KIKUSUI

Part No. IB035122
Feb 2020

Communication Interface Manual

Electronic Load
PLZ-5WH2 Series

PLZ1005WH2
PLZ2005WH2
PLZ4005WH2
PLZ12005WH2
PLZ20005WH2

Contents

Command List

IEEE 488.2 common commands

*CLS

Clears all event registers including the status byte, event status, and error queue.

*ESE

Sets the event status enable register that is counted by the event summary bit (ESB) of the status byte.

*ESR

Queries the event status register.

*IDN

Queries the model name and firmware version of the product.

*OPC

Sets the OPC bit (bit 0) of the event status register when all the commands that are in standby have been processed.

*OPT

Queries the options that are installed in the product.

*PSC

Sets whether the event status enable register and service request enable register are cleared when the POWER switch is turned on.

*RCL

Recalls setup memories.

*RST

Resets the panel settings.

*SAV

Saves the panel settings to the setup memory.

*SRE

Sets the service request enable register.

*STB

Queries the contents of the status byte register and the MSS (master summary status) message.

*TRG

Trigger command.

*TST

Executes a self-test.

*WAI

Prevents the device from executing subsequent commands until all operations that are in standby have completed.

ABORt Command

ABOR

Aborts measurements, tests, and other operations in all trigger subsystems (ACQuire, TRANsient, PULSe, SINE).

ABOR:ACQ

Aborts measurement operations.

ABOR:PULS

Stops the pulse operation.

ABOR:SINE

Stops the sine operation.

ABOR:TRAN

Stops the sequence operation.

DATA Command

DATA:BSIZ

Sets the buffer size (maximum number of measurement points) of the data logger function.

DATA:FORM

Sets the response format (type of measured data and order) to be used when referencing data in the data logger.

DATA:INT:GATE

Sets the integration time.

DATA:INT:GATE:ARES

Sets whether past integration is reset at the start of the integration operation.

DATA:INT:RES

Resets integrated data (current capacity (Ah), power capacity (Wh), elapsed time (s)).

DATA:INT:STAR

Manually starts the integration operation.

DATA:INT:STOP

Manually stops the integration operation.

DATA:POIN

Queries the number of measurement points registered in the data logger.

DATA:R

Queries the measurement values registered in the data logger from oldest to youngest.

DISPlay Command

DISP:CAP

Sets whether to display the current capacity (Ah) on the display.

DISP:ENER

Sets whether to display the power capacity (Wh) on the display.

DISP:ETIM

Sets whether to display the elapsed time on the display.

HCOPy Command

HCOP:SDUM:DATA

Retrieves the screen capture of the present screen.

INITiate Command

INIT:ACQ

Starts the measurement trigger function.

INIT:PULS

Enables the pulse function.

INIT:SINE

Enables the sine function.

INIT:TRAN:PROG

Starts the sequence start trigger function in the program selected by PROG.

INPut/ OUTPut Command

INP/ OUTP

Sets the load on/off.

INP:CUTO:CAP/ OUTP:CUTO:CAP

Sets the integrated current of the cutoff function.

INP:CUTO:CAP:STAT/ OUTP:CUTO:CAP:STAT

Enables or disables the cutoff function for the integrated current.

INP:CUTO:ENER/ OUTP:CUTO:ENER

Sets the integrated power of the cutoff function.

INP:CUTO:ENER:STAT/ OUTP:CUTO:ENER:STAT

Enables or disables the cutoff function for the integrated power.

INP:CUTO:ETIM/ OUTP:CUTO:ETIM

Sets the elapsed time of the cutoff function.

INP:CUTO:ETIM:STAT/ OUTP:CUTO:ETIM:STAT

Enables or disables the cutoff function for the elapsed time.

INP:CUTO:VOLT/ OUTP:CUTO:VOLT

Sets the voltage drop of the cutoff function.

INP:CUTO:VOLT:STAT/ OUTP:CUTO:VOLT:STAT

Enables or disables the cutoff function for the voltage drop.

INP:EXT:LOG/ OUTP:EXT:LOG

Sets the signal logic for external control input.

INP:MSYN/ OUTP:MSYN

Sends a load-on or load-off sync signal.

INP:MSYN:ACC/ OUTP:MSYN:ACC

Sets whether to accept load on/off sync signal (INP:MSYN) input from other synchronized devices.

INP:PON:STAT/ OUTP:PON:STAT

Sets the condition panel setting state when the POWER switch is turned on.

INP:PROT:CLE/ OUTP:PROT:CLE

Clears the protection mode.

INP:PROT:WDOG/ OUTP:PROT:WDOG

Enables/disables the watchdog protection function.

INP:PROT:WDOG:DEL/ OUTP:PROT:WDOG:DEL

Sets the delay time of the watchdog protection function.

MEASure/READ/FETCh Command

MEAS/ READ/ FETC

Queries the measurement results.

MEAS:CAP/ READ:CAP/ FETC:CAP

Queries the measured current capacity (Ah).

MEAS:CURR/ READ:CURR/ FETC:CURR

Queries the measured current (A).

MEAS:ENER/ READ:ENER/ FETC:ENER

Queries the measured power capacity (Wh).

MEAS:ETIM/ READ:ETIM/ FETC:ETIM

Queries the measured elapsed time (s).

MEAS:POW/ READ:POW/ FETC:POW

Queries the measured power (W).

MEAS:VOLT/ READ:VOLT/ FETC:VOLT

Queries the measured voltage (V).

MEMory Command

MEM:REC

Recalls settings from preset memories.

MEM:SAVE

Saves the settings to ABC preset memories.

SENSe Command

SENS:APER

Sets the recording time period per measurement. The average over the time period is recorded.

SOURce:ARBitrary Command

ARB:APPL

Enables the specified I-V characteristics.

ARB:CLE

Returns the I-V characteristics to the default setting.

ARB:COUN

Sets the number of data points of the I-V characteristics.

ARB:DATA

Sets the entire I-V characteristics in binary block data format.

ARB:MAP

Sets the voltage and current values for the specified parts (point index) of the I-V characteristics.

ARB:MAP:LIST

Queries the entire contents of the I-V characteristics.

ARB:RESP

Sets the response speed for ARB mode.

SOURce:CONDuctance Command

COND

Sets the conductance in CR mode.

COND:EXT:FCON

Enables/disables external control of CR mode.

COND:PULS:LEV

Sets the conductance of the pulse function.

COND:PULS:FREQ

Sets the frequency of the pulse function.

COND:PULS:DCYC

Sets the duty cycle of the pulse function.

COND:RANG

Set the range of CR mode.

COND:RESP

Sets the response speed for CR mode.

SOURce:CURRent Command

CURR

Sets the current in CC mode.

CURR:EXT:ACON

Enables or disables the function that superimposes current on the current in CC mode using an external voltage.

CURR:EXT:FCON

Enables/disables external control of CC mode.

CURR:PROT

Sets the current value for overcurrent protection (OCP).

CURR:PROT:STAT

Sets the operation when overcurrent protection (OCP) is activated.

CURR:PULS:LEV

Sets the current of the pulse function.

CURR:PULS:FREQ

Sets the frequency of the pulse function.

CURR:PULS:DCYC

Sets the duty cycle of the pulse function.

CURR:SINE:AMPL

Sets the amplitude of the sine function.

CURR:SINE:FREQ

Sets the frequency of the sine function.

CURR:SLEW

Set the slew rate value.

CURR:SST

Sets the soft start time.

SOURce:FUNCtion Command

FUNC

Set the operation mode.

FUNC:CVOP

Enables/disables +CV mode (adds CV mode to CC mode or CR mode).

SOURce:POWer Command

POW

Sets the power in CP mode.

POW:EXT:FCON

Enables/disables external control of CP mode.

POW:PROT

Sets the power of overpower protection (OPP).

POW:PROT:STAT

Sets the operation when overpower protection (OPP) is activated.

SOURce:PROGram Command

PROG

Selects or deselects a program.

PROG:CRE

Creates a new program.

PROG:CURR:PROT

Sets the current value for overcurrent protection (OCP) in the program selected by PROG.

PROG:CURR:PROT:ACT

Sets the operation to be performed when overcurrent protection (OCP) is activated in the program selected by PROG.

PROG:CURR:PROT:STAT

Sets the operation to be performed when overcurrent protection (OCP) is activated in the program selected by PROG.

PROG:CVOP:LEV

Sets the +CV (addition of CV mode) voltage in the program selected by PROG.

PROG:DEL

Deletes the program of the specified name.

PROG:LIST

Queries the list of all the registered programs.

PROG:LOOP

Sets the number of times that the program selected by PROG will repeat.

PROG:POW:PROT

Sets the power value for overpower protection (OPP) in the program selected by PROG.

PROG:POW:PROT:ACT

Sets the operation to be performed when overpower protection (OPP) is activated in the program selected by PROG.

PROG:POW:PROT:STAT

Sets the operation to be performed when overpower protection (OPP) is activated in the program selected by PROG.

PROG:LAST:INP/ PROG:LAST:OUTP

Sets the load state (load on or load off) at the end of the sequence for the program selected by PROG.

PROG:LAST:LEV

Sets the load value at the end of the sequence for the program selected by PROG.

PROG:LAST:SLEW

Sets the slew rate at the end of the sequence for the program selected by PROG.

PROG:LAST:STAT

Enables or disables the following command that sets the load state at the end of the sequence for the program selected by PROG.

PROG:REN

Changes the name of the program selected by PROG.

PROG:SAVE

Saves the program selected by PROG.

PROG:STEP<n>:DWEL

Sets the step execution time at the specified step of the program selected by PROG.

PROG:STEP<n>:INP/ PROG:STEP<n>:OUTP

Sets load on/off at the specified step of the program selected by PROG.

PROG:STEP<n>:LEV

Sets the load value at the specified step of the program selected by PROG.

PROG:STEP<n>:SLEW

Sets the slew rate at the specified step of the program selected by PROG.

PROG:STEP<n>:TRAN

Sets the setting transition method at the specified step of the program selected by PROG.

PROG:STEP<n>:TRIG:GEN

Sets whether to perform trigger output at the specified step of the program selected by PROG.

PROG:STEP<n>:TRIG:WAIT

Sets trigger wait at the specified step of the program selected by PROG.

PROG:STEPS:COUN

Sets the number of steps to be registered to the program selected by PROG.

PROG:VOLT:PROT:ACT

Sets the action to be performed when undervoltage protection (UVP) is activated in the program selected by PROG.

PROG:VOLT:PROT:LOW

Sets the voltage for undervoltage protection (UVP) in the program selected by PROG.

PROG:VOLT:PROT:STAT

Enables or disables the undervoltage protection (UVP) in the program selected by PROG.

SOURce:VOLTage Command

VOLT

Sets the voltage in CV mode.

VOLT:EXT:FCON

Enables/disables external control of CV mode.

VOLT:PROT:ACT

Sets the action to be performed when undervoltage protection (UVP) is activated.

VOLT:PROT:LOW

Sets the voltage of undervoltage protection (UVP).

VOLT:PROT:STAT

Enables/disables undervoltage protection (UVP).

VOLT:RESP

Sets the response speed for CV mode.

TRIGger Command

TRIG:ACQ:COUN

Sets the number of times measurement values are to be recorded.

TRIG:ACQ:DEL

Sets the delay time for applying measurement triggers.

TRIG:ACQ:MSYN

Sends a measurement sync signal.

TRIG:ACQ:SOUR

Clears the measurement trigger weight state and sets the trigger source for starting measurements.

TRIG:ACQ:INT:STAT

Sets whether to measure at intervals when the trigger count is 2 or higher.

TRIG:ACQ:INT:TIME

Sets the measurement interval time when TRIG:ACQ:INT:STAT is set to ON.

TRIG:TRAN:DEL

Sets the delay time for applying sequence triggers.

TRIG:TRAN:EXEC

Queries the execution state of the sequence function, pulse function, and sine function.

TRIG:TRAN:MSYN

Sends a sequence sync signal.

TRIG:TRAN:SOUR

Clears the sequence trigger weight state and sets the trigger source for starting sequences.

TRIG:TRAN:SUSP

Pauses the running sequence trigger function.

TRIG:TRAN:RES

Resumes the paused sequence from the point at which it was paused.

STATus Command

STAT:OPER

Queries the event of the OPERation status register.

STAT:OPER:COND

Queries the condition of the OPERation status register.

STAT:OPER:ENAB

Sets the enable register of the OPERation status register.

STAT:OPER:NTR

Sets the negative transition filter of the OPERation status register.

STAT:OPER:PTR

Sets the positive transition filter of the OPERation status register.

STAT:QUES

Queries the event of the QUEStionable status register.

STAT:QUES:COND

Queries the condition of the QUEStionable status register.

STAT:QUES:ENAB

Sets the enable register of the QUEStionable status register.

STAT:QUES:NTR

Sets the negative transition filter of the QUEStionable status register.

STAT:QUES:PTR

Sets the positive transition filter of the QUEStionable status register.

STAT:PRES

Resets the ENABle, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values.

SYSTem Command

SYST:BEEP

Turns all buzzers on and off collectively.

SYST:BEEP:KEY

Sets the buzzer sound on/off in case of invalid operation.

SYST:BEEP:PROT

Turns on or off the buzzer that sounds when a protection function is activated.

SYST:BEEP:SCPI

Turns on or off the buzzer that sounds when an SCPI error occurs.

SYST:COMM:RLST

Switches the PLZ-5WH2 to local or remote mode.

SYST:CONF:DIGITAL2:DIR

Sets the I/O direction of the DIGITAL2 signal.

SYST:CONF:MSYN:IND

Queries whether synchronization connection is enabled or disabled.

SYST:CONF:RSEN

Sets remote sensing function on/off.

SYST:DATE

Sets the date.

SYST:ERR

Reads the oldest error information or event information from the error queue.

SYST:ERR:COUN

Returns the number of unread errors in the error queue.

SYST:KLOC

Sets or releases panel control lock.

SYST:KLOC:LEV

Sets the panel control lock level.

SYST:PASS

Enables the execution of the command that returns the product to its factory default settings.

SYST:PASS:CDIS

Disables the execution of the command that returns the settings back to factory default.

SYST:PASS:NEW

Sets the password.

SYST:PASS:STAT

Queries whether the product is in a state that allows the execution of the command that returns the product to its factory default settings.

SYST:SEC:IMM

Deletes all the user data other than the calibration data and present time and returns the product to its factory default settings.

SYST:SSAV

Enables or disables the screen saver.

SYST:SSAV:DEL

Sets the time until the screen saver starts.

SYST:TIME

Sets the time.

SYST:TIME:ADJ

Automatically synchronizes the system clock using the NTP server on the network.

SYST:TZON

Sets the time zone of the system clock.

SYST:TZON:CAT

Queries the time zone IDs that can be used.

SYST:VERS

Queries the version of the SCPI specifications that the product complies with.

Introduction

The PLZ-5WH2 Series Communication Interface Manual explains the settings that are used to control the PLZ-5WH2 series remotely through the following interfaces and the available commands.

- RS232C interface
- USB interface
- LAN interface

When the product is operating under remote control, REMOTE appears on the front panel display. To switch the product back to local mode from the front panel, press LOCAL.

For the safety precautions, installation, operation, and specifications of the product, read the accompanying PLZ-5WH2 Series User's Manual.

■ Reading environment

This manual is in PDF format. The following environments are recommended for reading this manual.

PDF Reader: Adobe Reader

■ Intended readers

This manual is written for readers with sufficient basic knowledge of how to control measuring instruments using a PC.

Familiarize yourself with the syntax of the SCPI commands that are used with the product before you use them.

Structure of the manual

This manual consists of the following sections.

- Overview
- Setup
- Message Overview
- Commands
- Tutorial
- Appendix

Trademarks

Microsoft Windows is a trademark of Microsoft Corporation in the United States and/or other countries.

All other company and product names used in this manual are trademarks or registered trademarks of their respective owners.

Firmware version of the product to which this manual applies

This manual applies to products with firmware versions 2.2x.

■ Measuring instrument interface standards

This product complies with the following standards.

- IEEE Std 488.2-1992 IEEE Standard Codes, Formats, Protocols, and Common Commands For Use With IEEE Std 488.1-1987
- IEEE Std 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation
- Standard Commands for Programmable Instruments (SCPI) version 1999.0
- Universal Serial Bus Specification Rev 2.0
- Universal Serial Bus Test and Measurement Class Specification (USBTMC) Rev 1.0
- Universal Serial Bus Test and Measurement Class, Subclass USB488 Specification (USBTMC-USB488) Rev 1.0
- TCP/IP Instrument Protocol Specification VXI-11
- TCP/IP-IEEE488.2 Interface Specification VXI-11.3
- IVI-6.1 IVI High-Speed LAN Instrument Protocol (HiSLIP) Rev 1.0
- LXI 1.4 Core Specification 2011

■ Copyright

Installing the VISA library

VISA (Virtual Instrument Software Architecture) was developed by the IVI Foundation. It is the standard specification for measurement instrument connection software.

To use the VISA library (VISA COM) with the I/O library, the VISA library must be installed on the controller (Windows).

You have to install one of the following VISA libraries (driver software that is implemented according to the VISA specifications).

- NI-VISA by National Instruments Corporation (Ver. 5.1.1 or later)
- Keysight VISA (Keysight IO Libraries Suite 16.0 or later) by Keysight Technologies
- KI-VISA Ver. 5.0.4 or later

—Note—

- Do not install multiple VISA libraries on the same PC. Doing so may cause errors.
- If NI-VISA or Keysight VISA is already installed on your PC, you do not need to install KI-VISA.

KI-VISA is an original VISA library developed by Kikusui Electronics Corporation that supports the IVI VISA 5.0 specifications. You can download the most recent version of this library from the Kikusui Electronics Corporation website (http://www.kikusui.co.jp/en/download/).

Setting Up the Interface

The product is standard equipped with RS232C, USB, and LAN interfaces.

There is no need to switch interfaces. All interfaces can be used simultaneously. Each interface can be turned off using CONFIG settings.

RS232C (p. 25)

USB (p. 27)

LAN (p. 28)

Accessing and Operating the Product from a Web Browser (LAN) (p. 32)

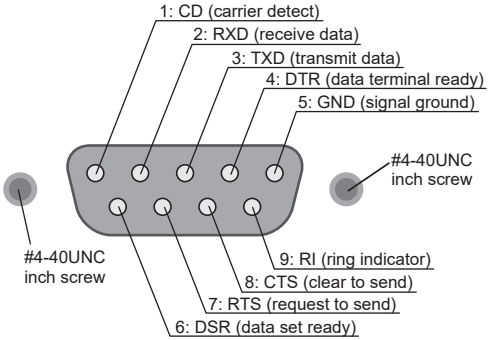
RS232C

■ RS232C connection

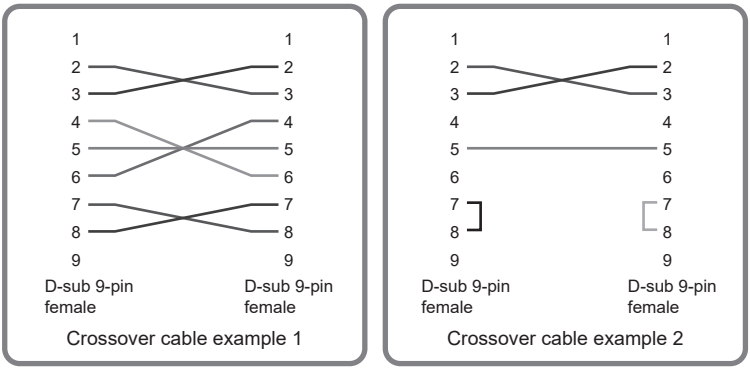
The RS232C port on the product side is D-sub 9-pin. Check that the product and your PC are off, and then connect them.

