<td>Set to “1” during harmonic measurement mode, when there is no input at the PLL source and synchronization cannot be achieved.</td>
</tr>
<tr>
<td>8</td>
<td>POV 1 (ch1 input Peak Over)</td>
<td>Set to “1” when channel 1 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>9</td>
<td>POV 2 (ch2 input Peak Over)</td>
<td>Set to “1” when channel 2 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>10</td>
<td>POV 3 (ch3 input Peak Over)</td>
<td>Set to “1” when channel 3 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>11</td>
<td>POV 4 (ch4 input Peak Over)</td>
<td>Set to “1” when channel 4 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>12</td>
<td>POV 5 (ch5 input Peak Over)</td>
<td>Set to “1” when channel 5 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>13</td>
<td>POV 6 (ch6 input Peak Over)</td>
<td>Set to “1” when channel 6 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>14</td>
<td>POV 7 (ch7 input Peak Over)</td>
<td>Set to “1” when channel 7 input detects a signal that exceeds the range.</td>
</tr>
<tr>
<td>15</td>
<td>POV 8 (ch8 input Peak Over)</td>
<td>Set to “1” when channel 8 input detects a signal that exceeds the range.</td>
</tr>
</tbody>
</table>
The filter is applied to each bit of the condition register separately, and can be selected from the following. Note that the numbering of the bits used in the filter setting differs from the actual bit number (1 to 16 vs. 0 to 15).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rise</td>
<td>The bit of the extended event register becomes “1” when the bit of the condition register changes from “0” to “1”.</td>
</tr>
<tr>
<td>Fall</td>
<td>The bit of the extended event register becomes “1” when the bit of the condition register changes from “1” to “0”.</td>
</tr>
<tr>
<td>Both</td>
<td>The bit of the extended event register becomes “1” when the bit of the condition register changes from “0” to “1”, or from “1” to “0”.</td>
</tr>
<tr>
<td>Never</td>
<td>The bit of the extended event register is disabled and always “0”.</td>
</tr>
</tbody>
</table>

### 5.5 Output Queue and Error Queue

#### Overview of the Output Queue

The output queue is provided to store response messages to queries. For example, when the `WAVeform:SEND?` query is sent to request output of the acquired waveform, the response data will be stored in the output queue until it is read out. The example below shows that data is stored record by record in the output queue, and is read out oldest item first, newest item last. 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 dead lock occurs (page 3-2)
- When a device clear command (DCL or SDC) is received
- When power is turned ON again

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.

#### Overview of the 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 number, “113, “Undefined header””, and the error message 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, messages are read oldest first, newest last (refer to the previous page).

If the error queue becomes full, the final message will be replaced by message “350, “Queue overflow””.

The error queue is emptied in the following cases (in addition to when read-out is performed):

- When the *CLS command is received
- When power is turned ON again

To see whether the error queue is empty or not, check bit 2 (EAV) of the status byte.
6.1 Before Programming

Environment

Model: MS-DOS computer equipped with AT-GPIB/TNT IEEE-488.2 board from National Instruments.
Language: Quick BASIC

Setting up the PZ4000

• Address 1
  All the sample programs given in this chapter use address 1 for the PZ4000 so be sure to assign the instrument to address 1 as described on page 1-5.

• Data Acquisition “STOP”
  The sample programs in this chapter are written with the premise that the data acquisition on the instrument is in the “STOP” condition. If the data acquisition is in progress, press the “START/STOP” key so that “Stopped” is displayed in the lower left section of the screen. Then, execute the program.
6.2 Example of Normal Measurement Data Output

```basic
REM $INCLUDE: 'qbdecl4.bas'
OPTION BASE 1 ' Minimum value of array subscript = 1
DIM D$(40) ' Array of numerical data strings

DEVICE$ = "DEV1": CALL IBFIND(DEVICE$, PZ%)
CALL IBSIC(PZ%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%) ' Remote setting

' Set the measurement condition and range
CMD$ = "SETUP:MODE NORMAL" ' Normal measurement mode
CALL IBWRT(PZ%, CMD$)
CMD$ = "VOLTAGE:RANGE 200" ' Voltage range = 200Vpk
CALL IBWRT(PZ%, CMD$)
CMD$ = "CURRENT:RANGE 4" ' Current range = 4Apk
CALL IBWRT(PZ%, CMD$)
CMD$ = "FILTER:LINE OFF" ' Line filter = OFF
CALL IBWRT(PZ%, CMD$)
CMD$ = "TIMEBASE:OBSERVE 100MS" ' Observation time = 100msec
CALL IBWRT(PZ%, CMD$)

' Set the numerical data output items (ASCII format, preset to pattern 1, number of output data = 40)
CMD$ = "NUMERIC:FORMAT ASCII;NORMAL:PRESET 1;NUMBER 40"
CALL IBWRT(PZ%, CMD$)

' Set the transition filter used to detect the completion of the numerical data updating
CMD$ = "STATUS:FILTER2 RISE" ' Rising edge of bit1 (DAV)
CALL IBWRT(PZ%, CMD$)

' Clear the extended event register(Read and trash the response)
CMD$ = "STATUS:EESR?"
CALL IBWRT(PZ%, CMD$)
RG$ = SPACE$(8)
CALL IBRD(PZ%, RG$)

' Measurement start
CMD$ = "START"
CALL IBWRT(PZ%, CMD$)

' Read and display the numerical data (Repeated 10 times in this program)
FOR I% = 1 TO 10
    ' Wait for the completion of the numerical data updating
    CMD$ = "COMMUNICATE:WAIT 2"
    CALL IBWRT(PZ%, CMD$)
    ' Clear the extended event register (Read and trash the response)
    CMD$ = "STATUS:EESR?"
    CALL IBWRT(PZ%, CMD$)
    RG$ = SPACE$(8)
    CALL IBRD(PZ%, RG$)
    ' Read out numerical data
    CMD$ = "NUMERIC:NORMAL:VALUE?"
    CALL IBWRT(PZ%, CMD$)
    RES$ = SPACE$(1000)
    CALL IBRD(PZ%, RES$)
    ' Extract items that are separated by commas (,) from the received numerical data
    C$ = LEFT$(RES$, IBCNT%)
    IBCNT% = Number of received bytes
    FOR J% = 1 TO 40
        LN% = LEN(C$)
        LN% = LN% - 1
        IF LN% = 0 THEN LN% = LN% + 1
        DS(J%) = LEFT$(C$, (LN% - 1))
        C$ = MID$(C$, (LN% + 1))
    NEXT J%
```