For the RS232C cable, use a D-sub, 9-pin, female-to-female AT crossover cable.

The following figure shows the connector pinout when you are facing the rear panel.



RS232C pinout (male end)



If you are not using flow control, you do not have to connect all the pins (as shown in “Crossover cable example 2”). Flow control cannot be performed using the 9-pin AT connector (cable shown in “Crossover cable example 2”).

■ Protocol

The RS232C protocol is shown in the following table.

The underlined value is the factory default value.

Parameter	Value
Bitrate: Data rate	9600 bps/ 19200 bps/ 38400 bps/ 57600 bps/ 115200 bps
Data Bits: Data length	8 bit (fixed)
Stop Bits: Stop bits	1 bit (fixed)
Parity: Parity	None (fixed)
Flow Control: Flow control	CTS/RTS, none

■ RS232C settings

1 **Press SYSTEM > Interface.**

2 **Press Modify, and use the rotary knob to select the parameter you want to change.**

3 **Press Edit, and then use the numeric keypad or the rotary knob to select the value you want to change.**

Press ENTER to continue setting other parameters.

4 **Press Apply.**

A confirmation screen appears.

5 **Press ENTER.**

To cancel, use the rotary knob to select NO, and then press ENTER.

■ Break signal

The break signal is used as a substitute for the IEEE488.1 dcl/sdc (Device Clear, Selected Device Clear) message.

—Note—

If you are going to use remote control, we recommend that you set the PLZ-5WH2 in remote mode (set SYST:COMM:RLST to REMote or RWLock). To use remote programming, send “SYST:COMM:RLST REM” or “SYST:COMM:RLST RWL” at the beginning of the program.

USB

To use the USB interface to control the product, a device driver that supports the USB Test & Measurement class (USBTMC) must be installed on the controller. The USBTMC driver is installed automatically by the VISA library.

■ USB connection

Turn off the PLZ-5WH2 and PC.

Using a USB cable, connect the PLZ-5WH2 to the PC and then turn on the PLZ-5WH2.

■ Service request

The product is equipped with service request and serial polling functions.

■ USB function

Complies with USB Specification 2.0

Complies with USBTMC Specification 1.0 and USBTMC-USB488 Specification 1.0

Baud rate: 480 Mbps maximum (high speed)

Message terminator: LF or EOM during reception, LF + EOM during transmission

VID (vendor ID)

0x0B3E

PID (product ID)

0x1042

—Note—

If you are going to use remote control, we recommend that you set the PLZ-5WH2 in remote mode (set SYST:COMM:RLST to REMote or RWLock). To use remote programming, send “SYST:COMM:RLST REM” or “SYST:COMM:RLST RWL” at the beginning of the program.

LAN

To use the LAN interface to control the product, middleware that supports the SCPI-Telnet, VXI-11, HiSLIP, or SCPI-RAW protocol is required. The middleware is installed automatically by the VISA library.

The LAN interface board has a Web browser interface (Web Browser Interface). You can configure the LAN interface settings from your PC's Web browser.

For information on topics such as connecting to your corporate LAN, your IP address, your host name, and security, contact your network administrator.

If you are using a host name (a Bonjour host name), you have to install Apple Bonjour.

⚠WARNING

If a network problem occurs, an unexpected dangerous voltage may occur that may cause electric shock, fire, physical damage to the DUT, and so on. If you are going to remotely control the PLZ-5WH2 from a distance, install a Web camera or take other measures to monitor the status.

■ LAN connection

Use a standard LAN cable (category 5 and straight) to connect the product to a network hub or router.

■ LAN settings

Normally, use the product with “Method” under “IP Address” set to “Automatic” (factory default setting). To set the IP address manually, set Method to Static, and then set the IP address.

For details, see the user’s manual.

1 **Press SYSTEM > Interface.**

2 **Press Modify, and use the rotary knob to select the parameter you want to change.**

3 **Press Edit, and then use the numeric keypad or the rotary knob to select the value you want to change.**

Press ENTER to continue setting other parameters.

4 **Press Apply.**

A confirmation screen appears.

5 **Press ENTER.**

The LAN interface restarts, and the settings are applied. To cancel, use the rotary knob to select NO, and then press ENTER.

⚠WARNING

Possible damage to the equipment and electric shock. The LAN interface can be accessed from anywhere on the network that the product is connected to. If necessary, set a password for the Web browser interface.

—Note—

If you are going to use remote control, we recommend that you set the PLZ-5WH2 in remote mode (set SYST:COMM:RLST to REMote or RWLock). To use remote programming, send “SYST:COMM:RLST REM” or “SYST:COMM:RLST RWL” at the beginning of the program.

■ Service request

The product is equipped with service request and serial polling functions.

■ LAN function

Complies with LXI 1.4 Core Specification 2011

Complies with the SCPI-Telnet, VXI-11, HiSLIP, and SCPI-RAW protocols

Baud rate: 100 Mbps maximum (auto negotiation)

AUTO MDIX

Web browser interface features

Instrument information, network information, display of VISA resource information, checking the connected product, remote control from browser, changing network settings, system information, license information, password setting

■ Resetting or Initializing the LAN interface

You can use the SYSTEM settings to reset or initialize the LAN settings.

When reset or initialized, network settings are changed as follows.

The items with an ✓ mark are returned to their default values.

Reset	Default values, resetting to	Parameter	Default value
✓	✓	IP Address Method	Auto
	✓	DNS Server	—
	✓	WINS Server	—
	✓	Desired Hostname	Model name and serial number
	✓	Desired Description	KIKUSUI XXXX Electronic Load (XXXX is the model name) and serial number
✓	✓	Dynamic DNS	Enable
✓	✓	mDNS	Enable
✓	✓	NetBIOS Over TCP/IP	Enable
✓	✓	Password for the Web browser interface	Not set

Resetting

1 **Press SYSTEM > Interface.**

2 **Press Modify > LAN Reset.**

A confirmation screen appears.

3 **Press ENTER.**

The LAN interface settings are reset.

Returning to factory default settings (Initializing)

1 **Press SYSTEM > Interface.**

2 **Press Modify > Default.**

A confirmation screen appears.

3 **Press ENTER.**

The LAN interface settings are returned to their factory default settings.

Accessing and Operating the Product from a Web Browser (LAN)

You can configure the LAN interface settings from your PC's Web browser.

Use the latest browser version (Recommended browser: Internet Explorer11, Chrome, Safari).

You can access the Web browser in the following manners.

- Enter the URL in the address bar of the browser.

  The website URL is http:// followed by the IP address of the PLZ-5WH2. You can check the IP address on the Information screen (press SYSTEM and then Information).

  Example: When the IP address is 169.254.7.8

  http://169.254.7.8

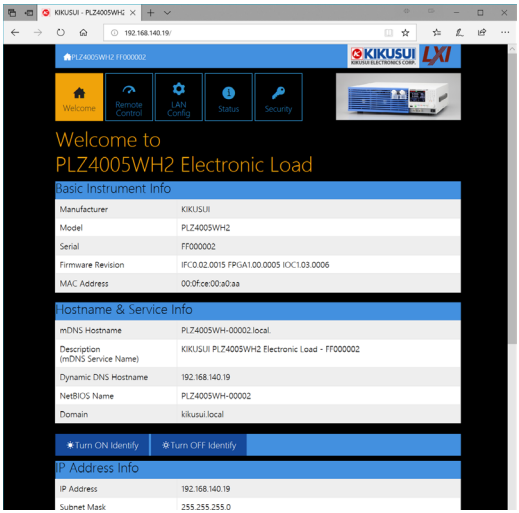
- If you using a VISA library, click on the web link from the search results of the application supplied by the VISA vendor.

  Using an application (National Instruments NI-MAX, Keysight Connection Expert, Kikusui KI-VISA Instrument Explorer, etc.) provided by VISA vendors, search for VXI-11 measuring instruments, and from the search results, simply click the web link to open the Web browser interface.

■ WELCOME page

This is the first page that is displayed when you connect to the Web browser interface.

There, you can display the measuring instrument information, network information, and VISA resources (I/O resources) information, and also check the connected PLZ-5WH2. Click the navigation menu at the top to go to another page.

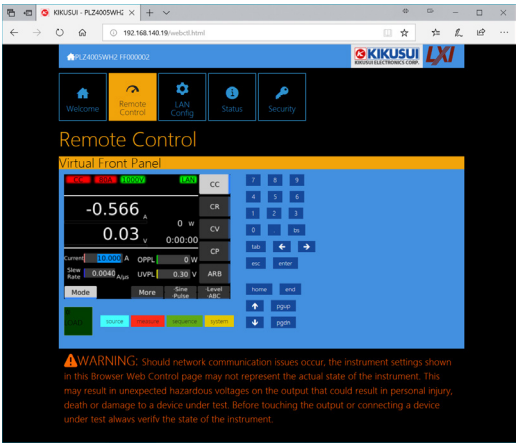


Turn ON Identify: “LXI Web Identify” appears in the front panel display of the connected PLZ-5WH2. This allows you to identify the connected instrument.

Turn OFF Identify: The displayed “LXI Web Identify” disappears.

■ Remote Control page

You can remotely control the product from a browser. The various buttons have the same functions as those on the front panel of the product.

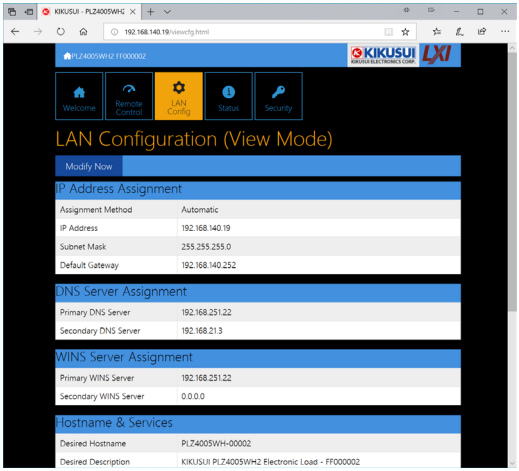


⚠WARNING

When a problem occurs in the network, the information displayed in the browser may differ from the actual state of the device. As a result, an unexpected dangerous voltage may occur that may cause human death or injury, or physical damage to the DUT and so on. Therefore, prior to touching the input terminals and connecting the DUT, be sure to check the status of the equipment.

■ LAN Configuration page

You can display (View Mode) and change (Modify Mode) the network settings.



Navigation (View Mode)

Modify Now: Changes to the network setting edit screen (Modify Mode).

Navigation (Modify Mode)

Undo: Returns the edited contents to the state before editing.

Apply: Applies the edited contents.

Reset: Resets the network settings.

Default: Returns the network settings to the factory default settings.

Back to View Mode: Changes to the network setting view screen (View Mode).

IP Address Assignment

You can set the IP address. You can choose between automatic assignment and assignment of a fixed address.

In the case of automatic assignment of IP address, we recommend using the DHCP server function using a router as far as possible.

If the DHCP server function is not used, it takes about 60 seconds until determination that address assignment with DHCP has failed. Then, an address between 169.254.0.0 to 169.254.255.255 is assigned by link local address (Auto-IP).

DNS Server Assignment

Sets the address of the DNS server.

WINS Server Assignment

Sets the address of the WINS server.

Hostname & Services

You can set the host name and so on. If you set the host name, you can use it in place of the IP address to access the LAN interface. Normally, we recommend that you select “Enable Dynamic DNS”, “Enable mDNS”, and “Enable NetBIOS Over TCP/IP”.

If you leave the Hostname and Description boxes empty and click “Apply,” the host name will be created from the model name and serial number.

TCP Ports (View Mode)

The number of the TCP port in use is displayed. You cannot change the port number.

Auto Clock Adjustment

You can set auto clock adjustment. Enter the NTP server address to be used for clock adjustment and select the “Enable Auto Adjustment” check box to enable auto clock adjustment when the power is turned on.

To use auto clock adjustment, the product must be connected to the Internet when it is turned on. If a link local address was set for the IP address, auto clock adjustment does not function.

Reset and factory default settings

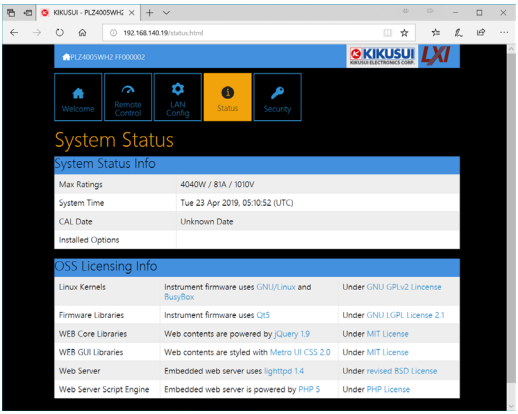
If you click Reset or Default, network settings are changed as follows.

The items with an ✓ mark are returned to their default values.

Reset	Default	Parameter	Default value
✓	✓	IP Address Assignment	Automatic
	✓	DNS Server Assignment	—
	✓	WINS Server Assignment	—
	✓	Desired Hostname	Model name and serial number
	✓	Desired Description	KIKUSUI XXXX Electrical Safety Analyzer (XXXX is the model name) and serial number
✓	✓	Enable Dynamic DNS	Yes
✓	✓	Enable mDNS	Yes
✓	✓	Enable NetBIOS Over TCP/IP	Yes

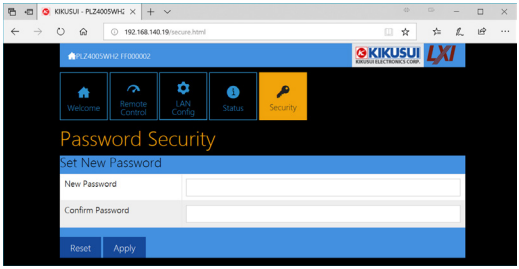
System Status page

This page shows the system information and the license information of the open-source software.



■ Password Security page

You can set and change the password for the Web browser interface here.



When a password has been set, that password is required in order to use the following functions.

Remote control from Remote Control page

Editing of LAN Configuration page

Changing/deleting the password

Set New Password

Enter the password.

You can use alphanumeric characters, hyphens, and underscores for the password. The first character must be an alphabet. You can enter up to 15 characters.

Changing or deleting the password

After the password has been set, the screen for changing the password appears when you enter the password.

To change the password, enter the present password in “Current Password”, enter the new password in “New Password” and “Confirm Password”, and then click “Apply”.

To disable password protection, enter the present password in “Current Password”, leave “New Password” and “Confirm Password” blank, and click “Apply”.

If you forget the password

If you forget the password, reset the LAN interface setting in the SYSTEM settings or initialize the product to its factory default settings.

About Commands

The information that is transferred between the controller (the PC) and the device (the product) is referred to as messages. This product uses the SCPI language for these messages.

The messages that the PC sends to the product are commands. The messages that the product sends to the PC are responses.

Command hierarchy

SCPI is an ASCII-based command language that was designed for test and measuring equipment. The command structure is composed of the common roots or nodes that are the building blocks of the SCPI subsystem. A command consists of a program header, parameters, and punctuation marks.

The following table uses the SOURce subsystem as an example to explain the hierarchy.

Program header	Parameter	Node level
[SOUR]		Root node
:VOLT		2nd level
:PROT		3rd level
:STAT	<boolean>	4th level
:PROG		2nd level
:SEL		3rd level
:REN	<string>	4th level

A colon (:) separates a higher node from a lower node.

Command syntax

In this manual, SCPI commands are expressed in the following format.

```
[:SOURce]:CURRent:SLEWrate {<numeric>|MINimum|MAXimum}
```

■ Basic Notation

- You can use long form and short form.

  Long form: SCPI commands are written with all the characters.

  Short form: SCPI commands are written by omitting the lowercase characters.
- No distinction is made between uppercase and lowercase.

  CURRENT, Current, and current are all acceptable as long form notations.

  CURR, Curr, and curr are all acceptable as short form notations.
- A space separates a program header and its parameters.
- Multiple parameters are separated by commas.
- Multiple commands are separated by semicolons (compound command).

  ```
  VOLTage:PROTection:STATe ON;LOWer 10V
  ```

  In the second command, VOLTage:PROTection: is omitted. This is possible because that path is set to VOLTage:PROTection by the first command (VOLTage:PROTection:STATe ON).

  This compound command is equivalent to entering the following commands.

  ```
  VOLTage:STARt:LEVel 50PCT
  VOLTage:STARt:STATe ON
  ```

  If you specify a node that is not defined in the current path, an error will occur.

  By using colons and semicolons, you can concatenate commands of different subsystems.

  ```
  SOURce:FUNCtion ACW;:SENSe:CURRent:SECondary PHOLd
  ```

  There are two root nodes in this compound command: SOURce and SENSe.

  When the second command or later begins with a colon, the path that was specified by the previous command is cleared.
- The maximum length of a command that you can transmit on a single line is 512 bytes.

■ Special symbols

The special symbols that are used in SCPI commands are explained below.

- Characters and numbers enclosed by { and } and delimited by "|" indicate that one of the delimited items is to be selected.

  Do not include the { and } symbols in the actual program.
- <> denotes program data.

  Do not include the < and > symbols in the actual program.
- [ ] denotes optional data.

  The data enclosed in square brackets ([ ]) can be omitted. If omitted, the default value is applied.

  Do not include the [and] symbols in the actual program.

■ Queries

You can query the device settings and status.

To make a query, append a question mark to the end of the program header section. If the query has parameters, insert a space after the question mark, and then write the parameters.

```
CURRent? MIN
```

Response

This is the response to a query. It is a message always sent from the device to the PC. It conveys device status or measured value to the PC.

—Note—

If you want to send two queries on separate lines, send the second query after you have received the response to the first one.

■ Program terminator

All commands must be terminated with a valid terminator.

	RS232C	USB	LAN	
			VXI-11, HiSLIP	SCPI-RAW
Receiving	LF	LF or EOM	LF or END	LF
Sending	LF	LF+EOM	LF+END	LF

When you terminate a command string, the path is reset to the root level.

—Note—

CR (ASCII 0x0D) is not a terminator.

■ Common commands

There are commands that are common to the IEEE488.2 and SCPI standards for functions such as resetting devices and performing self-diagnoses. These common commands start with an asterisk (“*”). These commands may have one or multiple parameters.