REM $INCLUDE: 'qbdecl4.bas'
OPTION BASE 1 ' Minimum value of array subscript = 1
DIM D$(40) ' Array of numerical data strings

DEVICE$ = "DEV1": CALL IBFIND(DEVICE$, PZ%)
CALL IBSIC(PZ%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%) ' Remote setting

' Set the measurement condition and range
CMD$ = "SETUP:MODE NORMAL" ' Normal measurement mode
CALL IBWRT(PZ%, CMD$)
CMD$ = "VOLTAGE:RANGE 200" ' Voltage range = 200Vpk
CALL IBWRT(PZ%, CMD$)
CMD$ = "CURRENT:RANGE 4" ' Current range = 4Apk
CALL IBWRT(PZ%, CMD$)
CMD$ = "FILTER:LINE OFF" ' Line filter = OFF
CALL IBWRT(PZ%, CMD$)
CMD$ = "TIMEBASE:OBSERVE 100MS" ' Observation time = 100msec
CALL IBWRT(PZ%, CMD$)

' Set the numerical data output items (ASCII format, preset to pattern 1, number of output data = 40)
CMD$ = "NUMERIC:FORMAT ASCII;NORMAL:PRESET 1;NUMBER 40"
CALL IBWRT(PZ%, CMD$)

' Set the transition filter used to detect the completion of the numerical data updating
CMD$ = "STATUS:FILTER2 RISE" ' Rising edge of bit1 (DAV)
CALL IBWRT(PZ%, CMD$)

' Clear the extended event register(Read and trash the response)
CMD$ = "STATUS:EESR?"
CALL IBWRT(PZ%, CMD$)
RG$ = SPACE$(8)
CALL IBRD(PZ%, RG$)

' Measurement start
CMD$ = "START"
CALL IBWRT(PZ%, CMD$)

' Read and display the numerical data (Repeated 10 times in this program)
FOR I% = 1 TO 10
    ' Wait for the completion of the numerical data updating
    CMD$ = "COMMUNICATE:WAIT 2"
    CALL IBWRT(PZ%, CMD$)
    ' Clear the extended event register (Read and trash the response)
    CMD$ = "STATUS:EESR?"
    CALL IBWRT(PZ%, CMD$)
    RG$ = SPACE$(8)
    CALL IBRD(PZ%, RG$)
    ' Read out numerical data
    CMD$ = "NUMERIC:NORMAL:VALUE?"
    CALL IBWRT(PZ%, CMD$)
    RES$ = SPACE$(1000)
    CALL IBRD(PZ%, RES$)
    ' Extract items that are separated by commas (,) from the received numerical data
    C$ = LEFT$(RES$, IBCNT%)
    IBCNT% = Number of received bytes
    FOR J% = 1 TO 40
        LN% = LEN(C$)
        LN% = LN% - 1
        IF LN% = 0 THEN LN% = LN% + 1
        DS(J%) = LEFT$(C$, (LN% - 1))
        C$ = MID$(C$, (LN% + 1))
    NEXT J%
' Display the numerical data
PRINT I%, "Element1", "Element2", "Element3", "Element4"
PRINT "Urms [V]", D$(1), D$(11), D$(21), D$(31)
PRINT "Irms [A]", D$(2), D$(12), D$(22), D$(32)
PRINT "P [W]", D$(3), D$(13), D$(23), D$(33)
PRINT "S [VA]", D$(4), D$(14), D$(24), D$(34)
PRINT "Q [var]", D$(5), D$(15), D$(25), D$(35)
PRINT "\(\Lambda\)\[]\]", D$(6), D$(16), D$(26), D$(36)
PRINT "\(\Phi\)\[]\]", D$(7), D$(17), D$(27), D$(37)
PRINT "fU [Hz]", D$(8), D$(18), D$(28), D$(38)
PRINT "fI [Hz]", D$(9), D$(19), D$(29), D$(39)
PRINT
NEXT I%
' Measurement stop
CMD$ = "STOP"
CALL IBWRTPZ%, CMD$)
V% = 0: CALL IBRSREBD%, V%) ' Clear remote mode
END

***********************************************************************
* * PZ4000 Sample Program1 for serial(RS-232) interface            *
* * Microsoft QuickBASIC 4.0/4.5 Version                           *
* * Rate:9600 Parity:None CHR:8 STOPBIT:1 XON/XON Term:CR+LF       *
* ***********************************************************************
* * In the normal measurement mode, set the measurement conditions *
* * and start the measurement.                                      *
* * The following numerical data (ASCII format) are read and displayed *
* * on every update.                                                  *
* * voltage (Urms), current (Irms), active power (P),               *
* * apparent power (S), reactive power (Q), power factor (\(\lambda\)),
* * phase difference (\(\phi\)), voltage frequency (fU),              *
* * current frequency(fI)                                            *
* ***********************************************************************

OPEN "COM1:9600, N, 8, 1, ASC, CS0, DS0, LF" FOR RANDOM AS #1
OPTION BASE 1 ' Minimum value of array subscript=1
DIM D$(40) ' Array of numerical data strings
PRINT #1, "COMMUNICATE:REMOTE ON" ' Remote setting
' Set the measurement condition and range
PRINT #1, "SETUP:MODE NORMAL" ' Normal measurement mode
PRINT #1, "VOLTAGE:RANGE 200" ' Voltage range = 200 Vpk
PRINT #1, "CURRENT:RANGE 4" ' Current range = 4 Apk
PRINT #1, "FILTER:LINE OFF" ' Line filter=OFF
PRINT #1, "TIMEBASE:OBSERVE 100MS" ' Observation time = 100 msec
' Set the numerical data output items (ASCII format, preset to pattern 1, number of output data=40)
PRINT #1, "NUMERIC:FORMAT ASCII;NORMAL:PRESET 1;NUMBER 40"
' Set the transition filter used to detect the completion of the numerical data updating
PRINT #1, "STATUS:FILTER2 RISE" ' Rising edge of bit1 (DAV)
' Clear the extended event register (Read and trash the response)
PRINT #1, "STATUS:EESR?"
LINE INPUT #1, RG$ ' Measurement start
PRINT #1, "START"
' Read and display the numerical data (It is repeated 10 times in this program)
FOR I% = 1 TO 10
  PRINT #1, "COMMUNICATE:WAIT 2"
  PRINT #1, "STATUS:EESR?"
  LINE INPUT #1, RG$ ' Read out numerical data
  PRINT #1, "NUMERIC:NORMAL:VALUE?"
  ' Receive the items of numerical data that are separated by commas (,)
  FOR J% = 1 TO 40
    INPUT #1, D$(J%)
  NEXT J%
Display the numerical data

```
PRINT I%, "Element1", "Element2", "Element3", "Element4"
PRINT "Urms [V]", D$(1), D$(11), D$(21), D$(31)
PRINT "Irms [A]", D$(2), D$(12), D$(22), D$(32)
PRINT "P [W]", D$(3), D$(13), D$(23), D$(33)
PRINT "S [VA]", D$(4), D$(14), D$(24), D$(34)
PRINT "Q [var]", D$(5), D$(15), D$(25), D$(35)
PRINT "Lambda[]", D$(6), D$(16), D$(26), D$(36)
PRINT "Phi [\]]", D$(7), D$(17), D$(27), D$(37)
PRINT "fU [Hz]", D$(8), D$(18), D$(28), D$(38)
PRINT "fI [Hz]", D$(9), D$(19), D$(29), D$(39)
```

NEXT I%

Measurement stop
PRINT #1, "STOP"

PRINT #1, "COMMUNICATE:REMOTE OFF" ' Clear remote mode
CLOSE #1

END

Output example

<table>
<thead>
<tr>
<th>I</th>
<th>Element1</th>
<th>Element2</th>
<th>Element3</th>
<th>Element4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>102.44E+00</td>
<td>103.67E+00</td>
<td>104.32E+00</td>
<td>103.68E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.1224E+00</td>
<td>0.8108E+00</td>
<td>1.1202E+00</td>
<td>1.1052E+00</td>
</tr>
<tr>
<td>3</td>
<td>114.98E+00</td>
<td>84.06E+00</td>
<td>116.87E+00</td>
<td>114.59E+00</td>
</tr>
<tr>
<td>4</td>
<td>76.19E+00</td>
<td>63.06E+00</td>
<td>77.42E+00</td>
<td>75.93E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.7489E+00</td>
<td>0.6612E+00</td>
<td>0.7491E+00</td>
<td>0.7490E+00</td>
</tr>
<tr>
<td>6</td>
<td>41.50E+00</td>
<td>311.39E+00</td>
<td>41.49E+00</td>
<td>41.50E+00</td>
</tr>
<tr>
<td>7</td>
<td>49.975E+00</td>
<td>50.018E+00</td>
<td>49.985E+00</td>
<td>49.978E+00</td>
</tr>
</tbody>
</table>
In the harmonic measurement mode, set the measurement conditions and perform one measurement. The following numerical data regarding the current of element 1 are read and displayed:

PLL source frequency (the current frequency of element 1 in this program), total harmonic distortion (Ithd1), total rms value (I1(Total)), DC component (I1(dc)), fundamental signal (I1(1)), analyzed values from 2nd to 100th order (I1(2) to I1(100)).

```plaintext
REM $INCLUDE: 'qbdecl4.bas'

DIM D$(100) ' Array of numerical data strings

DEVICE$ = "DEV1": CALL IBFIND(DEVICE$, PZ%)
CALL IBSIC(PZ%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%) ' Remote setting

' Set the measurement conditions
CMD$ = "SETUP:MODE HARMONICS" ' Harmonic measurement mode
CALL IBWRT(PZ%, CMD$)
CMD$ = "SETUP:PLLSOURCE 2" ' PLL source = CH2(I1)
CALL IBWRT(PZ%, CMD$)
CMD$ = "MEASURE:HARMONICS:ORDER 0,100" ' Harmonic orders analyzed =0 to 100
CALL IBWRT(PZ%, CMD$)

' Set the numerical data output items
CMD$ = "NUMERIC:FORMAT ASCII" ' ASCII format
CALL IBWRT(PZ%, CMD$)
CMD$ = "NUMERIC:HARMONICS:PRESET 4" ' Settings to output fI, Ithd1
CALL IBWRT(PZ%, CMD$)
CMD$ = "NUMERIC:LIST:ITEM I,1;ORDER 100;SELECT ALL" ' Numerical data of I1 from Total to 100th order
CALL IBWRT(PZ%, CMD$)

' Set the transition filter used to detect the completion of the numerical data updating
CMD$ = "STATUS:FILTER2 RISE" ' Rising edge of bit1 (DAV)
CALL IBWRT(PZ%, CMD$)

' Clear the extended event register(Read and trash the response)
CMD$ = "STATUS:EESR?"
CALL IBWRT(PZ%, CMD$)
RG$ = SPACE$(8)
CALL IBRD(PZ%, RG$)

' Single measurement start
CMD$ = "SSTART"
CALL IBWRT(PZ%, CMD$)

' Wait for the completion of the numerical data updating
CMD$ = "COMMUNICATE:WAIT 2"
CALL IBWRT(PZ%, CMD$)

' Clear the extended event register(Read and trash the response)
CMD$ = "STATUS:EESR?"
CALL IBWRT(PZ%, CMD$)
RG$ = SPACE$(8)
CALL IBRD(PZ%, RG$)

' Read out the PLL source frequency (fI1)
CMD$ = "NUMERIC:HARMONICS:VALUE? 20" ' Pattern 4 ITEM20=fI1
CALL IBWRT(PZ%, CMD$)
RES$ = SPACE$(20)
CALL IBRD(PZ%, RES$)
PLL$ = LEFT$(RES$, (IBCNT% - 1))

' Read out the total harmonic distortion (Ithd1)
CMD$ = "NUMERIC:HARMONICS:VALUE? 27" ' Pattern 4 ITEM27=Ithd1
CALL IBWRT(PZ%, CMD$)
RES$ = SPACE$(20)
CALL IBRD(PZ%, RES$)
THD$ = LEFT$(RES$, (IBCNT% - 1))
```
Read out the harmonic numerical list data (I1(Total) to I1(100))