Parameters

The SCPI parameter format is derived from the program parameter format that is defined in IEEE 488.2.

The program data expression format that the this product uses is shown below.

■ Non-numeric parameters

String data (String)

String data is used when a series of ASCII characters (20H to 7EH) are requested.

Enclose strings in single (' ') or double quotation ( ) marks. The opening and closing quotation marks must match (you cannot mix single and double quotation marks).

```
PROGram:CREate "/My test program"
```

If you want to include a quotation mark as part of the string, enter consecutive quotation marks (with no characters between them).

Character data (Character)

Character data is used when only a limited number of values are available for a program setting. Responses are returned in short form.

```
TRIGger:ACQuire:SOURce IMMediate
```

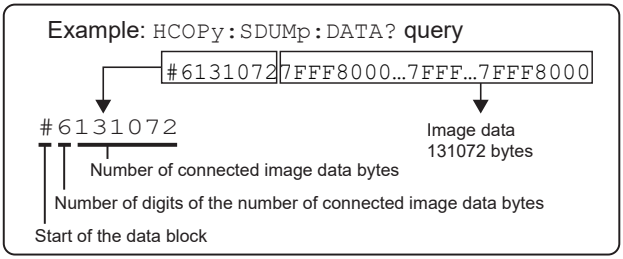
Boolean data (Boolean)

Boolean data is used to express a condition of 1 or 0, or ON or OFF. Responses are returned as 1 or 0.

```
OUTPut {ON|OFF|1|0}
```

Block data (block)

Arbitrary block data that starts with #.



■ Numeric parameters

NR1

Represents an integer value.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

If a 0 is returned in the response data, it is returned as +0.

NR2

Represents a real number in floating-point format.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

NR3

Represents a real number in scientific notation.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

If 380 is returned in the response data, it is returned as +3.80000E+02. Five decimal places are used.

NRf

NRf is a generic term that includes NR1, NR2, and NR3.

If a value outside the setting range is specified, an error (-222, “Data out of range”) will occur.

Numeric

Numeric parameter for values such as the decimal point, optional prefixes, and measurement units.

Numbers are expressed the same as NRf.

MINimum, MAXimum, and the like are available as substitutes for declaring certain values.

You can also use units such as V, A, and W in numeric parameters.

If a value outside the setting range is specified, an error (-222, “Data out of range”) will occur.

■ Special form numeric parameters

The special form numeric parameters MINimum and MAXimum can be used as substitutes for the actual maximum and minimum values when the parameter is numeric.

The following example sets the overcurrent protection value to the minimum value.

```
SOURce:CURRent MINimum
```

You can query the minimum and maximum values for most parameters.

```
SOURce:CURRent? MAX
```

```
SOURce:CURRent? MIN
```

■ Measurement units

The default measurement units are listed below. Commands are accepted even if measurement units are not specified.

- V (voltage)
- S (seconds)
- A (current)
- HZ (frequency)
- AH (current capacity)
- SIE (conductance)
- WH (power capacity)
- PCT (%)

The following optional prefixes are supported. If you use optional prefixes, specify the measurement unit.

- M (milli)
- K (kilo)
- U (micro)

—Note—

- The unit symbols in the International System of Units contain lowercase characters. The IEEE standard uses uppercase characters. SCPI commands are not case sensitive.
- Commands are accepted whether or not measurement units are specified.
- To enter “μ” in the data, use “U” instead.

*CLS

Clears all event registers including the status byte, event status, and error queue.

Clears the operation complete standby that was created by the *OPC or *OPC? command.

Command

```
*CLS
```

*ESE

Sets the event status enable register that is counted by the event summary bit (ESB) of the status byte.

Command

```
*ESE <NRf>
```

```
*ESE?
```

Parameter

Value: 0 to 255

Example: When *ESE 16 is transmitted, bit 4 of the event status enable register is set. Each time the execution error bit (bit 4) of the event status register is set, the summary bit (ESB) of the status byte is set.

Response: NR1

*ESR

Queries the event status register.

The event status register is cleared when read.

Command

```
*ESR?
```

Response: NR1

*IDN

Queries the model name and firmware version of the product.

Command

```
*IDN?
```

Response

Returns the following information in response to *IDN?.

Response example: When the model name is PLZ1005WH2, the serial number is AB123400, and the versions are IFO0.53.0086, FPGA0.126.0011, and IOC0.07.0062

```
KIKUSUI,PLZ1005WH2,AB123400,IFO0.53.0086 FPGA0.126.0011 IOC0.07.0062
```

*OPC

Sets the OPC bit (bit 0) of the event status register when all the commands that are in standby have been processed.

See IEEE 488.2-1992 section 12.5.3.

Command

```
*OPC
```

```
*OPC?
```

Response

Returns “1” when all the commands that are in standby have been processed.

*OPT

Queries the options that are installed in the product.

Command

```
*OPT?
```

Response

Returns the installed options in comma-separated string format. Returns “0” if no options are installed.

*PSC

Sets whether the event status enable register and service request enable register are cleared when the POWER switch is turned on.

Command

```
*PSC <boolean>
```

```
*PSC?
```

Parameter <boolean>

Value:	ON(1)	When the POWER switch is turned on, the *ESE and *SRE settings are cleared.
	OFF(0)	When the POWER switch is turned on, the *ESE and *SRE settings are not cleared.

Example To enable the power-on SRQ function:

```
*PSC 0;*SRE 32;*ESE 128
```

Response: NR1

*RCL

Recalls setup memories.

Clears alarms.

Aborts the trigger subsystem operation.

Command

```
*RCL <NRf>
```

Parameter

Value: 0 to 19 memory number

Example

```
*RCL 1
```

*RST

Resets the panel settings.

Clears alarms (if they cannot be cleared, alarms continue).

Aborts the trigger subsystem operation.

Clears the OPC bit (bit 0) of the status event register.

Command

```
*RST
```

Values after initialization

Commands	*RST
DATA:BSIZ	65536
DATA:FORM	CURR, VOL, POW
DATA:INT:GATE	LOAD_ON
DATA:INT:GATE:ARES	ON
DISP:CAP	OFF
DISP:ENER	OFF
DISP:ETIM	ON
INP/ OUTP	OFF
INP:CUTO:CAP/ OUTP:CUTO:CAP	0
INP:CUTO:CAP:STAT/ OUTP:CUTO:CAP:STAT	OFF
INP:CUTO:ENER/ OUTP:CUTO:ENER	0
INP:CUTO:ENER:STAT/ OUTP:CUTO:ENER:STAT	OFF
INP:CUTO:ETIM/ OUTP:CUTO:ETIM	0
INP:CUTO:ETIM:STAT/ OUTP:CUTO:ETIM:STAT	OFF
INP:CUTO:VOLT/ OUTP:CUTO:VOLT	0
INP:CUTO:VOLT:STAT/ OUTP:CUTO:VOLT:STAT	OFF
INP:EXT:LOG/ OUTP:EXT:LOG	POS
INP:MSYN:ACC/ OUTP:MSYN:ACC	OFF
INP:PON:STAT/ OUTP:PON:STAT	AUTO
SENS:APER	0.1S
ARB:COUN	3S
ARB:RESP	500US
COND	0SIE
COND:EXT:FCON	OFF
COND:PULS:LEV	0SIE

Commands	*RST
COND:PULS:FREQ	1HZ
COND:PULS:DCYC	50PCT
COND:RANG	HIGH
COND:RESP	NORM
CURR	0A
CURR:EXT:ACON	OFF
CURR:EXT:FCON	OFF
CURR:PROT	MAX
CURR:PROT:STAT	OFF
CURR:PULS:LEV	0A
CURR:PULS:FREQ	1HZ
CURR:PULS:DCYC	50PCT
CURR:SINE:AMPL	0A
CURR:SINE:FREQ	1HZ
CURR:SLEW	MIN
CURR:SST	500US
FUNC	CC
FUNC:CVOP	OFF
POW	0W
POW:EXT:FCON	OFF
POW:PROT	MAX
POW:PROT:STAT	OFF
PROG	""
VOLT	0V
VOLT:EXT:FCON	OFF
VOLT:PROT:ACT	LIM
VOLT:PROT:LOW	0V
VOLT:PROT:STAT	OFF
VOLT:RESP	NORM
TRIG:ACQ:COUN	1
TRIG:ACQ:DEL	0S
TRIG:ACQ:SOUR	IMM
TRIG:ACQ:INT:STAT	OFF
TRIG:ACQ:INT:TIME	0.1S
TRIG:TRAN:DEL	0S
TRIG:TRAN:SOUR	IMM
SYST:BEEP	ON
SYST:BEEP:KEY	ON

Commands	*RST
SYST:BEEP:PROT	ON
SYST:BEEP:SCPI	ON
SYST:CONF:DIGITAL2:DIR	INP
SYST:CONF:RSEN	OFF
SYST:KLOC:LEV	3

Commands	*RST
SYST:PASS:NEW	""
SYST:SSAV	OFF
SYST:SSAV:DEL	60S
SYST:TZON	UTC

*SAV

Saves the panel settings to the setup memory.

Command

```
*SAV <NRf>
```

Parameter

Value: 0 to 19 memory number

Example

```
*SAV 1
```

*SRE

Sets the service request enable register.

The service request enable register can be used to select which summary messages in the status byte register will perform service requests.

To clear the service request enable register, send *SRE 0. If the register is cleared, service requests cannot be generated using status information.

Command

```
*SRE <NRf>
```

```
*SRE?
```

Parameter

Value: 0 to 255

Example: Sending *SRE 8 sets bit 3 of the service request enable register. Each time the summary bit (bit 3) of the QUEStionable status register in the status byte is set, a service request message is generated.

Response: NR1

*STB

Queries the contents of the status byte register and the MSS (master summary status) message.

The response is the same as serial polling only with the exception that the MSS message appears in place of the RQS message in bit 6.

Command

```
*STB?
```

Response: NR1

*TRG

Trigger command.

Executes a trigger on the TEST trigger group.

This is a substitute command for IEEE 488.1 get (Group Execute Trigger).

If the device is in a state in which it does not accept triggers, an SCPI error (-211, “Trigger ignored”) occurs.

See IEEE 488.2-1992 section 10.37.

Command

```
*TRG
```

*TST

Executes a self-test.

You can check which error occurred with SYST:ERR? command. See IEEE 488.2-1992 section 10.38.

Command

```
*TST?
```

Response

If there is no problem, returns 0 in response to *TST?. Returns an error code if there is a problem.

*WAI

Prevents the device from executing subsequent commands until all operations that are in standby have completed.

Command

```
*WAI
```

ABORt Command

This product has four trigger subsystems (ACQuire, TRANsient, PULSe, SINE).

ACQuire is a measurement trigger subsystem.

TRANsient is a trigger subsystem for executing sequences.

PULSe is a trigger subsystem for the pulse function.

SINE is a trigger subsystem for the sine function.

ABOR

Aborts measurements, tests, and other operations in all trigger subsystems (ACQuire, TRANsient, PULSe, SINE).

The product's trigger state immediately after it turns on is the same as its trigger state after it receives an ABOR command.

If an ABOR command is sent when the trigger function is not running, the measurement data is not discarded.

You cannot specify a trigger subsystem with the ABOR command. It is always interpreted as ALL.

Command

```
ABORt[:ALL]
```

ABOR:ACQ

Aborts measurement operations.

If an ABOR command is sent when the trigger function is not running, the measurement data is not discarded.

Command

```
ABORt:ACQuire
```

ABOR:PULS

Stops the pulse operation.

Command

```
ABORt:PULSe
```

ABOR:SINE

Stops the sine operation.

Command

```
ABORt:SINE
```

ABOR:TRAN

Stops the sequence operation.

Command

```
ABORt:TRANsient
```

DATA:BSIZ

Sets the buffer size (maximum number of measurement points) of the data logger function.

If the number of measurement points exceeds the buffer size, measured values are cleared in order from the oldest value. Changing the buffer size deletes all the data that has already been registered.

Command

```
DATA:BSIZe <numeric>
```

```
DATA:BSIZe?
```

Parameter

Value: 1024, 2048, 4096, 8192, 16384, 32768, 65536

Settings are reset to default when the *RST command is sent.(65536)

Response: NR1

Example

```
DATA:BSIZ 16384
```

DATA:FORM

Sets the response format (type of measured data and order) to be used when referencing data in the data logger.

You cannot specify the same setting value 2 or more times.

Command

```
DATA:FORMat
    <Character>[,<Character>,<Character>,<Character>,<Character>,
    <Character>]
```

```
DATA:FORMat?
```

Parameter

Value:	CURRent	Sets the response format to current value (A)
	VOLTage	Sets the response format to voltage value (V)
	POWer	Sets the response format to power value (W)
	ETIMe	Sets the response format to elapsed time (s)
	CAPacity	Sets the response format to current capacity value (Ah)
	ENERgy	Sets the response format to power capacity value (Wh)

Settings are reset to default when the *RST command is sent.(CURR,VOL,POW)

Response: Character

Example

```
DATA:FORM VOLT,CURR,ETIM
```

DATA:INT:GATE

Sets the integration time.

Command

```
DATA:INTegral:GATE {NONE|LOAD_ON|PROG_RUN}
```

```
DATA:INTegral:GATE?
```

Parameter

Value:	NONE	Manual integration start/stop only.
	LOAD_ON	Starts integration at load on. Stops integration at load off. (default)
	PROG_RUN	Starts integration at sequence start. Stops integration at sequence completion.

Settings are reset to default when the *RST command is sent.

Response: Character

Example

```
DATA:INT:GATE LOAD_ON
```

DATA:INT:GATE:ARES

Sets whether past integration is reset at the start of the integration operation.

Command

```
DATA:INTegral:GATE:ARESet <boolean>
```

```
DATA:INTegral:GATE:ARESet?
```

Parameter

Value:	ON(1)	Reset (default)
	OFF(0)	Don’t reset (accumulation of past integrated values)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
DATA:INT:GATE:ARES ON
```

DATA:INT:RES

Resets integrated data (current capacity (Ah), power capacity (Wh), elapsed time (s)).

Command

```
DATA:INTegral:RESet
```

DATA:INT:STAR

Manually starts the integration operation.

The currently held integrated values are saved. This command is ignored if the integration operation is already started.

Command

```
DATA:INTegral:STARt
```

DATA:INT:STOP

Manually stops the integration operation.

The currently held integrated values are saved. This command is ignored if the integration operation is currently stopped.

Command

```
DATA:INTegral:STOP
```

DATA:POIN

Queries the number of measurement points registered in the data logger.

Command

```
DATA:POINt?
```

Response: NR1

DATA:R

Queries the measurement values registered in the data logger from oldest to youngest.

The queried data is deleted from the data logger records.

Command

```
DATA:R? [<numeric>]
```

Parameter

Value: Specifies the number of data points to be queried in NRf format (1 or more). By default, all the data is queried.

Response

Returns the measured values registered in the data logger in comma-separated NR3 format.

The type of measured values and the order depends on the DATA:FORM (p. 57) setting.

The maximum number of data points that can be queried at one time varies as follows depending on the number of items specified by DATA:FORM.

Number of items specified by DATA:FORM	Number of data points that can be queried
6	512
5	640
4	768
3	1024
2	1872
1	3072

Example

If DATA:FORM is set to CURR, VOL, POW, the recorded measurement values of the first point are 10 A, 15 V, 150 W, and those of the second point are 12 A, 15 V, 180 W, the following is returned.

+1.00000E+01,+1.50000E+01,+1.50000E+02,+1.20000E+01,+1.50000E+01,+1.80000E+02

DISP:CAP

Sets whether to display the current capacity (Ah) on the display.

Command

```
DISPlay[:WINDow]:CAPacity[:STATe] <boolean>
```

```
DISPlay[:WINDow]:CAPacity[:STATe]?
```

Parameter

Value:	ON(1)	Displayed
	OFF(0)	Not displayed (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
DISP:CAP ON
```

DISP:ENER

Sets whether to display the power capacity (Wh) on the display.

Command

```
DISPlay[:WINDow]:ENERgy[:STATe] <boolean>
```

```
DISPlay[:WINDow]:ENERgy[:STATe]?
```

Parameter

Value:	ON(1)	Displayed
	OFF(0)	Not displayed (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
DISP:ENER ON
```

DISP:ETIM

Sets whether to display the elapsed time on the display.

Command

```
DISPlay[:WINDow]:ETIMe[:STATe] <boolean>
```

```
DISPlay[:WINDow]:ETIMe[:STATe]?
```

Parameter

Value:	ON(1)	Displayed (default)
	OFF(0)	Nothing is displayed.

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
DISP:ETIM ON
```

HCOP:SDUM:DATA

Retrieves the screen capture of the present screen.

Command

```
HCOPy:SDUMp:DATA?
```

Response

The screen image (PNG) is returned in block (#6<length><data>) format.

INITiate Command

INIT:ACQ

Starts the measurement trigger function.

If TRIG:ACQ:SOUR (p. 163) is set to IMM, measurement is immediately started.
If it is set to other than IMM, the system switches to the trigger wait state, and measurement is started with the respective conditions. Executing this command deletes all the measurement records in the data logger.

Command

```
INITiate[:IMMediate]:ACQuire
```

INIT:PULS

Enables the pulse function.

The pulse function is enabled until ABOR:PULS or ABOR is executed. This command cannot be used while the sequence trigger function is running.

Command

```
INITiate[:IMMediate]:PULSe
```

INIT:SINE

Enables the sine function.

The sine function is enabled until ABOR:SINE or ABOR is executed. This command cannot be used while the sequence trigger function is running.

Command

```
INITiate[:IMMediate]:SINE
```

INIT:TRAN:PROG

Starts the sequence start trigger function in the program selected by PROG.

If TRIG:TRAN:SOUR (p. 168) is set to IMM, the sequence is immediately started. If it is set to other than IMM, the system switches to the trigger wait state, and the sequence is started with the respective conditions.

Command

```
INITiate[:IMMediate]:TRANsient:PROGram
```

INP/ OUTP

Sets the load on/off.

Only when load is set to on, the duration to keep the load turned on can be specified.

Command

```
INPut[:STATe] <boolean>[,<numeric>]
OUTPut[:STATe] <boolean>[,{<numeric>]
INPut[:STATe]?
OUTPut[:STATe]?
```

Parameter <boolean>

Value:	ON(1)	Load on
	OFF(0)	Load off (default)

Parameter <numeric>

If set to ON (1), specify the duration to keep the load turned on. When the duration passes, the load turns off. When omitted, the load does not turn off based on the elapsed time.