CMD$ = "NUMERIC:LIST:VALUE?" ' All 102 data
CALL IBWRT(PZ%, CMD$)
RES$ = SPACE$(1200)
CALL IBRD(PZ%, RES$)
C$ = LEFT$(RES$, IBCNT%)

' Extract items that are separated by commas (,) from the received numerical data
B% = INSTR$(C$, ",")' Total
TOTAL$ = LEFT$(C$, (B% - 1))
C$ = MID$(C$, (B% + 1))
FOR I% = 0 TO 100 ' 0(dc) to 100
  L% = LEN(C$)
  B% = INSTR$(C$, ",")
  IF B% = 0 THEN B% = L% + 1
  D$(I%) = LEFT$(C$, (B% - 1))
  C$ = MID$(C$, (B% + 1))
NEXT I%

' Display the numerical data
PRINT "Freq[Hz]", PLL$ ' PLL source frequency
PRINT "Ithd [%]", THD$ ' Total harmonic distortion
PRINT "Total[A]", TOTAL$ ' Total rms value
PRINT "dc [A]", D$(0) ' DC component
FOR I% = 1 TO 100 STEP 2
  PRINT "Or." + STR$(I%), D$(I%), ' Odd order components
  PRINT "Or." + STR$(I% + 1), D$(I% + 1) ' Even order components
NEXT I%
PRINT
V% = 0: CALL IBSRE(BD%, V%) ' Clear remote mode

END

Output example
Freq[Hz] 50.251E+00
Ithd [%] 70.31E+00
Total[A] 1.9485E+00
dc [A] 1.3834E+00
Or. 1 0.0772E+00 Or. 2 1.0735E+00
Or. 3 0.0621E+00 Or. 4 0.1141E+00
Or. 5 0.0742E+00 Or. 6 0.4238E+00
Or. 7 0.0726E+00 Or. 8 0.1510E+00
Or. 9 0.0892E+00 Or. 10 0.3400E+00
: : : :
Or. 99 0.0601E+00 Or. 100 0.0131E+0
6.4 Output Example of Waveform Data in ASCII Format

* PZ4000 Sample Program3 for GP-IB interface
  Microsoft QuickBASIC 4.0/4.5 Version
* Read the CH1(U1) waveform data from PZ4000 in ASCII format
* 

REM $INCLUDE: 'qbdecl4.bas'

DEVICES$ = "DEVI": CALL IBFIND(DEVICES$, PZ%)
CALL IBSCI(PZ%)
BORDS$ = "GP1BO": CALL IBFIND(BORDS$, BD%)
CALL IBSCI(BD%)
V% = 1: CALL IBRSR(BD$, V%)
  ' Set to remote

  ' Set conditions for reading the waveform
CMD$ = "WAVEFORM:TRACE 1;FORMAT ASCII" 
  ' Target waveform=CH1, ASCII format
CALL IBWRT(PZ%, CMD$)
  ' Query the total number of data points that can be read
CMD$ = "COMMUNICATE:HEADER OFF"
CALL IBWRT(PZ%, CMD$)
CMD$ = "WAVEFORM:LENGTH?"
CALL IBWRT(PZ%, CMD$)
LNS = SPACES(10)
CALL IBRD(PZ%, LNS)
B% = INSTR(LNS, CHR$(10))
L& = VAL(LEFT$(LNS, B% - 1))
  ' Read in the waveform data 10 data points at a time
IF L& = 0 THEN GOTO WAVEEXIT
WAV$ = SPACES(200)
CNK = 0
FOR I& = 0 TO (L& - 2) STEP 10
  CMD$ = "WAVEFORM:START" + STR$(I&) + ";END" + STR$(I& + 9) + ";SEND?"
  CALL IBWRT(PZ%, CMD$)
  CALL IBRD(PZ%, WAV$)
  K% = 1
  FOR J% = 0 TO 9
    IF J% < 9 THEN S% = INSTR(K%, WAV$, ",") ELSE S% = INSTR(K%, WAV$, CHR$(10))
    CNK = CNK + 1
    PRINT CNK, MID$(WAV$, K%, (S% - K%))
    K% = S% + 1
  NEXT J%
NEXT I&
WAVEEXIT:
V% = 0: CALL IBRSR(BD$, V%)
  ' Clear remote mode
END
OPEN "COM1:9600,N,8,1,ASC,CS0,DS0,LF" FOR RANDOM AS #1
PRINT #1, "COMMUNICATE:REMOTE ON" ' Set to remote
' Set conditions for reading the waveform
PRINT #1, "WAVEFORM:TRACE 1;FORMAT ASCII" ' Target waveform=CH1, ASCII format
' Query the total number of data points that can be read
PRINT #1, "COMMUNICATE:HEADER OFF"
PRINT #1, "WAVEFORM:LENGTH?"
LINE INPUT #1, LN$  
L& = VAL(LN$)
' Read in the waveform data
IF L& = 0 THEN GOTO WAVEEXIT
PRINT #1, "WAVEFORM:START 0" + ";END" + STR$(L& - 1) + ";SEND?"
FOR I& = 1 TO L&
  INPUT #1, WAV$
  PRINT I&, WAV$
NEXT I&
WAVEEXIT:
PRINT #1, "COMMUNICATE:REMOTE OFF" ' Clear remote mode
CLOSE #1
END

Output example
1 1.8311E+00
2 2.0752E+00
3 1.8311E+00
4 2.0752E+00
5 1.9531E+00
6 2.1973E+00
7 2.3193E+00
8 2.3193E+00
9 2.3193E+00
10 2.0752E+00
: : :
100000 -2.0752E+00
6.5 Output Example of Waveform Data in Binary Format

Read the CH1(U1) waveform data from PZ4000 in binary (WORD) format.

REM $INCLUDE: 'qbdecl4.bas'

DEVICE$ = "DEV1": CALL IBFIND(DEVICE$, PZ%)
CALL IBSIC(PZ%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%) ' Set to remote

'Set conditions for reading the waveform
CMDS = "WAVEFORM:TRACE 1;FORMAT BINARY;BYTEORDER LSBFIRST" ' Target waveform=CH1, WORD format
CALL IBWRT(PZ%, CMDS)

'Query the range value (needed to convert binary data to physical values)
CMDS = "COMMUNICATE:HEADER OFF"
CALL IBWRT(PZ%, CMDS)
CMDS = "WAVEFORM:RANGE?"
CALL IBWRT(PZ%, CMDS)
RNG$ = SPACE$(20)
CALL IBRD(PZ%, RNG$)
B% = INSTR(RNG$, CHR$(10))
R! = VAL(LEFT$(RNG$, B% - 1))

'Query the total number of data points that can be read
CMDS = "WAVEFORM:LENGTH?"
CALL IBWRT(PZ%, CMDS)
LN$ = SPACE$(10)
CALL IBRD(PZ%, LN$)
B% = INSTR(LN$, CHR$(10))
L& = VAL(LEFT$(LN$, B% - 1))

'Read in the waveform data 100 data points at a time
IF L& = 0 THEN GOTO WAVEEXIT
WAV$ = SPACE$(220)
CN& = 0
FOR I& = 0 TO (L& - 2) STEP 100
  CMDS = "WAVEFORM:START" + STR$(I&) + ";END" + STR$(I& + 99) + ";SEND?"
  CALL IBWRT(PZ%, CMDS)
  CALL IBRD(PZ%, WAV$)
  FOR J% = 0 TO 99
    CN& = CN& + 1
    PRINT CN&, (CVI(MID$(WAV$, J% * 2 + 11, 2)) - 2048) / 2048! * R!
  NEXT J%
NEXT I&
WAVEEXIT:
V% = 0: CALL IBSRE(BD%, V%) ' Clear remote mode

END

Output example
1 1.831055
2 2.075195
3 1.831055
4 2.075195
5 1.953125
6 2.197266
7 2.319336
8 2.319336
9 2.319336
10 2.075195
... ...
100000 -2.075195
Appendix 1  ASCII Character Code

ASCII character codes are given

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>120</td>
<td>140</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10</td>
<td>16</td>
<td>20</td>
<td>32</td>
<td>38</td>
<td>40</td>
<td>44</td>
</tr>
<tr>
<td>SOH</td>
<td>DC1</td>
<td>ETX</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>STX</td>
<td>DC2</td>
<td>ETX</td>
<td>D</td>
<td>C</td>
<td>E</td>
<td>F</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12</td>
<td>16</td>
<td>22</td>
<td>30</td>
<td>32</td>
<td>34</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>23</td>
<td>33</td>
<td>43</td>
<td>53</td>
<td>63</td>
<td>73</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>34</td>
<td>44</td>
<td>54</td>
<td>64</td>
<td>74</td>
<td>84</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>45</td>
<td>55</td>
<td>65</td>
<td>75</td>
<td>85</td>
<td>95</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>56</td>
<td>66</td>
<td>76</td>
<td>86</td>
<td>96</td>
<td>106</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>67</td>
<td>77</td>
<td>87</td>
<td>97</td>
<td>107</td>
<td>117</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>78</td>
<td>88</td>
<td>98</td>
<td>108</td>
<td>118</td>
<td>128</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>89</td>
<td>99</td>
<td>109</td>
<td>119</td>
<td>129</td>
<td>139</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>99</td>
<td>109</td>
<td>119</td>
<td>129</td>
<td>139</td>
<td>149</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>109</td>
<td>119</td>
<td>129</td>
<td>139</td>
<td>149</td>
<td>159</td>
<td>169</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>119</td>
<td>129</td>
<td>139</td>
<td>149</td>
<td>159</td>
<td>169</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>129</td>
<td>139</td>
<td>149</td>
<td>159</td>
<td>169</td>
<td>179</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>139</td>
<td>149</td>
<td>159</td>
<td>169</td>
<td>179</td>
<td>189</td>
<td>199</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>149</td>
<td>159</td>
<td>169</td>
<td>179</td>
<td>189</td>
<td>199</td>
<td>209</td>
</tr>
</tbody>
</table>

Example

- **octal**: 25  PPU  GP-IB code
- **hexadecimal**: 15  21  ASCII character code
- **Address Command**
- **Universal Command**
- **Listener Address**
- **Talker Address**
- **Secondary Command**
Appendix 2  Error Messages

Error messages related to communications are given below.

- The instrument allows error messages to be displayed in either Japanese or English, however, they are shown only in English when they are displayed on a personal computer.

- When servicing is required, contact your nearest YOKOGAWA representative, given on the back cover of this manual.

- Only error messages relating to communications are given. For other error messages, refer to the User’s Manual IM 253710-01E.