Value:	1 to 3600000	
	OFF(0)	Load off (default)
Unit:	S	

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP ON,30
```

INP:CUTO:CAP/ OUTP:CUTO:CAP

Sets the integrated current of the cutoff function.

Command

```
INPut:CUTOff:CAPacity[:LEVel][:UPPer] <numeric>
OUTPut:CUTOff:CAPacity[:LEVel][:UPPer] <numeric>
INPut:CUTOff:CAPacity[:LEVel][:UPPer]?
OUTPut:CUTOff:CAPacity[:LEVel][:UPPer]?
```

Parameter

	PLZ1005WH2	PLZ2005WH2	PLZ4005WH2	PLZ12005WH2	PLZ20005WH2
Value:	0.000 m to 70.000 k	0.000 m to 140.000 k	0.000 m to 280.000 k	0.000 m to 800.000 k	0.000 m to 1400.000 k
Resolution:	0.001 m (0.000 m to 1 000.000 m) 0.001 (1.001 to 1000.000) 0.001 k (1.001 k to 1000.000 k) 0.001 M (1.001 M to 1.400 M)				
Unit:	AH				

Settings are reset to default when the *RST command is sent.(0 Ah)

Response: NR3

Example

```
INP:CUTO:CAP 3.001
```

INP:CUTO:CAP:STAT/ OUTP:CUTO:CAP:STAT

Enables or disables the cutoff function for the integrated current.

Command

```
INPut:CUTOff:CAPacity:STATe <boolean>
INPut:CUTOff:CAPacity:STATe?
OUTPut:CUTOff:CAPacity:STATe <boolean>
OUTPut:CUTOff:CAPacity:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP:CUTO:CAP:STAT ON
```

INP:CUTO:ENER/ OUTP:CUTO:ENER

Sets the integrated power of the cutoff function.

Command

```
INPut:CUTOff:ENERgy[:LEVel][:UPPer] <numeric>
OUTPut:CUTOff:ENERgy[:LEVel][:UPPer] <numeric>
INPut:CUTOff:ENERgy[:LEVel][:UPPer]?
OUTPut:CUTOff:ENERgy[:LEVel][:UPPer]?
```

Parameter

	PLZ1005WH2	PLZ2005WH2	PLZ4005WH2	PLZ12005WH2	PLZ20005WH2
Value:	0.000 to 40.000 M	0.000 to 80.000 M	0.000 to 160.000 M	0.000 to 500.000 M	0.000 to 800.000 M
Resolution:	0.001 (0.000 to 1000.000) 0.001 k (1.001 k to 1000.000 k) 0.001 M (1.001 M to 800.000 M)				
Unit:	WH				

Settings are reset to default when the *RST command is sent.(0 Wh)

Response: NR3

Example

```
INP:CUTO:ENER 3.001
```

INP:CUTO:ENER:STAT/ OUTP:CUTO:ENER:STAT

Enables or disables the cutoff function for the integrated power.

Command

```
INPut:CUTOff:ENERgy:STATe <boolean>
INPut:CUTOff:ENERgy:STATe?
OUTPut:CUTOff:ENERgy:STATe <boolean>
OUTPut:CUTOff:ENERgy:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP:CUTO:ENER:STAT ON
```

INP:CUTO:ETIM/ OUTP:CUTO:ETIM

Sets the elapsed time of the cutoff function.

Command

```
INPut:CUTOff:ETIMe[:LEVel] <numeric>
OUTPut:CUTOff:ETIMe[:LEVel] <numeric>
INPut:CUTOff:ETIMe[:LEVel]?
OUTPut:CUTOff:ETIMe[:LEVel]?
```

Parameter

Value: 0 to 3600000
Unit: S

Settings are reset to default when the *RST command is sent.(0 s)

Response: NR3

Example

```
INP:CUTO:ETIM 3
```

INP:CUTO:ETIM:STAT/ OUTP:CUTO:ETIM:STAT

Enables or disables the cutoff function for the elapsed time.

Command

```
INPut:CUTOff:ETIMe:STATe <boolean>
INPut:CUTOff:ETIMe:STATe?
OUTPut:CUTOff:ETIMe:STATe <boolean>
OUTPut:CUTOff:ETIMe:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP:CUTO:ETIM:STAT ON
```

INP:CUTO:VOLT/ OUTP:CUTO:VOLT

Sets the voltage drop of the cutoff function.

Command

```
INPut:CUTOff:VOLTage[:LEVel][:LOWer] <numeric>
OUTPut:CUTOff:VOLTage[:LEVel][:LOWer] <numeric>
INPut:CUTOff:VOLTage[:LEVel][:LOWer]?
OUTPut:CUTOff:VOLTage[:LEVel][:LOWer]?
```

Parameter

Value: 0.00 to 1000.00
Resolution: 0.02
Unit: V

Settings are reset to default when the *RST command is sent.(0 V)

Response: NR3

Example

```
INP:CUTO:VOLT 3.02
```

INP:CUTO:VOLT:STAT/ OUTP:CUTO:VOLT:STAT

Enables or disables the cutoff function for the voltage drop.

Command

```
INPut:CUTOff:VOLTage:STATe <boolean>
INPut:CUTOff:VOLTage:STATe?
OUTPut:CUTOff:VOLTage:STATe <boolean>
OUTPut:CUTOff:VOLTage:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP:CUTO:VOLT:STAT ON
```

INP:EXT:LOG/ OUTP:EXT:LOG

Sets the signal logic for external control input.

Command

```
INPut[:STATe]:EXTernal:LOGic {POSitive|NEGative}
OUTPut[:STATe]:EXTernal:LOGic {POSitive|NEGative}
INPut[:STATe]:EXTernal:LOGic?
OUTPut[:STATe]:EXTernal:LOGic?
```

Parameter

Value:	POSitive	Positive (default)
	NEGative	Negative

Settings are reset to default when the *RST command is sent.

Response: character

Example

```
INP:EXT:LOG NEG
```

INP:MSYN/ OUTP:MSYN

Sends a load-on or load-off sync signal.

Sending this command to any of the synchronized PLZ-5WH2s turns the load on or off simultaneously on the synchronized PLZ-5WH2s.

INP:MSYN:ACC/ OUTP:MSYN:ACC (p. 83) must be set to ON on PLZ-5WH2s that you want to synchronize.

Command

```
INPut:MSYNc[:IMMediate] <boolean>
```

```
OUTPut:MSYNc[:IMMediate] <boolean>
```

Parameter

Value:	ON(1)	Sends a load-on sync signal
	OFF(0)	Sends a load-off sync signal

Example

```
INP:MSYN ON
```

INP:MSYN:ACC/ OUTP:MSYN:ACC

Sets whether to accept load on/off sync signal (INP:MSYN) input from other synchronized devices.

Command

```
INPut:MSYNc:ACCept[:STATe] <boolean>
OUTPut:MSYNc:ACCept[:STATe] <boolean>
INPut:MSYNc:ACCept[:STATe]?
OUTPut:MSYNc:ACCept[:STATe]?
```

Parameter

Value:	ON(1)	Accept
	OFF(0)	Not accept (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
INP:MSYN:ACC ON
```

INP:PON:STAT/ OUTP:PON:STAT

Sets the condition panel setting state when the POWER switch is turned on.

Command

```
INPut:PON:STATe {RST|RCL0|AUTO}
OUTPut:PON:STATe {RST|RCL0|AUTO}
INPut:PON:STATe?
OUTPut:PON:STATe?
```

Parameter

Value:	RST	Start in *RST command transmission state
	RCL0	Start in setup memory 0 condition.
	AUTO	Start with the same settings as when the power was switched off the previous time (but always load off). (default)

Settings are reset to default when the *RST command is sent.

Response: character

Example

```
INP:PON:STAT RST
```

INP:PROT:CLE/ OUTP:PROT:CLE

Clears the protection mode.

Command

```
INPut:PROTection:CLEar
OUTPut:PROTection:CLEar
```

INP:PROT:WDOG/ OUTP:PROT:WDOG

Enables/disables the watchdog protection function.

Command

```
INPut:PROTection:WDOG[:STATe] {ON|OFF|1|0}
OUTPut:PROTection:WDOG[:STATe] {ON|OFF|1|0}
INPut:PROTection:WDOG[:STATe]?
OUTPut:PROTection:WDOG[:STATe]?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Response: NR1

Example

```
INP:PROT:WDOG ON
```

INP:PROT:WDOG:DEL/ OUTP:PROT:WDOG:DEL

Sets the delay time of the watchdog protection function.

Command

```
INPut:PROTection:WDOG:DELay <numeric>
OUTPut:PROTection:WDOG:DELay <numeric>
INPut:PROTection:WDOG:DELay?
OUTPut:PROTection:WDOG:DELay?
```

Parameter

Value: 1 to 3600 (The default value is 60)
Unit: S

Response: NR3

Example

```
INP:PROT:WDOG:DEL 3
```

MEASure/READ/FETCh Command

Queries the measurement results. Each command is different in the following ways.

- MEAS resets a portion of the setting conditions to their default values, starts a new measurement, and then queries for one point of measurement data.
- READ starts a new measurement and queries the measured value after waiting for the completion of a single measurement. If TRIG:ACQ:SOUR is set to IMM, measurement starts immediately. If set to BUS, an error is returned. If set to any other value, measurement starts after a trigger is applied.
- FETC does not wait for the measurement to finish. It queries the most recent measured value when the command is received. If there no measurement values, a query for the measured value is made after waiting for the completion of a single measurement. If TRIG:ACQ:SOUR is set to BUS, an error is returned.

MEAS/ READ/ FETC

Queries the measurement results.

The type of measured values and the order depends on the DATA:FORM (p. 57) setting.

Command

```
MEASure?
READ?
FETCh?
```

Response: NR3

The type of measured values and the order depends on the DATA:FORM setting.

MEAS:CAP/ READ:CAP/ FETC:CAP

Queries the measured current capacity (Ah).

Command

```
MEASure:CAPacity?
READ:CAPacity?
FETCh:CAPacity?
```

Response: NR3
Unit: AH

MEAS:CURR/ READ:CURR/ FETC:CURR

Queries the measured current (A).

Command

```
MEASure:CURRent[:DC]?
READ:CURRent[:DC]?
FETCh:CURRent[:DC]?
```

Response: NR3
Unit: A

MEAS:ENER/ READ:ENER/ FETC:ENER

Queries the measured power capacity (Wh).

Command

```
MEASure:ENERgy?
READ:ENERgy?
FETCh:ENERgy?
```

Response: NR3
Unit: WH

MEAS:ETIM/ READ:ETIM/ FETC:ETIM

Queries the measured elapsed time (s).

Command

```
MEASure:ETIMe?
READ:ETIMe?
FETCh:ETIMe?
```

Response: NR3
Unit: S

MEAS:POW/ READ:POW/ FETC:POW

Queries the measured power (W).

Command

```
MEASure:POWer[:DC]?
READ:POWer[:DC]?
FETCh:POWer[:DC]?
```

Response: NR3
Unit: W

MEAS:VOLT/ READ:VOLT/ FETC:VOLT

Queries the measured voltage (V).

Command

```
MEASure:VOLTage[:DC]?
READ:VOLTage[:DC]?
FETCh:VOLTage[:DC]?
```

Response: NR3
Unit: V

MEM:REC

Recalls settings from preset memories.

Command

```
MEMory:RECall {1|2|3}
```

Parameter

Value:	1	Recalls from preset memory A
	2	Recalls from preset memory B
	3	Recalls from preset memory C

Example

```
MEM:REC 2
```

MEM:SAVE

Saves the settings to ABC preset memories.

Command

```
MEMory:SAVE {1|2|3}
```

Parameter

Value:	1	Save to preset memory A
	2	Save to preset memory B
	3	Save to preset memory C

Example

```
MEM:SAVE 2
```

SENS:APER

Sets the recording time period per measurement. The average over the time period is recorded.

Command

```
SENSe:APERture <numeric>
```

```
SENSe:APERture?
```

Parameter

Value: 0.00001 to 1.00000
Unit: S

Settings are reset to default when the *RST command is sent.(0.1 s)

Response: NR3

Example

```
SENS:APER 0.3
```

ARB:APPL

Enables the specified I-V characteristics.

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:APPLy
```

ARB:CLE

Returns the I-V characteristics to the default setting.

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:CLEar
```

ARB:COUN

Sets the number of data points of the I-V characteristics.
To enable the setting, send ARB:APPL (p. 94).

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:COUNt <numeric>
```

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:COUNt?
```

Parameter

Value: 3 to 100
Unit: S

Settings are reset to default when the *RST command is sent.(3 s)

Response: NR3

Example

```
ARB:COUN 30
```

ARB:DATA

Sets the entire I-V characteristics in binary block data format.

The data of the entire I-V characteristics is set at one time, so you do not need to perform initialization with ARB:CLE and set the number of points with ARB:COUN in advance. The ARB:COUN value is changed by the block data setting.

To enable the setting, send ARB:APPL (p. 94).

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:DATA
    #<length-width><length><data>
```

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:DATA?
```

Parameter <length-width>

Number of digits of <length>

Parameter <length>

Byte length of <data>

Parameter <data>

Voltage and current data is expressed in a binary 4 byte integer.

A value is expressed with four bytes (eight digits). For each point, a pair of voltage and current is indicated.

Each value is in little-endian format, and the unit is written using μV and μA.

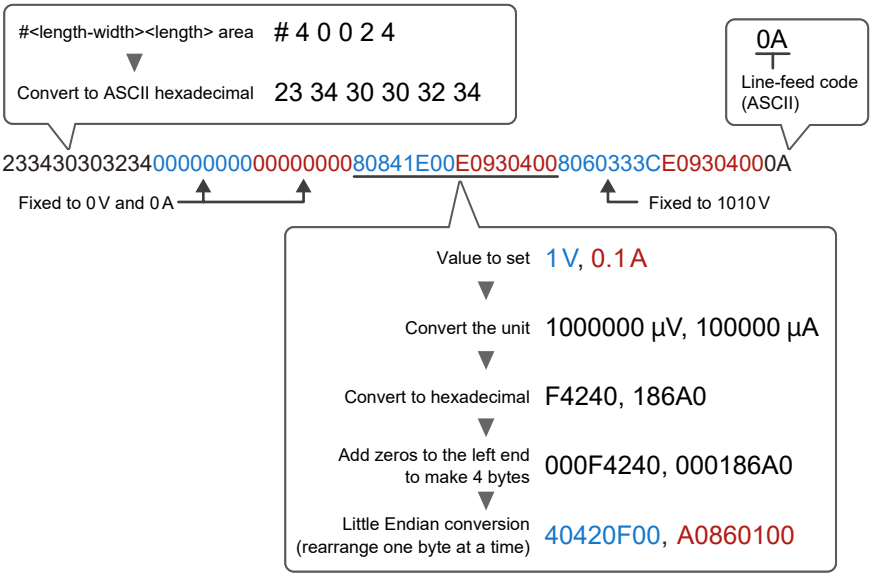
The first voltage and current are fixed to “0V, 0A,” and the last voltage is fixed to “1010V.” If you specify any other values, an error is returned.

Example

If you were to set the combination of “0V, 0A,” “1V, 0.1A,” and “1010V, 0.1A,” the byte length of <data> would be 24. Send the following command in binary. During a binary transfer, we do not recommend concatenating other commands. Terminate with LF.

```
ARB:DATA 23343030323400000000000000000080841E00E09304008060333CE09304000A
```

How to create parameters is explained below.



Response: block

#<length-width><length><data> is returned.

Example

In the case of the combination “0V, 0A,” “1V, 0.1A,” and “1010V, 0.1A,” the following response is returned.

```
233430303234000000000000000080841E00E09304008060333CE09304000A
```

The “233430303234” in the beginning, when converted into hexadecimal ASCII character code every two digits, becomes “#40024.” This indicates that <length-width> is 4, and <length> is 0024.

Related information

“Sample of ARB:DATA” (p. 225)

ARB:MAP

Sets the voltage and current values for the specified parts (point index) of the I-V characteristics.

To enable the setting, send ARB:APPL (p. 94).

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:MAP
   <index>,<voltage>,<current>
```

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:MAP? <index>
```

Parameter <index>

Specifies the point index of the I-V characteristics.

Value: 2 to ARB:COUN (p. 96)

Point index 1 is fixed to 0 V, 0 A. The voltage of the point index of the value specified by ARB:COUN is fixed to 1010.00 V. If you try to change it, an error is returned.

Parameter <voltage>

Voltage of the specified point index

Unit: V

Parameter <current>

Current of the specified point index

Unit: A

Response: NR3

Example

```
ARB:MAP 2,5.3,1.4
```

ARB:MAP:LIST

Queries the entire contents of the I-V characteristics.

Command

```
[SOURce:]ARBitrary[:LEVel][:IMMediate]:MAP:LIST?
```

Response: NR3

Returns the voltage and current values of the data points set with ARB:COUN in comma-delimited NR3 format.

Example

```
<v1>,<c1>,<v2>,<c2>...
```

<v1>	Voltage of 1st point
<c1>	Current of 1st point
<v2>	Voltage of 2nd point
<c2>	Current of 2nd point

ARB:RESP

Sets the response speed for ARB mode.

Command

```
[SOURce:]ARBitrary:RESPonse <numeric>
```

```
[SOURce:]ARBitrary:RESPonse?
```

Parameter

Value: 0 (off), 500u, 1m, 2m, 5m, 10m, 20m, 50m, 100m
Unit: S

Settings are reset to default when the *RST command is sent.(500 μs)

Response: NR3

Example

```
ARB:RESP 20MS
```

SOURce:CONDuctance Command

COND

Sets the conductance in CR mode.

Command

```
[SOURce:]CONDuctance[:LEVel][:IMMediate][:AMPLitude] <numeric>
```

```
[SOURce:]CONDuctance[:LEVel][:IMMediate][:AMPLitude]?
```

Unit: SIE

Settings are reset to default when the *RST command is sent.(0 S)

Response: NR3

Example

```
COND 2
```

COND:EXT:FCON

Enables/disables external control of CR mode.

Command

```
[SOURce:]CONDuatcnce[:LEVel]:EXTernal:FCONtrol <boolean>
```

```
[SOURce:]CONDuatcnce[:LEVel]:EXTernal:FCONtrol?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
COND:EXT:FCON ON
```

COND:PULS:LEV

Sets the conductance of the pulse function.

During pulse function operation, two settings are executed repetitively.

The COND (p. 102) setting value is used as the setting for the first value of the set of two values, and this setting value is applied for the second value of the set of two values.

Command

```
[SOURce:]CONDuctance[:LEVel]:PULSe:LEVel <numeric>
```

```
[SOURce:]CONDuctance[:LEVel]:PULSe:LEVel?
```

Unit: SIE

Settings are reset to default when the *RST command is sent.(0 S)

Response: NR3

Example

```
COND:PULS:LEV 2
```

COND:PULS:FREQ

Sets the frequency of the pulse function.

This frequency is shared during CURRent and CONDuctance pulse function operation.

This command is an alias for CURR:PULS:FREQ.

Command

```
[SOURce:]CONDuctance[:LEVel]:PULSe:FREQuency <numeric>
```

```
[SOURce:]CONDuctance[:LEVel]:PULSe:FREQuency?
```

Unit: Hz

Settings are reset to default when the *RST command is sent.(1 Hz)

Response: NR3

(Reference) The period resolution ⊿T is set to 1 μs in the device. The response returns the inverse of the period set in the device. For example, if you specify 9300 Hz, 108 μs is set in the device. The response in this situation will be +9.25926E+03.

Example