### Errors in communication command (100 to 199)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>Syntax error</td>
<td>Incorrect syntax.</td>
<td>Chapter 3, 4</td>
</tr>
<tr>
<td>103</td>
<td>Invalid separator</td>
<td>Insert a comma between data items to separate them.</td>
<td>3-1</td>
</tr>
<tr>
<td>104</td>
<td>Data type error</td>
<td>Refer to pages 3-5 to 3-6 and enter using the correct data format.</td>
<td>3-5 to 3-6</td>
</tr>
<tr>
<td>108</td>
<td>Parameter not allowed</td>
<td>Check the number of parameters.</td>
<td>3-5, Chapter 4</td>
</tr>
<tr>
<td>109</td>
<td>Missing parameter</td>
<td>Enter required parameters.</td>
<td>3-5, Chapter 4</td>
</tr>
<tr>
<td>111</td>
<td>Header separator error</td>
<td>Insert a space between header and data to separate them.</td>
<td>3-1</td>
</tr>
<tr>
<td>112</td>
<td>Program mnemonic too long</td>
<td>Check the mnemonic (a character string consisting of letters and numbers).</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>113</td>
<td>Undefined header</td>
<td>Check the header.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>114</td>
<td>Header suffix out of range</td>
<td>Check the header.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>120</td>
<td>Numeric data error</td>
<td>Numeric value must be entered for &lt;NR&gt; format.</td>
<td>3-5</td>
</tr>
<tr>
<td>123</td>
<td>Exponent too large</td>
<td>Use a smaller exponent for &lt;NR&gt; format.</td>
<td>3-5, Chapter 4</td>
</tr>
<tr>
<td>124</td>
<td>Too many digits</td>
<td>Limit the number of digits to 255 or less.</td>
<td>3-5, Chapter 4</td>
</tr>
<tr>
<td>128</td>
<td>Numeric data not allowed</td>
<td>Enter in a format other than &lt;NR&gt; format.</td>
<td>3-5, Chapter 4</td>
</tr>
<tr>
<td>131</td>
<td>Invalid suffix</td>
<td>Check the unit for &lt;Voltage&gt;, &lt;Time&gt; and &lt;Frequency&gt;.</td>
<td>3-5</td>
</tr>
<tr>
<td>134</td>
<td>Suffix too long</td>
<td>Check the units for &lt;Voltage&gt;, &lt;Time&gt; and &lt;Frequency&gt;.</td>
<td>3-5</td>
</tr>
<tr>
<td>138</td>
<td>Suffix not allowed</td>
<td>No units are allowed other than &lt;Voltage&gt;, &lt;Time&gt; and &lt;Frequency&gt;.</td>
<td>3-5</td>
</tr>
<tr>
<td>141</td>
<td>Invalid character data</td>
<td>Enter one of the character strings in {...</td>
<td>...</td>
</tr>
<tr>
<td>144</td>
<td>Character data too long</td>
<td>Check the character strings in {...</td>
<td>...</td>
</tr>
<tr>
<td>148</td>
<td>Character data not allowed</td>
<td>Enter in a format other than in {...</td>
<td>...</td>
</tr>
<tr>
<td>150</td>
<td>String data error</td>
<td>&lt;Character string&gt; must be enclosed by double quotation marks or single quotation marks.</td>
<td>3-6</td>
</tr>
<tr>
<td>151</td>
<td>Invalid string data</td>
<td>&lt;Character string&gt; is too long or contains characters which cannot be used.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>158</td>
<td>String data not allowed</td>
<td>Enter in a data format other than &lt;Character string&gt;.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>161</td>
<td>Invalid block data</td>
<td>&lt;Block data&gt; is not allowed.</td>
<td>3-6, Chapter 4</td>
</tr>
<tr>
<td>168</td>
<td>Block data not allowed</td>
<td>&lt;Block data&gt; is not allowed.</td>
<td>3-6, Chapter 4</td>
</tr>
<tr>
<td>171</td>
<td>Invalid expression</td>
<td>Equation is not allowed.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>178</td>
<td>Expression data not allowed</td>
<td>Equation is not allowed.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>181</td>
<td>Invalid outside macro definition</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td></td>
</tr>
</tbody>
</table>
## Error in communications execution (200 to 299)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>221</td>
<td>Setting conflict</td>
<td>Check the relevant setting.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>222</td>
<td>Data out of range</td>
<td>Check the setting range.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>223</td>
<td>Too much data</td>
<td>Check the data byte length.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>224</td>
<td>Illegal parameter value</td>
<td>Check the setting range.</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>241</td>
<td>Hardware missing</td>
<td>Check availability of options.</td>
<td>—</td>
</tr>
<tr>
<td>260</td>
<td>Expression error</td>
<td>Equation is not allowed.</td>
<td>—</td>
</tr>
<tr>
<td>270</td>
<td>Macro error</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>272</td>
<td>Macro execution error</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>273</td>
<td>Illegal macro label</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>275</td>
<td>Macro definition too long</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>276</td>
<td>Macro recursion error</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>277</td>
<td>Macro redefinition not allowed</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
<tr>
<td>278</td>
<td>Macro header not allowed</td>
<td>Does not conform to the macro function specified in IEEE488.2.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Error in communications Query (400 to 499)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>410</td>
<td>Query INTERRUPTED</td>
<td>Check transmission/reception order.</td>
<td>3-2</td>
</tr>
<tr>
<td>420</td>
<td>Query UNTERMINATED</td>
<td>Check transmission/reception order.</td>
<td>3-2</td>
</tr>
<tr>
<td>430</td>
<td>Query DEADLOCKED</td>
<td>Limit the length of the program message including &lt;PMT&gt; to 1024 bytes or less.</td>
<td>3-2</td>
</tr>
<tr>
<td>440</td>
<td>Query UNTERMINATED after indefinite response</td>
<td>Do not enter any query after *IDN? and *OPT?.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Error in System Operation (912 to 914)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>912</td>
<td>Fatal error in Communications-driver</td>
<td>Servicing is required.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Warning

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>*OPC/? exists in message</td>
<td>Place the *OPC or *OPC? at the end of the program message.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Other errors (350 and 390)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Action</th>
<th>Reference Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>350</td>
<td>Queue overflow</td>
<td>Read the error queue. Code 350 occurs when the error queue is full up. This message is output only for the STATus:ERRor? query and is not displayed on the screen.</td>
<td>5-5</td>
</tr>
<tr>
<td>390</td>
<td>Overrun error (only Serial(RS-232))</td>
<td>Execute with a lower baud rate.</td>
<td>—</td>
</tr>
</tbody>
</table>

**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  Overview of IEEE 488.2-1987

The GP-IB interface provided with PZ4000 conforms to IEEE 488.2-1987. This standard requires the following 23 points be stated in this document. This Appendix describes these points.

1  Subsets supported by IEEE 488.1 interface functions
   Refer to Section 1.4 “GP-IB Interface Specifications”.

2  Operation of device when the device is assigned to an address other than addresses 0 to 30.
   The PZ4000 does not allow assignment to an address other than 0 to 30.

3  Reaction when the user changes the address
   The current address is changed when a new address is set using the MISC key. The newly set address is valid until another new address is set.

4  Device set-up at power ON. Commands which can be used at power ON
   Basically, the previous settings (i.e. the settings which were valid when power was turned OFF) are valid. All commands are available at power ON.

5  Message transmission options
   a  Input buffer size
      1024 bytes
   b  Queries which return multiple response messages
      Refer to Chapter 4, “Command List”.
   c  Queries which generate response data during analysis of the syntax
      Every query generates a response data when analysis of the syntax is completed.
   d  Queries which generate response data during reception
      No query generates response data when the query is received by the controller.
   e  Commands consisting of parameters which restrict one other
      Refer to Chapter 4, “Command List”.

6  Options included in command function elements and composite header elements
   Refer to Chapters 3 and 4.

7  Buffer size which affects transmission of block data
   During transmission of block data, the output queue is extended according to the size of the data blocks.

8  List of program data elements which can be used in equations, and nesting limit
   No equations can be used.

9  Syntax of response to queries
   Refer to the description of the commands given in Chapter 4.

10 Communications between devices which do not follow the response syntax
    No communications between devices.
11 Size of data block of response data
   1 to 16000004 (4000001×4) bytes

12 List of supported common commands
   Refer to Section 4.26 “Common Command Group”.

13 Condition of device when calibration is successfully completed
   Same as the one under which measurements are performed

14 Maximum length of block data which can be used for definition of *DDT trigger macro
   Not supported

15 Maximum length of macro label used in definition of macro, maximum length of block data which can be used for
   definition of macro, processing when recursion is used in definition of macro
   Macro functions are not supported.

16 Response to *IDN?
   Refer to Section 4.30 “Common Command Group”.

17 Size of storage area for protected user data for PUD and *PUD?
   *PUD and *PUD? are not supported.

18 Length of *RDT and *RDT? resource name
   *RDT and *RDT? are not supported.

19 Change in status due to *RST, *LRN?, *RCL and *SAV
   *RST
   Refer to Section 4.26 “Common Command Group”.
   *LRN?, *RCL, *SAV
   These commands are not supported.

20 Execution range of self-test using the *TST?
   All the memory tests (for each internal memory) given in the Self Test menu displayed
   using the MISC key can be executed.

21 Structure of extended return status
   Refer to Chapter 5.

22 To find out whether each command is performed in parallel or sequentially
   Refer to Section 3.5 “Synchronization with the Controller” and to Chapter 4.

23 Description of execution of each command
   Refer to Chapter 4 of this manual and to the User’s Manual IM 253710-01E.
Index

A
ABORT Group ................................................................. 4-11
ACQuire Group ............................................................ 4-11
Abbreviated form ......................................................... 3-5
Address commands ...................................................... 1-7
address ........................................................................ 1-5
apparent power ............................................................ 4-60
averaging ..................................................................... 4-57

B
BMP format ..................................................................... 4-39
Baud rate ....................................................................... 2-8
Bit Masking .................................................................. 5-2, 5-3
Block data ...................................................................... 3-7
Boolean .......................................................................... 3-6
bar graph display .......................................................... 4-25
brightness (LCD monitor) .............................................. 4-74

current input terminals .................................................. 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

D
DISPlay Group ............................................................ 4-23
Data ............................................................................... 3-5
Deadlock ....................................................................... 3-2
Decimal ......................................................................... 3-5
data compressing ....................................................... 4-39
data format .................................................................... 2-7, 2-8
data output ..................................................................... 4-38
date ............................................................................... 4-26, 4-74
delta computation ......................................................... 4-58
directory ........................................................................ 4-35
display .......................................................................... 4-25
display color (LCD monitor) .......................................... 4-74
display color (graphic item) ........................................... 4-74
display color (others) ................................................... 4-74
display color (text item) ................................................ 4-74
display color (text, others) ............................................ 4-75
display format .............................................................. 4-26
display format (waveform) ........................................... 4-29
display format of the zoomed waveform ....................... 4-84
displayed digit .............................................................. 4-69
drive .............................................................................. 4-34

current range .................................................................. 4-48
current input terminals ................................................ 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

current input terminals .................................................. 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

C
CHANnel Group ............................................................ 4-12
COMMunicate Group .................................................. 4-16
CT ratio ......................................................................... 4-50
CURSor Group ............................................................ 4-18
Character Data .............................................................. 3-6
Character String Data ................................................... 3-6
Command Group .......................................................... 3-3
Commands ....................................................................... 3-3
Common Command Group .......................................... 4-85
Common Command Header .......................................... 3-3
Compound Command Header ....................................... 3-3
calibration ................................................................. 4-85
center level ................................................................... 4-79
channel waveform display .......................................... 4-29
color (TIFF/BMP format) ............................................. 4-39
color (external printer) .................................................. 4-38
color (image data) ....................................................... 4-40
command format ......................................................... 4-38
command list ................................................................ 4-1
comment ....................................................................... 4-35
computation .............................................................. 4-53
computation period ..................................................... 4-59
computed waveform display ........................................ 4-29
condition register ......................................................... 5-4
connection example (serial) ........................................ 2-4
converting the scale .................................................... 4-54
corrected power .......................................................... 4-59
current auto range ...................................................... 4-48
current directory .......................................................... 4-34
current input channel ................................................. 4-13

current range .................................................................. 4-48
current input terminals ................................................ 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

current range .................................................................. 4-48
current input terminals ................................................ 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

current range .................................................................. 4-48
current input terminals ................................................ 4-49
current measurement .................................................. 4-49
current range ............................................................... 4-47
cursor (measurement) ................................................... 4-19
cursor type .................................................................... 4-21

E
Error Messages .......................................................... App-2
Error Queue ................................................................... 5-5
Extended Event Register ............................................ 5-4
edge trigger ............................................................... 4-78
end point ....................................................................... 4-54
error ........................................................................... 4-72
error queue ............................................................... 4-72, 4-85
extended event enable register .................................... 4-71
extended event register ................................................ 4-71, 4-85
external printer output ................................................ 4-38

F
FFT .............................................................................. 4-54
FILE Group ............................................................... 4-33
Filename ........................................................................ 3-7
Front Panel ................................................................. 1-1, 2-1
feeding (paper) ............................................................ 4-38
file name ...................................................................... 4-35, 4-38
file operation ............................................................. 4-34
Index

filter ................................................................. 4-49
floppy disk ........................................................ 4-34
format (floppy disk) ........................................... 4-34
free space ......................................................... 4-34

G
GP-IB Interface Functions ........................................ 1-3
GP-IB Interface Specifications .............................. 1-4
GP-IB connector ..................................................... 1-2
graticule type (grid) .............................................. 4-29
grid (graticule type) .............................................. 4-29

H
H cursor ............................................................... 4-19
HCOPY Group ......................................................... 4-37
Handshaking ......................................................... 2-5
Handshaking method ............................................. 2-8
harmonic measurement ......................................... 4-58
horizontal axis (time base) .................................... 4-76