```
COND:PULS:FREQ 3
```

COND:PULS:DCYC

Sets the duty cycle of the pulse function.
This frequency is shared during CURRent and CONDuctance pulse operation.
This command is an alias for CURR:PULS:DCYC.

Command

```
[SOURce:]CONDuctance[:LEVel]:PULSe:DCYCle <numeric>
```

```
[SOURce:]CONDuctance[:LEVel]:PULSe:DCYCle?
```

Unit: PCT
Settings are reset to default when the *RST command is sent.(50 %)

Response: NR3

Example

```
COND:PULS:DCYC 30
```

COND:RANG

Set the range of CR mode.

Command

```
[SOURce:]CONDuctance:RANGe {LOW|HIGH}
```

```
[SOURce:]CONDuctance:RANGe?
```

Parameter

Value:	LOW	L range
	HIGH	H range (default)

Settings are reset to default when the *RST command is sent.

Response: character

Example

```
COND:RANG HIGH
```

COND:RESP

Sets the response speed for CR mode.

Command

```
[SOURce:]CONDuctance:RESPonse {NORMal|FAST}
```

```
[SOURce:]CONDuctance:RESPonse?
```

Parameter

Value:	NORMal	Normal (default)
	FAST	Fast

Settings are reset to default when the *RST command is sent.

Response: character

Example

```
COND:RESP FAST
```

SOURce:CURRent Command

CURR

Sets the current in CC mode.

Command

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <numeric>
```

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
```

Unit: A

Settings are reset to default when the *RST command is sent.(0 A)

Response: NR3

Example

```
CURR 3
```

CURR:EXT:ACON

Enables or disables the function that superimposes current on the current in CC mode using an external voltage.

Command

```
[SOURce:]CURRent[:LEVel]:EXTernal:ACONtrol <boolean>
```

```
[SOURce:]CURRent[:LEVel]:EXTernal:ACONtrol?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
CURR:EXT:ACON ON
```

CURR:EXT:FCON

Enables/disables external control of CC mode.

Command

```
[SOURce:]CURRent[:LEVel]:EXTernal:FCONtrol <boolean>
```

```
[SOURce:]CURRent[:LEVel]:EXTernal:FCONtrol?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
CURR:EXT:FCON ON
```

CURR:PROT

Sets the current value for overcurrent protection (OCP).

Command

```
[SOURce:]CURRent:PROTection[:LEVel][:UPPer] <numeric>
```

```
[SOURce:]CURRent:PROTection[:LEVel][:UPPer]?
```

Unit: A

Settings are reset to default when the *RST command is sent.(MAXimum)

Response: NR3

Example

```
CURR:PROT 15
```

CURR:PROT:STAT

Sets the operation when overcurrent protection (OCP) is activated.

Command

```
[SOURce:]CURRent:PROTection:STATe <boolean>
```

```
[SOURce:]CURRent:PROTection:STATe?
```

Parameter

Value:	ON(1)	Load off (Trip)
	OFF(0)	Limits the current (Limit) (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
CURR:PROT:STAT ON
```

CURR:PULS:LEV

Sets the current of the pulse function.

During pulse function operation, two settings are executed repetitively.

The CURR (p. 109) setting value is used for the first value of the set of two values, and this setting value is applied for the second value of the set of two values.

Command

```
[SOURce:]CURRent[:LEVel]:PULSe:LEVel <numeric>
```

```
[SOURce:]CURRent[:LEVel]:PULSe:LEVel?
```

Unit: A

Settings are reset to default when the *RST command is sent.(0 A)

Response: NR3

Example

```
CURR:PULS:LEV 3
```

CURR:PULS:FREQ

Sets the frequency of the pulse function.

This frequency is shared during CURRent and CONDuctance pulse function operation.

This command is an alias for COND:PULS:FREQ.

Command

```
[SOURce:]CURRent[:LEVel]:PULSe:FREQuency <numeric>
```

```
[SOURce:]CURRent[:LEVel]:PULSe:FREQuency?
```

Parameter

Value:	1.0 to 10.0 k
Resolution:	1 to 10, 0.1 11 to 100, 1 110 to 1000, 10 1.1 k to 10.0 k, 0.1 k
Unit:	HZ

Settings are reset to default when the *RST command is sent.(1 Hz)

Response: NR3

Example

```
CURR:PULS:FREQ 300
```

CURR:PULS:DCYC

Sets the duty cycle of the pulse function.

This frequency is shared during CURRent and CONDuctance pulse function operation.

This command is an alias for COND:PULS:DCYC.

Command

```
[SOURce:]CURRent[:LEVel]:PULSe:DCYCle <numeric>
```

```
[SOURce:]CURRent[:LEVel]:PULSe:DCYCle?
```

Parameter

Value: 5.0 to 95.0
The minimum duty cycle is limited by the minimum pulse time span (20 μs).

Resolution: This varies depending on the frequency specified by CURR:PULS:FREQ.

1 Hz to 1000 Hz, 0.1
1.1 kHz to 10.0 kHz, 1

Unit: PCT

Settings are reset to default when the *RST command is sent.(50 %)

Response: NR3

Example

```
CURR:PULS:DCYC 30
```

CURR:SINE:AMPL

Sets the amplitude of the sine function.

When a sine operation is in progress, the center value of the amplitude is set to the CURR (p. 109) value.

Command

```
[SOURce:]CURRent[:LEVel]:SINE:AMPLitude <numeric>
```

```
[SOURce:]CURRent[:LEVel]:SINE:AMPLitude?
```

Unit: A

Settings are reset to default when the *RST command is sent.(0 A)

Response: NR3

Example

```
CURR:SINE:AMPL 10
```

CURR:SINE:FREQ

Sets the frequency of the sine function.

Command

```
[SOURce:]CURRent[:LEVel]:SINE:FREQuency <numeric>
```

```
[SOURce:]CURRent[:LEVel]:SINE:FREQuency?
```

Parameter

Value: 1 to 1000, 2000, 3000, 10000

Resolution: 1 to 10, 1
21 to 100, 10
200 to 1000, 100

Unit: HZ

Settings are reset to default when the *RST command is sent.(1 Hz)

Response: NR3

(Reference) The period resolution ⊿T is set to 20 μs in the device. The response returns the inverse of the period set in the device. For example, if you specify 900 Hz, 1 120 μs is set in the device. The response in this situation will be +8.92860E+02.

Example

```
CURR:SINE:FREQ 300
```

CURR:SLEW

Set the slew rate value.

Command

```
[SOURce:]CURRent:SLEWrate <numeric>
```

```
[SOURce:]CURRent:SLEWrate?
```

Parameter

	PLZ1005WH2	PLZ2005WH2	PLZ4005WH2	PLZ12005WH2	PLZ20005WH2
Value:	0.001 to 1	0.002 to 2	0.004 to 4	0.01 to 12	0.02 to 20
Resolution:	0.00002	0.00005	0.0001	0.0002	0.0005
Unit:	A/μs				

Settings are reset to default when the *RST command is sent.(MINimum)

Response: NR3

Example

```
CURR:SLEW 0.1
```

CURR:SST

Sets the soft start time.

Command

```
[SOURce:]CURRent:SSTart <numeric>
```

```
[SOURce:]CURRent:SSTart?
```

Parameter

Value: 0 (off), 500 µ, 1 m, 2 m, 5 m, 10 m, 20 m, 50 m, 100 m
Unit: S

Settings are reset to default when the *RST command is sent.(500 µs)

Response: NR3

Example

```
CURR:SST 100MS
```

FUNC

Set the operation mode.

Command

```
[SOURce:]FUNCtion[:MODE] {CC|CR|CV|CP|ARB}
```

```
[SOURce:]FUNCtion[:MODE]?
```

Parameter

Value:	CC	CC mode (default)
	CR	CR mode
	CV	CV mode
	CP	CP mode
	ARB	ARB mode

Settings are reset to default when the *RST command is sent.

Response: character

Example

```
FUNC CV
```

FUNC:CVOP

Enables/disables +CV mode (adds CV mode to CC mode or CR mode).

+CV mode can only be set with an SCPI command. This command is available to maintain compatibility with the PLZ-5W series (PLZ205W/PLZ405W/PLZ1205W).

If +CV mode is enabled, the UVP action (VOLT:PROT:ACT (p. 157)) is set to LIMit. The LIMit value changes according to the changes in the voltage (VOLT (p. 155)) in CV mode.

If +CV mode is disabled, the UVP operation is set to off (disabled) regardless of the TRIP/LIMit setting.

Command

```
[SOURce:]FUNCtion[:MODE]:CVOPtion[:STATe] <boolean>
```

```
[SOURce:]FUNCtion[:MODE]:CVOPtion[:STATe]?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
FUNC:CVOP ON
```

POW

Sets the power in CP mode.

Command

```
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <numeric>
```

```
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?
```

Unit: W

Settings are reset to default when the *RST command is sent.(0 W)

Response: NR3

Example

```
POW 500
```

POW:EXT:FCON

Enables/disables external control of CP mode.

Command

```
[SOURce:]POWer[:LEVel]:EXTernal:FCONtrol <boolean>
```

```
[SOURce:]POWer[:LEVel]:EXTernal:FCONtrol?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
POW:EXT:FCON ON
```

POW:PROT

Sets the power of overpower protection (OPP).

Command

```
[SOURce:]POWer:PROTection[:LEVel][:UPPer] <numeric>
```

```
[SOURce:]POWer:PROTection[:LEVel][:UPPer]?
```

Unit: W

Settings are reset to default when the *RST command is sent.(MAXimum)

Response: NR3

Example

```
POW:PROT 800
```

POW:PROT:STAT

Sets the operation when overpower protection (OPP) is activated.

Command

```
[SOURce:]POWer:PROTection:STATe <boolean>
```

```
[SOURce:]POWer:PROTection:STATe?
```

Parameter

Value:	ON(1)	Load off (Trip)
	OFF(0)	Limits the power (Limit) (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
POW:PROT:STAT ON
```

PROG

Selects or deselects a program.

Command

```
[SOURce:]PROGram[:SELected] "/<String>"
```

```
[SOURce:]PROGram[:SELected]?
```

Parameter

Entering a name of an available program selects that program. Enter a slash at the beginning of the program name.

To clear the program selection, enter "".

Settings are reset to default when the *RST command is sent.(clears selection)

Response: String

Example

```
PROG "/My test program"
```

PROG:CRE

Creates a new program.

The program is created on a file on the file system. Because the location where the program is created depends on the present operation mode, even if a program of the same name is created for example in CC mode and CP mode, the resulting programs will be treated as different programs.

Programs cannot be created when a program is being selected. If this happens, enter PROG "" to clear the program selection.

To edit a program, use PROG to specify the program after creating the program.

Command

```
[SOURce:]PROGram:CREate "/<String>"
```

Parameter

Naming convention: Alphabet characters A-Za-z, numbers 0-9, dot (.), comma (,), parentheses (), brackets [], braces {}, and (&), dollar ($), sharp (#), caret (^), percent (%), equal (=), hyphen (-), plus (+), underscore (_), space ( ), case-sensitive

Up to 255 characters

Enter a slash at the beginning of the program name.

Example:

```
PROG:CRE "/My test program"
```

PROG:CURR:PROT

Sets the current value for overcurrent protection (OCP) in the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection[:LEVel][:UPPer]
    <numeric>
```

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection[:LEVel][:UPPer]?
```

Unit: A

Response: NR3

Example

```
PROG:CURR:PROT 10
```

PROG:CURR:PROT:ACT

Sets the operation to be performed when overcurrent protection (OCP) is activated in the program selected by PROG.

This command is an alias for PROG:CURR:PROT:STAT.

Command

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection:ACTion {TRIP|LIMit}
```

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection:ACTion?
```

Parameter

Value:	TRIP	Load off
	LIMit	Limits the current

Response: Character

Example

```
PROG:CURR:PROT:ACT TRIP
```

PROG:CURR:PROT:STAT

Sets the operation to be performed when overcurrent protection (OCP) is activated in the program selected by PROG.

This command is an alias for PROG:CURR:PROT:ACT.

Command

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection[:TRIP]:STATe
    <boolean>
```

```
[SOURce:]PROGram[:SELected]:CURRent:PROTection[:TRIP]:STATe?
```

Parameter

Value:	ON(1)	Load off (Trip)
	OFF(0)	Limits the current (Limit)

Response: NR1

Example

```
PROG:CURR:PROT:STAT ON
```

PROG:CVOP:LEV

Sets the +CV (addition of CV mode) voltage in the program selected by PROG.

+CV mode can only be set with an SCPI command. This command is available to maintain compatibility with the PLZ-5W series (PLZ205W/PLZ405W/PLZ1205W).

If +CV mode is enabled, the UVP action (PROG:VOLT:PROT:ACT (p. 152)) is set to LIMit. The LIMit value changes according to the changes in the voltage (VOLT (p. 155)) in CV mode.

If +CV mode is disabled, the UVP action is set to TRIP.

Command

```
[SOURce:]PROGram[:SELected]:CVOPtion:LEVel <numeric>
```

```
[SOURce:]PROGram[:SELected]:CVOPtion:LEVel?
```

Unit: V

Response: NR3

Example

```
PROG:CVOP:LEV 300
```

PROG:DEL

Deletes the program of the specified name.

Programs cannot be deleted when a program is being selected. If this happens, enter PROG "" to clear the program selection.

Command

```
[SOURce:]PROGram:DELete "/<String>"
```

Parameter

Enters a created program name. Enter a slash at the beginning of the program name.

Example:

```
PROG:DEL "/My test program"
```

PROG:LIST

Queries the list of all the registered programs.

Command

```
[SOURce:]PROGram:LIST?
```

Response: "string","string"...

Returns "" if no programs are stored.

PROG:LOOP

Sets the number of times that the program selected by PROG will repeat.

Command

```
[SOURce:]PROGram[:SELected]:LOOP {<numeric>|INFinite}
```

```
[SOURce:]PROGram[:SELected]:LOOP?
```

Parameter

Value:	1 to 100000	Repeat count
	INFinite	Infinite

Response: NR1 or INF

Returns 0 when no program has been selected by PROG.

Example

```
PROG:LOOP 3
```

PROG:POW:PROT

Sets the power value for overpower protection (OPP) in the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:POWer:PROTection[:LEVel][:UPPer]
    <numeric>
```

```
[SOURce:]PROGram[:SELected]:POWer:PROTection[:LEVel][:UPPer]?
```

Unit: W

Response: NR3

Example

```
PROG:POW:PROT 800
```

PROG:POW:PROT:ACT

Sets the operation to be performed when overpower protection (OPP) is activated in the program selected by PROG.

This command is an alias for PROG:POW:PROT:STAT.

Command

```
[SOURce:]PROGram[:SELected]:POWer:PROTection:ACTion {TRIP|LIMit}
```

```
[SOURce:]PROGram[:SELected]:POWer:PROTection:ACTion?
```

Parameter

Value:	TRIP	Load off
	LIMit	Limits the power

Response: Character

Example

```
PROG:POW:PROT:ACT TRIP
```

PROG:POW:PROT:STAT

Sets the operation to be performed when overpower protection (OPP) is activated in the program selected by PROG.

This command is an alias for PROG:POW:PROT:ACT.

Command

```
[SOURce:]PROGram[:SELected]:POWer:PROTection[:TRIP]:STATe
    <boolean>
```

```
[SOURce:]PROGram[:SELected]:POWer:PROTection[:TRIP]:STATe?
```

Parameter

Value:	ON(1)	Load off (Trip)
	OFF(0)	Limits the power (Limit)

Response: NR1

Example

```
PROG:POW:PROT:STAT ON
```

PROG:LAST:INP/ PROG:LAST:OUTP

Sets the load state (load on or load off) at the end of the sequence for the program selected by PROG.

To enable the setting, PROG:LAST:STAT (p. 141) must be set to on.

Command

```
{SOURce:}PROGram{:SELected}:LAST:INPut <boolean>
```

```
{SOURce:}PROGram{:SELected}:LAST:OUTPut <boolean>
```

```
{SOURce:}PROGram{:SELected}:LAST:INPut?
```

```
{SOURce:}PROGram{:SELected}:LAST:OUTPut?
```

Parameter

Value:	ON(1)	Load on
	OFF(0)	Load off

Response: NR1

Example

```
PROG:LAST:INP ON
```

PROG:LAST:LEV

Sets the load value at the end of the sequence for the program selected by PROG.
To enable the setting, PROG:LAST:STAT (p. 141) must be set to on.

Command

```
{SOURce:}PROGram{:SELected}:LAST:LEVel <numeric>
```

```
{SOURce:}PROGram{:SELected}:LAST:LEVel?
```

Parameter <numeric>

Unit: A, SIE, V, W

This varies depending on the present operation mode.

Response: NR3

Example

To set the current to 10 A when the present operation mode is CC

```
PROG:LAST:LEV 10
```

PROG:LAST:SLEW

Sets the slew rate at the end of the sequence for the program selected by PROG.
To enable the setting, PROG:LAST:STAT (p. 141) must be set to on.

Command

```
{SOURce:}PROGram{:SELected}:LAST:SLEWrate <numeric>
```

```
{SOURce:}PROGram{:SELected}:LAST:SLEWrate?
```

Parameter <numeric>

	PLZ1005WH2	PLZ2005WH2	PLZ4005WH2	PLZ12005WH2	PLZ20005WH2
Value:	0.001 to 1	0.002 to 2	0.004 to 4	0.01 to 12	0.02 to 20
Resolution:	0.0001	0.0002	0.0004	0.001	0.002
Unit:	A/μs				

Response: NR3

Example

```
PROG:LAST:SLEW 0.1
```

PROG:LAST:STAT

Enables or disables the following command that sets the load state at the end of the sequence for the program selected by PROG.

- PROG:LAST:INP/ PROG:LAST:OUTP (p. 138)
- PROG:LAST:LEV (p. 139)
- PROG:LAST:SLEW (p. 140)

Command

```
{SOURce:}PROGram{:SELected}:LAST:STATe <boolean>
```

```
{SOURce:}PROGram{:SELected}:LAST:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled

Response: NR1

Example

```
PROG:LAST:STAT ON
```

PROG:REN

Changes the name of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:REName "<name>"
```

Parameter

Naming convention: Alphabet characters A-Za-z, numbers 0-9, dot (.), comma (,), parentheses (), brackets [], braces {}, and (&), dollar ($), sharp (#), caret (^), percent (%), equal (=), hyphen (-), plus (+), underscore (_), space ( ), case-sensitive

Up to 255 characters

Example:

```
PROG:REN "Your test program"
```

PROG:SAVE

Saves the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:SAVE
```

PROG:STEP<n>:DWEL

Sets the step execution time at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:DWELl <numeric>
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:DWELl?
```

Parameter <n>

Specifies the step number.

Parameter <numeric>

Value: 50 μ to 3600000
Resolution: 1 μ
Unit: S

Response: NR3

Example

```
PROG:STEP3:DWEL 1800
```

PROG:STEP<n>:INP/ PROG:STEP<n>:OUTP

Sets load on/off at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:INPut <boolean>
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:OUTPut <boolean>
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:INPut?
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:OUTPut?
```

Parameter <n>

Specifies the step number.

Parameter <boolean>

Value:	ON(1)	Load on
	OFF(0)	Load off

Response: NR1

Example

```
PROG:STEP3:INP ON
```

PROG:STEP<n>:LEV

Sets the load value at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:LEVel <numeric>
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:LEVel?
```

Parameter <n>

Specifies the step number.

Parameter <numeric>

Unit: A, SIE, V, W

This varies depending on the present operation mode.

Response: NR3

Example

To set the current to 10 A when the present operation mode is CC

```
PROG:STEP3:LEV 10
```

PROG:STEP<n>:SLEW

Sets the slew rate at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:SLEWrate <numeric>
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:SLEWrate?
```

Parameter <n>

Specifies the step number.

Parameter <numeric>

	PLZ1005WH2	PLZ2005WH2	PLZ4005WH2	PLZ12005WH2	PLZ20005WH2
Value:	0.001 to 1	0.002 to 2	0.004 to 4	0.01 to 12	0.02 to 20
Resolution:	0.0001	0.0002	0.0004	0.001	0.002
Unit:	A/μs				

Response: NR3

Example

```
PROG:STEP3:SLEW 10
```

PROG:STEP<n>:TRAN

Sets the setting transition method at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRANsient {IMMediate|RAMP}
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRANsient?
```

Parameter <n>

Specifies the step number.

Parameter

Value:	IMMediate	Transitions in steps from the value of the previous step
	RAMP	Transitions with a slope from the value of the previous step

Response: Character

Example

```
PROG:STEP3:TRAN RAMP
```

PROG:STEP<n>:TRIG:GEN

Sets whether to perform trigger output at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRIGger:GENerate
    <character>[,<character>,<character>,<character>,<character>]
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRIGger:GENerate?
```

Parameter <n>

Specifies the step number.

Parameter <character>

If NONE is set, it cannot be specified at the same time as other setting values. For all other setting values, up to 5 items can be set.

Value:	NONE	Trigger not generated
	TRIGOUT	Trigger output to TRIGOUT pin
	DIGITAL0	Trigger output to DIGITAL0 pin
	DIGITAL1	Trigger output to DIGITAL1 pin
	DIGITAL2	Trigger output to DIGITAL2 pin (When I/O is OUTPut)
	TALink	Output TALink triggers. If TRIG:ACQ:SOUR (p. 163) is set to TALink, measured values are recorded during the step execution of the sequence.

Response: Character

Example

```
PROG:STEP1:TRIG:GEN TRIGOUT,DIGITAL2,TAL
```

PROG:STEP<n>:TRIG:WAIT

Sets trigger wait at the specified step of the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRIGger:WAIT
    <character>[,<character>,<character>]
```

```
[SOURce:]PROGram[:SELected]:STEP<n>:TRIGger:WAIT?
```

Parameter <n>

Specifies the step number.

Parameter <character>

If IMMediate or BUS is set, it cannot be specified at the same time as other setting values.

You can specify up to 4 items in the case of TRIGIN, DIGITAL2, MSYNc, and CUTOFF.

Value:	IMMediate	Executes the step at the internal timing of the PLZ-5WH2 without waiting for trigger application.
	BUS	Executes the step at the timing of software trigger (*TRG) input. The trigger is applied to both trigger systems: (1) sequence and step (TRANsient) and (2) measure (ACQuire).
	TRIGIN	The next step is executed when a signal is input to pin 9 of the EXT CONT connector of the PLZ-5WH2.
	DIGITAL2	Executes the step when a signal is input to pin 13 of the EXT CONT connector of the PLZ-5WH2 (when SYST:CONF:DIGITAL2:DIR (p. 186) is set to INPut).
	MSYNc	Executes the step when an MSync key on the front panel of the synchronized PLZ-5WH2s is pressed. Or, executes the step when a TRIG:TRAN:MSYN (p. 167) synchronization command packet is received from a PC or the like.
	CUTOFF	Executes next step when a cutoff is activated.

Response: Character

Example

```
PROG:STEP1:TRIG:WAIT DIGITAL2,MSYNc
```

PROG:STEPS:COUN

Sets the number of steps to be registered to the program selected by PROG.

If the number of steps is reduced from the current number of steps, the contents of the remaining steps remain unchanged.

Command

```
[SOURce:]PROGram[:SELected]:STEPS:COUNt <numeric>
```

```
[SOURce:]PROGram[:SELected]:STEPS:COUNt?
```

Parameter

Value: Number of steps 1 to 10000

Response: NR1

Returns 0 when no program has been selected by PROG.

Example

```
PROG:STEPS:COUN 30
```

PROG:VOLT:PROT:ACT

Sets the action to be performed when undervoltage protection (UVP) is activated in the program selected by PROG.

Command

```
[SOURce:]PROGram:VOLTage:PROTection:ACTion {LIMit|TRIP}
```

```
[SOURce:]PROGram:VOLTage:PROTection:ACTion?
```

Parameter

Value:	LIMit	Limits the current.
	TRIP	Turns the load off.

Response: Character

Example

```
PROG:VOLT:PROT:ACT TRIP
```

PROG:VOLT:PROT:LOW

Sets the voltage for undervoltage protection (UVP) in the program selected by PROG.

Command

```
[SOURce:]PROGram[:SELected]:VOLTage:PROTection[:LEVel]:LOWer
    <numeric>
```

```
[SOURce:]PROGram[:SELected]:VOLTage:PROTection[:LEVel]:LOWer?
```

Parameter

Unit: V

Response: NR3

Example

```
PROG:VOLT:PROT:LOW 1
```

PROG:VOLT:PROT:STAT

Enables or disables the undervoltage protection (UVP) in the program selected by PROG.

When enabled, the PLZ-5WH2 turns the load off during UVP operation. You can set the UVP voltage value with PROG:VOLT:PROT:LOW (p. 153).

Command

```
[SOURce:]PROGram[:SELected]:VOLTage:PROTection:STATe <boolean>
```

```
[SOURce:]PROGram[:SELected]:VOLTage:PROTection:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled

Response: NR1

Example

```
PROG:VOLT:PROT:STAT ON
```

VOLT

Sets the voltage in CV mode.

Command

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <numeric>
```

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?
```

Unit: V

Settings are reset to default when the *RST command is sent.(0 V)

Response: NR3

Example

```
VOLT 30
```

VOLT:EXT:FCON

Enables/disables external control of CV mode.

Command

```
[SOURce:]VOLTage[:LEVel]:EXTernal:FCONtrol <boolean>
```

```
[SOURce:]VOLTage[:LEVel]:EXTernal:FCONtrol?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
VOLT:EXT:FCON ON
```

VOLT:PROT:ACT

Sets the action to be performed when undervoltage protection (UVP) is activated.

Command

```
[SOURce:]VOLTage:PROTection:ACTion {LIMit|TRIP}
```

```
[SOURce:]VOLTage:PROTection:ACTion?
```

Parameter

Value:	LIMit	Limits the current. (default)
	TRIP	Turns the load off.

Settings are reset to default when the *RST command is sent.

Response: Character

Example

```
VOLT:PROT:ACT TRIP
```

VOLT:PROT:LOW

Sets the voltage of undervoltage protection (UVP).

Command

```
[SOURce:]VOLTage:PROTection[:LEVel]:LOWer <numeric>
```

```
[SOURce:]VOLTage:PROTection[:LEVel]:LOWer?
```

Unit: V

Settings are reset to default when the *RST command is sent.(0 V)

Response: NR3

Example

```
VOLT:PROT:LOW 1
```

VOLT:PROT:STAT

Enables/disables undervoltage protection (UVP).

When enabled, the load is turned off or the current is limited according to the action set by VOLT:PROT:ACT. You can set the UVP voltage value with VOLT:PROT:LOW.

Command

```
[SOURce:]VOLTage:PROTection:STATe <boolean>
```

```
[SOURce:]VOLTage:PROTection:STATe?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
VOLT:PROT:STAT ON
```

VOLT:RESP

Sets the response speed for CV mode.

Command

```
[SOURce:]VOLTage:RESPonse {NORMal|FAST}
```

```
[SOURce:]VOLTage:RESPonse?
```

Parameter

Value:	NORMal	Normal (default)
	FAST	Fast

Settings are reset to default when the *RST command is sent.

Response: Character

Example

```
VOLT:RESP FAST
```

TRIG:ACQ:COUN

Sets the number of times measurement values are to be recorded.

Command

```
TRIGger:ACQuire:COUNt <numeric>
```

```
TRIGger:ACQuire:COUNt?
```

Parameter

Value: 1 to 65536

Settings are reset to default when the *RST command is sent.(1)

Response: NR1

Example

```
TRIG:ACQ:COUN 3
```

TRIG:ACQ:DEL

Sets the delay time for applying measurement triggers.
The delay time is applied to each trigger application.

Command

```
TRIGger:ACQuire:DELay <numeric>
```

```
TRIGger:ACQuire:DELay?
```

Parameter

Value: 0 to 100
Resolution: 0.1 μ
Unit: S

Settings are reset to default when the *RST command is sent.(0 s)

Response: NR3

Example

```
TRIG:ACQ:DEL 3
```

TRIG:ACQ:MSYN

Sends a measurement sync signal.

Sending this command to any of the synchronized PLZ-5WH2s starts measurements simultaneously on the synchronized PLZ-5WH2s.

TRIG:ACQ:SOUR (p. 163) must be set to MSYNc on PLZ-5WH2s that you want to synchronize.

Command

```
TRIGger:ACQuire:MSYNc[:IMMediate]
```

TRIG:ACQ:SOUR

Clears the measurement trigger weight state and sets the trigger source for starting measurements.

Command

```
TRIGger:ACQuire:SOURce <character>
```

```
TRIGger:ACQuire:SOURce?
```

Parameter

You can select from the available choices below.

Value:	IMMediate	Records measurements when the Initiate key on the PLZ-5WH2 front panel is pressed or when INIT:ACQ (p. 68) is received from a PC or the like.
	BUS	Records the measurement values at the timing of software trigger (*TRG) application. The trigger is applied to both trigger systems: (1) sequence and step (TRANsient) and (2) measure (ACQuire).
	DIGITAL2	Records the measurement values when a signal is input to pin 13 of the EXT CONT connector of the PLZ-5WH2 (when SYST:CONF:DIGITAL2:DIR (p. 186) is set to INPut).
	TALink	If PROG:STEP<n>:TRIG:GEN (p. 149) is set to TALink, measured values are recorded during the step execution of the sequence.
	MSYNc	Executes the step when TRIG:ACQ:MSYN (p. 162) is received from a PC or the like.
	LOADOFF	Starts the measurement when the load is turned off.

Settings are reset to default when the *RST command is sent.(IMMediate)

Response: Character

Example

```
TRIG:ACQ:SOUR DIGITAL2
```

TRIG:ACQ:INT:STAT

Sets whether to measure at intervals when the trigger count is 2 or higher.
Set the trigger count using TRIG:ACQ:COUN (p. 160).
Set the interval using TRIG:ACQ:INT:TIME.

Command

```
TRIGger:ACQuire:INTerval:STATe <boolean>
```

```
TRIGger:ACQuire:INTerval:STATe?
```

Parameter

Value:	ON(1)	Intervals are inserted between measurements.
	OFF(0)	Intervals are not inserted between measurements. (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
TRIG:ACQ:INT:STAT ON
```

TRIG:ACQ:INT:TIME

Sets the measurement interval time when TRIG:ACQ:INT:STAT is set to ON.

Command

```
TRIGger:ACQuire:INTerval:TIME <numeric>
```

```
TRIGger:ACQuire:INTerval:TIME?
```

Parameter

Value:	10 μ to 3600
Resolution:	10 μ
Unit:	S

Settings are reset to default when the *RST command is sent.(0.1 s)

Response: NR3

Example

```
TRIG:ACQ:INT:TIME 3
```

TRIG:TRAN:DEL

Sets the delay time for applying sequence triggers.

Command

```
TRIGger:TRANsient:DELay <numeric>
```

```
TRIGger:TRANsient:DELay?
```

Parameter

Value: 0 to 100
Resolution: 10 μ
Unit: S

Settings are reset to default when the *RST command is sent.(0 s)

Response: NR3

Example

```
TRIG:TRAN:DEL 3
```

TRIG:TRAN:EXEC

Queries the execution state of the sequence function, pulse function, and sine function.

Command

```
TRIGger:TRANsient:EXECution[:STATe]?
```

Response:

<IDLE|WTG|RUN|SUSP|PULS|SINE>, <Step_NR1>, <Loop_NR1> is returned.

Response	Description
<IDLE|WTG|RUN|SUSP|PULS|SINE>	IDLE: Sequence waiting and the pulse function is off WTG: Trigger wait state SUSP: Sequence paused RUN: Delay time or step in execution PULS: Pulse operation in progress SINE: Sine operation in progress
<Step_NR1>	Number of steps of program
<Loop_NR1>	Number of loops of program

TRIG:TRAN:MSYN

Sends a sequence sync signal.

If synchronized PLZ-5WH2s are in a sequence start trigger wait state or step start trigger wait state, sending this command to any of the PLZ-5WH2s starts the sequence or step simultaneously.

The trigger source (TRIG:TRAN:SOUR (p. 168) or PROG:STEP<n>:TRIG:WAIT (p. 150)) must be set to MSYNc on PLZ-5WH2s that you want to synchronize.

Command

```
TRIGger:TRANsient:MSYNc[:IMMediate]
```

TRIG:TRAN:SOUR

Clears the sequence trigger weight state and sets the trigger source for starting sequences.

Command

```
TRIGger:TRANsient:SOURce <character>
```

```
TRIGger:TRANsient:SOURce?
```

Parameter

You can select from the available choices below.

Value:	IMMediate	Starts the sequence when the Initiate key on the PLZ-5WH2 front panel is pressed or when INIT:TRAN:PROG (p. 71) is received from a PC or the like.
	BUS	Starts the sequence at the timing of software trigger (*TRG) application. The trigger is applied to both trigger systems: (1) sequence and step (TRANsient) and (2) measure (ACQuire).
	DIGITAL2	Starts the sequence when a signal is input to pin 13 of the EXT CONT connector of the PLZ-5WH2 (when SYST:CONF:DIGITAL2:DIR (p. 186) is set to INPut).
	MSYNc	Starts the sequence when an MSync key on the front panel of the synchronized PLZ-5WH2s is pressed. Or, starts the sequence when TRIG:TRAN:MSYN (p. 167) is received from a PC or the like.

Settings are reset to default when the *RST command is sent.(IMMediate)

Response: Character

Example

```
TRIG:TRAN:SOUR DIGITAL2
```

TRIG:TRAN:SUSP

Pauses the running sequence trigger function.

This function forcibly pauses the execution time (PROG:STEP<n>:DWEL (p. 144)) of the currently running program step.

Command

```
TRIGger:TRANsient:SUSPend
```

TRIG:TRAN:RES

Resumes the paused sequence from the point at which it was paused.

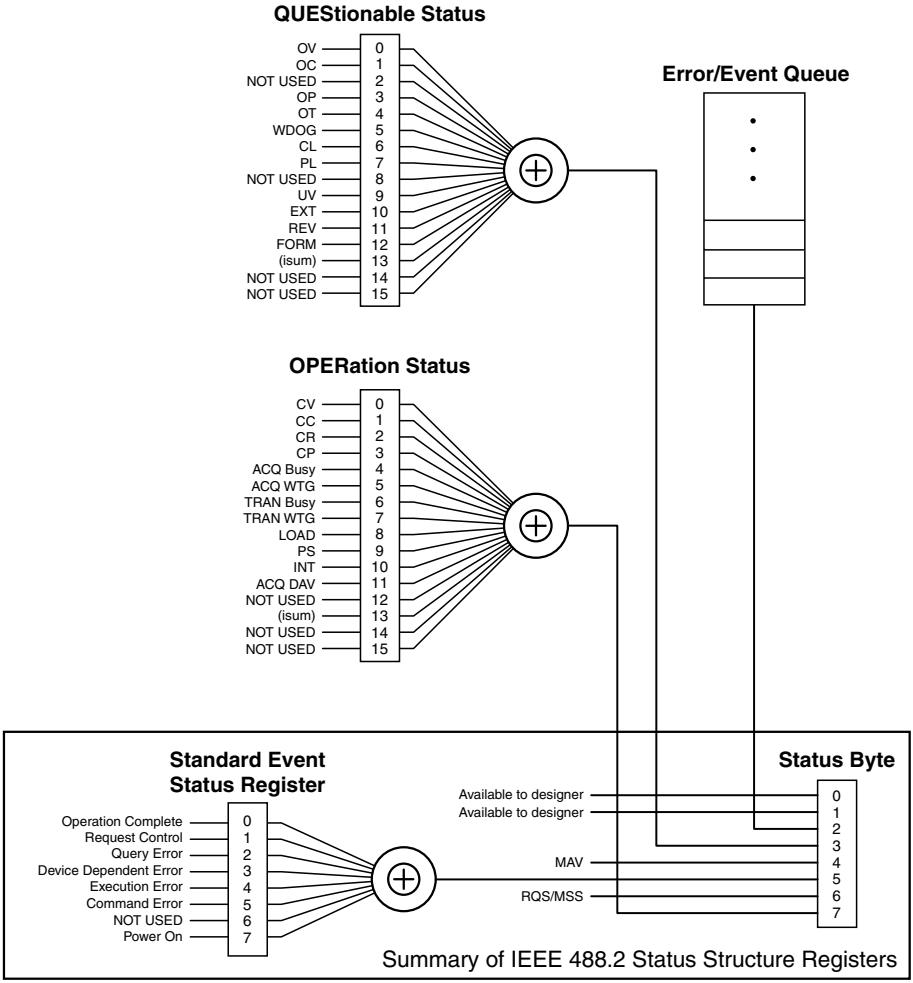
Command