I
IEEE 488.2-1987 ................................................... App-4
IMAGE Group ......................................................... 4-40
INPut Group .......................................................... 4-41
Initialize (setting) .................................................. 4-69
image data output .................................................. 4-40
input module ......................................................... 4-44
instrument model .................................................. 4-86
interface message ................................................ 1-6
interpolation method ............................................. 4-29
Initialize (setting) .................................................. 4-86

L
LCD monitor .......................................................... 4-74
Language ............................................................. 6-1
line filter ............................................................... 4-50
loading (abortion) .................................................. 4-34
loading (setup parameter) ..................................... 4-35
loading (waveform data) ....................................... 4-35
local lockout ......................................................... 4-16
lower limit ............................................................ 4-55
marker/cursor type ................................................. 4-21
measurement mode ............................................. 4-69
message language .............................................. 4-74
model name ......................................................... 4-44
mapping method .................................................. 4-30
marker (FFT) ....................................................... 4-20

N
NULL Group ........................................................... 4-61
NULL function ....................................................... 4-61
NUMeric Group ..................................................... 4-62
numerical data ..................................................... 4-63
numerical data file ............................................... 4-34
numerical display ............................................... 4-26
numerical display (harmonic measurement) .......... 4-26
numerical display (normal measurement) .............. 4-27

O
Output Queue ....................................................... 5-5
Overlap Commands ............................................ 3-7
observation time ................................................ 4-76
option ................................................................. 4-86
output data format .............................................. 4-38
output format (image data) ................................... 4-40

P
PLL source ........................................................... 4-69
PT ratio ............................................................... 4-50
Pc (Corrected Power) ............................................ 4-59
Program Messages ............................................... 3-1
Program data ......................................................... 3-1
Program header .................................................... 3-1
Program message unit .......................................... 3-1
phase difference .................................................. 4-60
power coefficient ............................................... 4-50
power measurement module .................................. 4-47
preset (normal measurement mode) ...................... 4-65
preset (harmonics measurement) ......................... 4-64

Q
queue ................................................................. 5-2

R
Rear Panel ........................................................... 1-1, 2-1
Register ............................................................. 3-6
Response Messages ............................................. 3-1
Response data ..................................................... 3-2
Response header .................................................. 3-2
record length ...................................................... 4-11
register .............................................................. 4-86, 5-2
response message ................................................................. 3-5
revolution sensor signal ..................................................... 4-14

S

SCSI-ID ................................................................. 4-75
SETup Group ............................................................... 4-69
SSStart Group ............................................................. 4-70
START Group ............................................................. 4-70
STATus Group ............................................................. 4-71
STOP Group ............................................................... 4-72
SYSTem Group ........................................................... 4-73
Sequential Commands ..................................................... 3-7
Serial Interface Specifications ............................................. 2-2
Simple Header ............................................................ 3-3
Standard Event Register .................................................. 5-3
Status Byte ............................................................... 5-2
Status Report .............................................................. 5-1
Synchronization with the Controller ................................. 3-7
sampling rate ............................................................... 4-76, 4-81
saving ............................................................... 4-35
saving (abortion) .......................................................... 4-35
saving (file) ................................................................. 4-38
saving (numerical data) ................................................... 4-35
saving (waveform data) .................................................. 4-36
scale value display ....................................................... 4-30
scaling ............................................................... 4-50
screen data output ........................................................ 4-38
screen display ............................................................ 4-25
screen image data file .................................................... 4-34
self-test ............................................................... 4-87
serial poll ................................................................. 4-72
service request enable register ....................................... 4-87
setup parameter file ...................................................... 4-34
single start ................................................................. 4-70, 4-87
standard event enable register ........................................ 4-85
standard event register .................................................. 4-85, 4-86
start acquisition ........................................................... 4-70
start point ................................................................. 4-54
status (line-specific) .................................................... 4-17
status byte register ....................................................... 4-87
status function ........................................................... 4-71
status register ............................................................ 4-71
stop acquisition .......................................................... 4-72
synchronizing source .................................................... 4-60
system ................................................................. 4-74

T

THD (total harmonic distortion) ......................................... 4-58
TIFF format ............................................................... 4-39
TIMebase Group ........................................................ 4-76
TRIGger Group .......................................................... 4-77
Terminator ............................................................... 2-8
time ................................................................. 4-26, 4-75
total harmonic distortion ................................................ 4-58
total number of data points ............................................. 4-81
transformation ratio ..................................................... 4-49
transition filter ........................................................... 4-72
trigger ................................................................. 4-78
trigger condition ........................................................ 4-79
trigger delay ............................................................. 4-78
trigger level .............................................................. 4-78
trigger mode ............................................................. 4-79
trigger position ......................................................... 4-78, 4-82
trigger slope ............................................................. 4-78
trigger source .......................................................... 4-79
trigger type ............................................................. 4-79
torque meter ............................................................ 4-15
type of the revolution sensor .......................................... 4-15

U

Unit ................................................................. 3-6
Universal commands ..................................................... 1-7
Upper-level Query ....................................................... 3-4
unit (math) ............................................................... 4-55
upper limit ............................................................... 4-55
user-defined function .................................................. 4-58

V

V cursor ................................................................. 4-21
vector display ........................................................... 4-28
vertical axis ............................................................... 4-13
voltage auto range ...................................................... 4-51
voltage input channel .................................................. 4-15
voltage measurement ................................................... 4-51
voltage range ........................................................... 4-51
vertical position ......................................................... 4-14

W

WAVeform Group ....................................................... 4-80
waveform data .......................................................... 4-80
waveform data file ...................................................... 4-34
waveform display ....................................................... 4-29
waveform label .......................................................... 4-14
waveform label display ................................................ 4-30
window function ......................................................... 4-54
window trigger .......................................................... 4-79
window width ........................................................... 4-79
wiring method ........................................................... 4-70
waveform mapping ...................................................... 4-29

X

X-axis value ............................................................. 4-22
X-Y display .............................................................. 4-30
XY cursor ............................................................... 4-22
Index

Y

Y-axis value (H cursor) ...................................................... 4-19

Z

ZOOM Group ................................................................. 4-83
zero crossing filter .......................................................... 4-50
zero level compensation .................................................. 4-85
zoom box ................................................................. 4-84
zoom factor ............................................................. 4-15, 4-84
zooming ................................................................. 4-83
zoom factor of the current (vector) ................................. 4-28
zoom factor of the voltage (vector) ................................. 4-29