```
TRIGger:TRANsient:RESume
```

Register structure

A "+" represents the logical OR of the register bits.

1999 SCPI Syntax & Style



Summary of IEEE 488.2 Status Structure Registers

Architecture

IEEE 488.2 and SCPI registers are used for status reports.

In each SCPI status register, there are the following sub registers: the CONDition register, the EVENt register, the ENABle register, the PTRansition filter, and the NTRansition filter.

CONDition register

Transitions of the CONDition register are automatic and reflect the condition of the product in real time. Reading this register does not affect its contents.

EVENt register

The EVENt register bits are automatically set according to the changes in the CONDition register. The rule for setting the bits varies depending on the positive and negative transition registers (PTRansition and NTRansition). The EVENt register is reset when it is read.

ENABle register

The ENABle register enables reports to the summary bit or status bit of the event bits.

Transition filters

Use the PTRansition (positive transition) filter to report events when the condition changes from false to true.

Use the NTRansition (negative transition) filter to report events when the condition changes from true to false.

If both the positive filter and negative filter are set to true, events can be reported each time the status changes.

If both filters are cleared, event reporting is disabled.

Status byte register

The status byte register stores STB and RQS (MSS) messages as defined by the IEEE 488.1 standard. The status byte register can be read by using IEEE 488.1 serial polling or the IEEE 488.2 common command *STB?.

When the controller executes serial polling, bit 6 responds with request service (RQS). The status byte value is not changed by serial polling.

*STB? makes the device transmit the contents of the status byte register and the master status summary (MSS) message.

*STB? does not change the status byte, MSS, and RQS.

Bit	Bit weight	Bit name	Description
0	1	Reserved	Reserved for future use by IEEE 488. The bit value is notified as zero.
1	2	Reserved	
2	4	Error/Event Queue	If data exists in the error or event queue, this bit is set to true.
3	8	Questionable Status Register (QUES)	This bit is set to true when a bit is set in the QUEStionable event status register and the corresponding bit in the QUEStionable status enable register is true.
4	16	Message Available (MAV)	This bit is set to true when a request is received from the digital programming interface and the product is ready to generate the data byte.
5	32	Standard Event Status Bit Summary (ESB)	This bit is set to true when a bit is set in the event status register.
6	64	Request Service (RQS)	This bit is set to true when a bit is set in the service request enable register and the corresponding bit exists in the status byte. The SRQ line of the GPIB is set.
		Master Status Summary (MSS)	This bit is set to true when any bit in the status byte register is set to 1 and the corresponding bit in the service request enable register is set to 1.
7	128	Operation Status Register (OPER)	This bit is set to true when a bit is set in the OPERation event status register and the corresponding bit in the OPERation status enable register is set.
8-15		Not Used	—

Event status register

The event status register bits are set when certain events occur during product operation. All the event status register bits are set by the error event queue.

This register is defined by the IEEE 488.2 standard and is controlled using the IEEE 488.2 common commands *ESE, *ESE?, and *ESR?.

You can check the error content with SYST:ERR?.

Bit	Bit weight	Bit name	Description	Error number
0	1	Operation Complete(OPC)	Set when an *OPC command is received and all operations in standby have been completed.	-800 to -899
1	2	Request Control (RQC)	—	--
2	4	Query Error(QYE)	Set when an attempt is made to read data from the output queue when there is no data or when the output queue is not in the wait state. This indicates that there is no data in the output queue.	-400 to -499
3	8	Device Dependent Error(DDE)	Set when there is a device-specific error.	-300 to -399 100 to 999
4	16	Execution Error(EXE)	Set when the product evaluates that the program data after the header is outside the formal input range or does not match the specifications of the product. This indicates that a valid SCPI command may not be executed correctly depending on the state of the product.	-200 to -299
5	32	Command Error(CME)	Set when an IEEE 488.2 syntax error is detected by the parser, when an unidentifiable header is received, or when a group execution trigger enters the internal IEEE 488.2 SCPI command input buffer.	-100 to -199
6	64	User Request(URQ)	—	--
7	128	Power On(PON)	—	--
8-15		Reserved	—	--

OPERation status register

The OPERation status register is a 16-bit register that contains information about the normal operating conditions of the product.

Bit	Bit weight	Bit name	Description
0	1	CV	Running in CV mode
1	2	CC	Running in CC mode
2	4	CR	Running in CR mode
3	8	CP	Running in CP mode
4	16	ACQ BUSY	ACQuire sequence execution in progress
5	32	ACQ WTG	ACQuire sequence waiting for trigger (TRIG)
6	64	TRAN Busy	TRANsient sequence execution in progress
7	128	TRAN WTG	TRANsient sequence waiting for trigger (TRIG)
8	256	LOAD	Load is currently on
9	512	PS	The sequence is paused.
10	1024	INT	Integration in progress
11	2048	ACQ DAV	Data to be acquired is valid (acquirable)
12	4096	NOT USED	—
13	8192	NOT USED	—
14	16384	NOT USED	—
15	32768	NOT USED	Always 0.

STAT:OPER

Queries the event of the OPERation status register.
A query clears the contents of the register.

Command

```
STATus:OPERation[:EVENt]?
```

Response: NR1

STAT:OPER:COND

Queries the condition of the OPERation status register.
A query does not clear the contents of the register.

Command

```
STATus:OPERation:CONDition?
```

Response: NR1

STAT:OPER:ENAB

Sets the enable register of the OPERation status register.

Command

```
STATus:OPERation:ENABle <NRf>
STATus:OPERation:ENABle?
```

Parameter
Value: 0 to 32767

Response: NR1

STAT:OPER:NTR

Sets the negative transition filter of the OPERation status register.

Command

```
STATus:OPERation:NTRansition <NRf>

STATus:OPERation:NTRansition?
```

Parameter

Value: 0 to 32767

Response: NR1

STAT:OPER:PTR

Sets the positive transition filter of the OPERation status register.

Command

```
STATus:OPERation:PTRansition <NRf>

STATus:OPERation:PTRansition?
```

Parameter

Value: 0 to 32767

Response: NR1

QUEStionable status register

The QUEStionable status register is a 16-bit register that stores information related to the following product's status and events.

The QUEStionable status register bits may indicate that there are problems with the product's measured data.

Bit	Bit weight	Bit name	Description
0	1	OV(Over Voltage Protection)	Overvoltage protection activated
1	2	OC(Over Current Protection)	Overcurrent protection activated
2	4	NOT USED	—
3	8	OP(Over Power protection)	Overpower protection activated
4	16	OT(Over Temperature Protection)	Overheat protection activated
5	32	WDOG(Watchdog protection)	Watchdog protection activated
6	64	CL	Current limiting in progress
7	128	PL	Current limiting in progress
8	256	NOT USED	—
9	512	UV(Under Voltage Protection)	Undervoltage protection activated
10	1024	EXT	External error occurred
11	2048	REV	Reverse voltage detected
12	4096	FORM	Connection error occurred
13	8192	NOT USED	—
14	16384	NOT USED	—
15	32768	RESERVED	Always 0.

STAT:QUES

Queries the event of the QUEStionable status register.

A query clears the contents of the register.

Command

```
STATus:QUEStionable[:EVENt]?
```

Response: NR1

STAT:QUES:COND

Queries the condition of the QUEStionable status register.
A query does not clear the contents of the register.

Command

```
STATus:QUEStionable:CONDition?
```

Response: NR1

STAT:QUES:ENAB

Sets the enable register of the QUEStionable status register.

Command

```
STATus:QUEStionable:ENABle <NRf>
STATus:QUEStionable:ENABle?
```

Parameter
Value: 0 to 32767

Response: NR1

STAT:QUES:NTR

Sets the negative transition filter of the QUEStionable status register.

Command

```
STATus:QUEStionable:NTRansition <NRf>
STATus:QUEStionable:NTRansition?
```

Parameter
Value: 0 to 32767

Response: NR1

STAT:QUES:PTR

Sets the positive transition filter of the QUEStionable status register.

Command

```
STATus:QUEStionable:PTRansition <NRf>
```

```
STATus:QUEStionable:PTRansition?
```

Parameter

Value: 0 to 32767

Response: NR1

Preset status

STAT:PRES

Resets the ENABle, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values.

Default values:

STATus:ENABle = 0x0000

STATus:PTRansition = 0x7FFF

STATus:NTRansition = 0x0000

Command

```
STATus:PRESet
```

SYST:BEEP

Turns all buzzers on and off collectively.

Command

```
SYSTem:BEEPer[:ALL][:STATe] <boolean>
```

```
SYSTem:BEEPer[:ALL][:STATe]?
```

Parameter

Value:	ON(1)	On (default)
	OFF(0)	Off

The setting may change when a SYST:BEEP:KEY, SYST:BEEP:PROT, or SYST:BEEP:SCPI is received.

Settings are reset to default when the *RST command is sent.

Response: NR1

Returns 1 when any buzzer is on when an invalid operation is executed, a protection function is activated, or an SCPI error occurs. Returns 0 when all settings are off.

Example

```
SYST:BEEP OFF
```

SYST:BEEP:KEY

Sets the buzzer sound on/off in case of invalid operation.

Command

```
SYSTem:BEEPer:KEY[:STATe] <boolean>
```

```
SYSTem:BEEPer:KEY[:STATe]?
```

Parameter

Value:	ON(1)	On (default)
	OFF(0)	Off

The setting may change when a SYST:BEEP is received.
Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:BEEP:KEY ON
```

SYST:BEEP:PROT

Turns on or off the buzzer that sounds when a protection function is activated.

Command

```
SYSTem:BEEPer:PROTection[:STATe] <boolean>
```

```
SYSTem:BEEPer:PROTection[:STATe]?
```

Parameter

Value:	ON(1)	On (default)
	OFF(0)	Off

The setting may change when a SYST:BEEP is received.
Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:BEEP:PROT ON
```

SYST:BEEP:SCPI

Turns on or off the buzzer that sounds when an SCPI error occurs.

Command

```
SYSTem:BEEPer:SCPI[:STATe] <boolean>
```

```
SYSTem:BEEPer:SCPI[:STATe]?
```

Parameter

Value:	ON(1)	On (default)
	OFF(0)	Off

The setting may change when a SYST:BEEP is received.
Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:BEEP:SCPI ON
```

SYST:COMM:RLST

Switches the PLZ-5WH2 to local or remote mode.

When set to REMote, pressing the front panel ESCAPE key causes the product to return to LOCal. When set to REMote or RWLock, REMOTE appears on the front panel display.

Command

```
SYSTem:COMMunicate:RLSTate <character>
```

```
SYSTem:COMMunicate:RLSTate?
```

Parameter

Value:	LOCal	Sets the product to local mode (Remote Disable; the RMT turns off). This enables both panel operations and commands. This is a substitute command for IEEE488.1 ren FALSE (Remote Disable).
	REMote	Switches the product to remote mode. All panel keys, except the ESCAPE key, are locked. This is a substitute command for IEEE 488.1 ren (Remote Enable). This is also the substitute command for address specification.
	RWLock	Switches the product to remote mode. All panel keys are locked. This is a substitute command for IEEE 488.1 llo (Local Lock Out).

Response: Character

Example

```
SYST:COMM:RLST REM
```

SYST:CONF:DIGITAL2:DIR

Sets the I/O direction of the DIGITAL2 signal.

Command

```
SYSTem:CONFigure:DIGITAL2:DIRection {INPut|OUTPut}
```

```
SYSTem:CONFigure:DIGITAL2:DIRection?
```

Parameter

Value:	INPut	Input (default)
	OUTPut	Output

Settings are reset to default when the *RST command is sent.

Response: Character

Example

```
SYST:CONF:DIGITAL2:DIR OUTPut
```

SYST:CONF:MSYN:IND

Queries whether synchronization connection is enabled or disabled.

Command

```
SYSTem:CONFigure:MSYNc:INDex?
```

Response: NR1

SYST:CONF:RSEN

Sets remote sensing function on/off.

Command

```
SYSTem:CONFigure:RSENsing[:STATe] <boolean>
```

```
SYSTem:CONFigure:RSENsing[:STATe]?
```

Parameter

Value:	ON(1)	On
	OFF(0)	Off (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:CONF:RSEN ON
```

SYST:DATE

Sets the date.
Also set the time (SYST:TIME (p. 198)).

Command

```
SYSTem:DATE <year_NR1>,<month_NR1>,<day_NR1>
```

```
SYSTem:DATE?
```

Parameter <year_NR1>
Value 2016 to 2037 (year)

Parameter <month_NR1>
Value 1 to 12 (month)

Parameter <day_NR1>

Value 1 to 31 (day)

Response
Returns the year, month, and day in a comma-separated NR1 format.

Example

```
SYST:DATE 2019,4,25
```

SYST:ERR

Reads the oldest error information or event information from the error queue.

Errors that have been read are cleared. The error/event queue can hold up to 16 errors.“Error checking” (p. 214)

The error queue is cleared if a *CLS command is sent.

Command

```
SYSTem:ERRor[:NEXT]?
```

Response

Returns the oldest error or event from the error/event queue in the following format, in response to SYST:ERR?.

Example: If there is no error or event

This command returns +0 "No error."

Example: If a command that cannot be executed in the present operating state is received

This command returns -221, “Settings conflict.”

For details on errors, see the Command error reference.

“List of errors” (p. 220)

SYST:ERR:COUN

Returns the number of unread errors in the error queue.

Command

```
SYSTem:ERRor:COUNt?
```

Response: NR1

SYST:KLOC

Sets or releases panel control lock.
This is invalid when the SYST:COMM:RLST (p. 185) is set to RWL.

Command

```
SYSTem:KLOCk <boolean>
```

```
SYSTem:KLOCk?
```

Parameter

Value:	ON(1)	Set the panel control lock
	OFF(0)	Release the panel control lock

Response: NR1

Example

```
SYSTem:KLOC ON
```

SYST:KLOC:LEV

Sets the panel control lock level.

Command

```
SYSTem:KLOCk:LEVel <NRf>
```

```
SYSTem:KLOCk:LEVel?
```

Parameter

Value:	1	Low
	2	Medium
	3	High (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:KLOC:LEV 3
```

SYST:PASS

Enables the execution of the command that returns the product to its factory default settings.

To return the settings back to factory default, send SYST:SEC:IMM (p. 196).

Command

```
SYSTem:PASSword[:CENable] "<string>"
```

Parameter

Value: Password set by SYST:PASS:NEW (p. 194)

Response: String

Example

```
SYST:PASS "my_password"
```

SYST:PASS:CDIS

Disables the execution of the command that returns the settings back to factory default.

Command

```
SYSTem:PASSword:CDISable "<string>"
```

Parameter

Value: Password set by SYST:PASS:NEW (p. 194)

Example

```
SYST:PASS:CDIS "my_password"
```

SYST:PASS:NEW

Sets the password.

Enable or disable password protection using SYST:PASS (p. 193).

Command

```
SYSTem:PASSword:NEW "<string_exist>","<string_new>"
```

Parameter "<string_exist>"

Enter the present password. Because no password is set by default, enter <"">.

Parameter "<string_new>"

Enter an arbitrary password under the following conditions.

Naming convention:	alphanumeric characters (A-Z, a-z, 0-9), underscore, hyphen
Number of characters:	4 to 15

The factory default password is "".

Example

```
SYST:PASS:NEW "existing password", "new password"
```

```
SYST:PASS:NEW "", "new password"
```

SYST:PASS:STAT

Queries whether the product is in a state that allows the execution of the command that returns the product to its factory default settings.

Command

```
SYSTem:PASSword[:CENable]:STATe?
```

Response: NR1

When the command can be executed, 1 is returned.

SYST:SEC:IMM

Deletes all the user data other than the calibration data and present time and returns the product to its factory default settings.

To use this command, you need to send SYST:PASS (p. 193) in advance.

Command

```
SYSTem:SECurity:IMMediate
```

SYST:SSAV

Enables or disables the screen saver.
Use SYST:SSAV:DEL to set the time until the screen saver starts.

Command

```
SYSTem:SSAVer[:STATe] <boolean>
```

```
SYSTem:SSAVer[:STATe]?
```

Parameter

Value:	ON(1)	Enabled
	OFF(0)	Disabled (default)

Settings are reset to default when the *RST command is sent.

Response: NR1

Example

```
SYST:SSAV ON
```

SYST:SSAV:DEL

Sets the time until the screen saver starts.
This command is valid when SYST:SSAV is set to ON.

Command

```
SYSTem:SSAVer:DELay <numeric>
```

```
SYSTem:SSAVer:DELay?
```

Parameter

Value:	60 to 59940
Unit:	S

Settings are reset to default when the *RST command is sent.(60 s)

Response: NR3

Example

```
SYST:SSAV:DEL 3600S
```

SYST:TIME

Sets the time.
Also set the date (SYST:DATE (p. 189)).

Command

```
SYSTem:TIME <hour_NR1>,<min_NR1>,<sec_NR1>
```

```
SYSTem:TIME?
```

Parameter <hour_NR1>
Value: 0 to 23 (hour)

Parameter <min_NR1>
Value: 0 to 59 (minute)

Parameter <sec_NR1>

Value: 0 to 59 (second)

Response
Returns the hour, minute, and second in NR1 format.

Example

```
SYST:TIME 23,0,0
```

SYST:TIME:ADJ

Automatically synchronizes the system clock using the NTP server on the network.

Command

```
SYSTem:TIME:ADJust
```

SYST:TZON

Sets the time zone of the system clock.
Use SYST:TZON:CAT? to check the time zone ID.

Command

```
SYSTem:TZONe "<string>"
```

```
SYSTem:TZONe?
```

Parameter

Value: Time zone ID or UTC (The default value is “UTC”)

Response: String

Example

```
SYST:TZON "Asia/Tokyo"
```

SYST:TZON:CAT

Queries the time zone IDs that can be used.

Command

```
SYSTem:TZONe:CATalog?
```

Response: Comma-separated character string

SYST:VERS

Queries the version of the SCPI specifications that the product complies with.

Command

```
SYSTem:VERSion?
```

Response

1999.0 is returned.

Basic configuration

Using the following load conditions as an example, this section explains how to write commands starting from basic configuration up to turning the load on.

■ Load conditions

Operation mode:	CC mode
Current value:	30 A
Slew rate:	3 A/μs
Pulse:	Pulse level 10 A Frequency 10 Hz Duty cycle 50 %
Overcurrent protection:	Activated at 40 A. The load is turned off when activated.

■ Command examples

```
FUNCtion CC    'Set the operation mode to CC mode
CURRent 30    'Set the current to 30 A
CURRent:SLEWrate 3    'Set the slew rate to 3A/μs
CURRent:PULSe:LEVel 10    'Set the pulse level to 10A
CURRent:PULSe:FREQuency 10    'Set the frequency to 10Hz
CURRent:PULSe:DCYCle 50    'Set the duty cycle to 50%
CURRent:PROTection 40    'Set OCP to 40A
CURRent:PROTection:STATe ON    'Turn off the load when OCP is activated
INITiate:PULSe    'Enable the pulse function
INPut ON    'Turn the load on
```

Checking the current's minimum/maximum values

The current's minimum/maximum setting values depend on the set current range. You can query the present minimum/maximum setting value by sending a query to which the MINimum or MAXimum parameter is added, as shown below.

```
CURRent? MINimum
CURRent? MAXimum
```

Taking measurements

This tutorial explains how to take measurements and query the results.

■ Taking a measurement and querying the result

In the following command example, after the load is turned on, INIT:ACQ is entered to immediately take a measurement, and then the measurement is queried.

```
TRIGger:ACQuire:SOURce IMM  'Set the measurement trigger source to IMMediate.
INPut ON     'Turn the load on.
INITiate:ACQuire     'Start measurement.
FETCh?       'Query the measured value.
```

In the line 3, the product switches to measurement trigger wait state, but since the trigger source is set to IMMediate in line 1, measurement starts immediately.

■ Taking several measurements and querying the results

In the following command example, after the load is turned on, INIT:ACQ is entered to immediately take three measurements, and then the measurement is queried.

```
TRIGger:ACQuire:SOURce IMM  'Set the measurement trigger source to IMMediate.
TRIGger:ACQuire:COUNt 3     'Set the measurement count to 3.
INPut ON     'Turn the load on.
INITiate:ACQuire     'Start measurement.
DATA:R?      'Query the measured value.
```

In the line 4, the product switches to measurement trigger wait state, but since the trigger source is set to IMMediate in line 1, measurement starts immediately. Since the measurement count is set to 3 in line 2, measurement is taken three times. When several measurements are taken, use DATA:R? (line 5) to query the measured values.

For the differences in the methods of querying measured values, see “Querying measured values”.

■ Starting a measurement with a trigger and querying the results

You can use a trigger to take measurements by setting the measurement trigger source (TRIG:ACQ:SOUR) to anything other than IMMediate. Here, an example of starting a measurement with a DIGITAL2 input after the load is turned on and querying the measured values is explained.

```
TRIGger:ACQuire:SOURce DIGITAL2 'Set the trigger source to DIGITAL2.

INPut ON     'Turn the load on.

INITiate:ACQuire     'Switch to measurement trigger wait state.

                     '(start a measurement with DIGITAL2 input).

FETCh?       'Query the measured value.
```

The product switches to measurement trigger wait state in line 3. Because the trigger source was set to DIGITAL2 in line 1, when a signal is received through pin 13 of the EXT CONT connector when the product is in the trigger wait state, measurement starts.

Overview of Trigger Function

The product allows you to control the start timing of measurement and sequences using triggers. In the trigger wait state, a measurement or a sequence can be started when an event selected beforehand (trigger source) matches an event either inside or outside the product (trigger).

■ Trigger systems

There are two trigger systems.

TRANsient	A sequence or step starts using trigger detection as the starting point.
ACQuire	Measurement starts using trigger detection as the starting point.

■ Trigger source

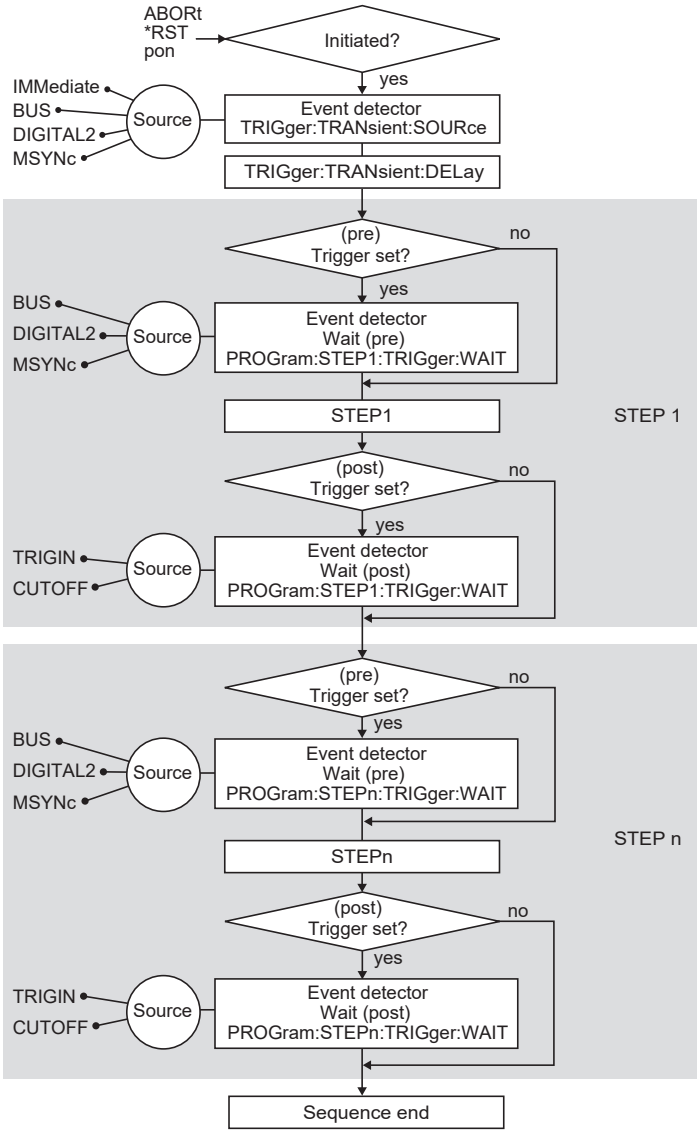
The following trigger sources are supported.

IMMediate	Execution starts at the internal timing of the product without waiting for trigger application. The delay until the start of execution is reduced to a minimum.
BUS	Execution starts when a software trigger (*TRG) is input. Software triggers are applied to both the TRANsient and ACQuire trigger systems.
DIGITAL2	Execution starts when a signal is input to pin 13 of the EXT CONT connector. (when SYST: CONF: DIGITAL2: DIR is set to INPut)
TALink (measurement only)	If PROG:STEP<n>:TRIG:GEN is set to TALink, measured values are recorded during the step execution of the sequence.
MSYNc	Execution starts when a signal is received from an synchronized external equipment or when a TRIG:ACQ:MSYN or TRIG:TRAN:MSYN sync signal is received from a PC or the like.
TRIGIN (step execution only)	The next step is executed when a signal is input to pin 9 of the EXT CONT connector.
CUTOFF (step execution only)	Executes next step when a cutoff is activated.
LOADOFF	Starts the measurement when the load is turned off.

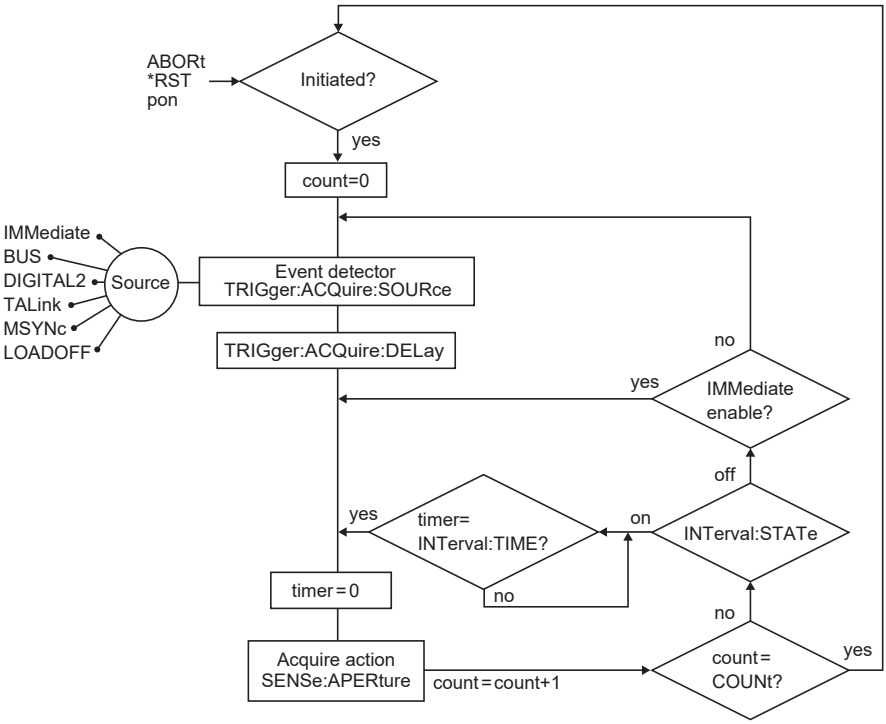
■ Trigger processing procedure

The trigger processing procedure is illustrated in the following figures.

Sequence trigger system



Measurement trigger system



Sequence operations

A sequence consists of programs and steps. A program is a collection of steps. Steps are executed in order one at a time, starting from step 1. Upon completion of the last step of a program, execution of that program has been completed once. The value at the end of the sequence will be the value of the last step.

You can create up to 30 programs in the product. You can use a combined total of 10000 steps for all the programs. For an overview of the sequence functions, see the user's manual contained in the accompanying CD-ROM.

In this example, we will create a sequence that works under the following conditions in constant current (CC) mode.

—Note—

Do not create more than 10000 steps. If the total number steps exceeds 10000, unexpected behavior may result.

■ Operating condition

Operation mode: CC

Program1

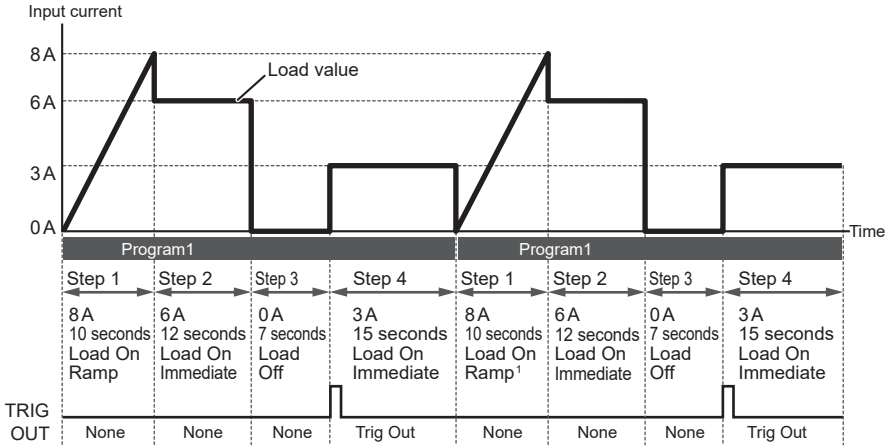
Program name: PROGRAM1, number of loops: 2

Run	Load value	Execution time	Load	Transition method	Trigger output
Step 1	8A	10 seconds	On	Ramp	None
Step 2	6A	12 seconds	On	Immediate	None
Step 3	0A	7 seconds	Off	—	None
Step 4	3A	15 seconds	On	Immediate	Trig Out

Explanation

Program 1 executes steps 1 to 4 and ends the first execution. When Program1 is executed one more time, this sequence is over.

Representing this sequence in graph form looks as follows.



1. If Load of step 1 is set to Ramp and the program is repeated, the start current of step 1 will always be 0 A.

■ Example of commands used to create a sequence

```
FUNC CC   'Set the operation mode to CC mode
PROG:CRE "/PROGRAM1"  'Create PROGRAM1
PROG "/PROGRAM1"      'Select PROGRAM1
PROG:LOOP 2     'Set the number of loops of PROGRAM1 to 2
PROG:STEPS:COUN 4     'Set the number of steps of PROGRAM1 to 4
PROG:STEP1:LEV 8A     'Set the setting value of Step 1 to 8 A
PROG:STEP1:DWEL 10S   'Set the execution time of Step 1 to 10 seconds
PROG:STEP1:INP ON     'Set Step 1 to load on
PROG:STEP1:TRAN RAMP  'Set Step 1 to Ramp transition
PROG:STEP2:LEV 6A     'Set the setting value of Step 2 to 6 A
PROG:STEP2:DWEL 12S   'Set the execution time of Step 2 to 12 seconds
PROG:STEP2:INP ON     'Set Step 2 to load on
PROG:STEP2:TRAN IMM   'Set Step 2 to Immediate transition
PROG:STEP3:DWEL 7S    'Set the execution time of Step 3 to 7 seconds
PROG:STEP3:INP OFF    'Set Step 3 to load off
PROG:STEP4:LEV 3A     'Set the setting value of Step 4 to 3 A
PROG:STEP4:DWEL 15S   'Set the execution time of Step 4 to 15 seconds
PROG:STEP4:INP ON     'Set Step 4 to load on
PROG:STEP4:TRAN IMM   'Set Step 4 to Immediate transition
PROG:STEP4:TRIG:GEN TRIGOUT 'Set the trigger output of Step 4 to Trigger out
```

■ Executing and stopping a sequence

Send the following commands to execute a sequence immediately.

```
TRIG:TRAN:SOUR IMM   'Set the trigger source to IMM.
PROG "/PROGRAM1"     'Select PROGRAM1
INIT:TRAN:PROG       'Start PROGRAM1
```

To stop a sequence that is being executed, send the following command.

```
ABOR
```

■ Executing a sequence using triggers

You can use triggers to synchronize sequence operation.

For example, if you are using a software trigger to execute PROGRAM1, set the trigger source to BUS.

```
TRIG:TRAN:SOUR BUS   'Set the trigger source to BUS.

PROG "/PROGRAM1"     'Select PROGRAM1

INIT:TRAN:PROG       'Enter wait state for sequence execution trigger

*TRG  'Sends a software trigger and starts the sequence
```

INIT:TRAN:PROG pulls the TRIGger subsystem out of the IDLE state and places it in the trigger function start (initiated) state.

At that time, if TRIG:TRAN:SOUR is IMM, the sequence is immediately executed by the selected program.

If TRIG:TRAN:SOUR is other than IMM, the state is the trigger wait (WTG) state, and when the source signal set with TRIG:TRAN:SOUR is sent, the sequence is executed by the selected program.

When the sequence completes or is stopped, the TRIGger subsystem enters the IDLE state again.

Waiting for operation complete

When the *OPC command is transmitted, the product sets the OPC bit (bit 0) of the event status register to TRUE (1) when the processing of all commands in standby has been completed. This information can be determined by checking the OPC bit (bit 0) of the *ESR? query.

Example: Starting a new measurement and transmitting the *OPC command

When the Event Status Enable Register is set to 1 and the Service Request Enable Register is set to 32, an SRQ (Service Request) is generated upon measurement completion. The SRQ function cannot be used if you are using the RS232 and SCPI-ROW interfaces.

```
*ESE 1;*SRE 32;*CLS;:INIT:ACQ;*OPC
```

<A service request is generated upon completion of measurement>

When the *OPC? request is used instead of the *OPC command, the product sets response data “1” (in NR1 format) in the output queue upon completion of measurement if there are no other operations waiting to be completed.

```
INIT:ACQ;*OPC?
```

<Read response upon measurement completion>

When IEEE488 sdc/dcl, *RST, or *RCL is transmitted when the power is turned on, the product sets the OPC bit (bit 0) to TRUE (1), and the response data in the output queue to “1” (in NR1 format).

Status monitoring

The product has two mandatory SCPI standard registers, STATus:OPERation and STATus:QUEStionable, in addition to the IEEE488.2 standard registers.

■ Register basics

All SCPI registers have a standard architecture that uses events/filters. CONDition, EVENt, and ENABle and optionally PTRansition and NTRansition can be used. CONDition and EVENt are read-only registers working as status indicators. ENABle, PTRansition and NTRansition are read-write registers working as event and summary filters.

■ STATus:OPERation

The OPERation Status register is used to record events and notifications that occur during normal operations.

For example, to check whether CV mode is in use, check the CV bit (bit 0) of the STATus:OPERation register.

```
STATus:OPERation?    'Checks whether the CV bit is set.
```

■ STATus:QUEStionable

The QUEStionable Status register is used to record events and notifications that occur during abnormal operations.

To check whether the overvoltage protection function has been activated, check the OV bit (bit 0) of the STATus:QUEStionable register.

```
STATus:QUEStionable?    'Checks whether the OV bit is set.
```

Error checking

■ Error/event queue

The SCPI specifications define a standard error reporting scheme, Error/Event Queue. This is a FIFO (First In First Out) queue, which records errors and events. The maximum number of errors/events that the product can record is 16. Each error/event can be read with the SYSTem:ERRor query.

```
SYSTem:ERRor?
```

The response to this query contains a numeric part (error/event number) and a textual description, such as:

```
-222,"Data out of range"
```

The error/event queue becomes empty when the *CLS common command is sent, when the last item in the queue is read, and when the product is turned on. When the error/event queue is empty, the query returns the following:

```
+0,"No error"
```

■ Displaying communication errors

When an SCPI error occurs, the error is shown on the display when SYSTEM and then Error are pressed on the front panel.

This is convenient for debugging remote control.

Visual Basic 2017

■ Configuring a project

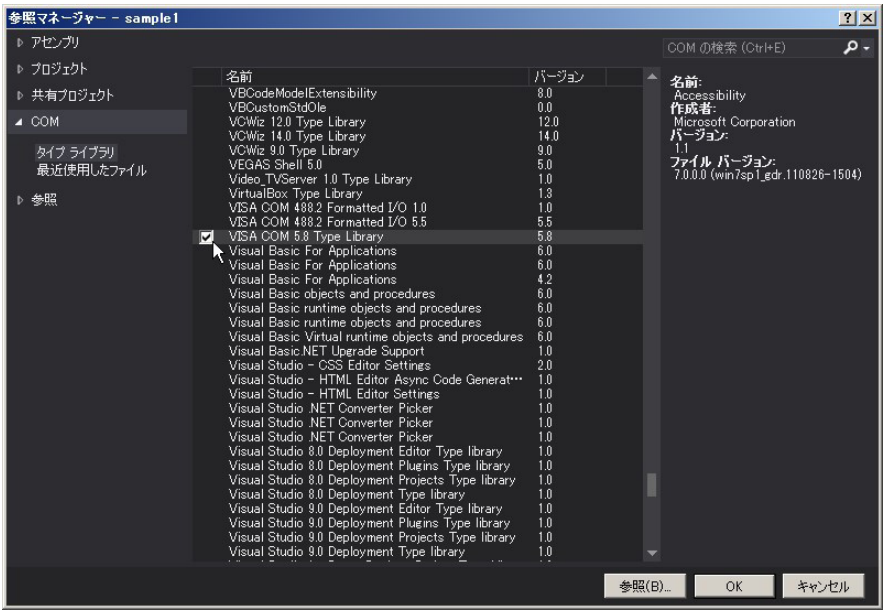
First, add the communication middleware (VISA library) to the project.

Click References on the Project menu to open the Reference Manager window.

On the navigation pane, click COM and then Type Libraries.

From the list in the center of the window, select “VISA COM *.* Type Library” (where *.* is the VISA library version number), and select the check box.

Click OK to close the dialog box.



■ Communicating via RS232C, USB, LAN

Opening VISA

Before you can use the VISA library to communicate with RS232C, USB, and LAN devices, you have to open VISA. Specify an I/O resource to open VISA.

Example: Opening VISA when using USB on the PLZ-5WH2

```
Set rm = CreateObject("VISA.GlobalRM")

Set msg = rm.Open("USB::0x0B3E::0x1042::00000001::INSTR", NO_LOCK, 0, "")
```

“USB::0x0B3E::0x1042::00000001::INSTR” is an I/O resource.

The I/O resource syntax is shown below. The parts surrounded by square brackets ([ ]) can be omitted. Enter the appropriate values in the parts written in italics.

Serial (RS232C)		ASRL[*board*][::INSTR] Example: A measuring instrument connected to serial port COM1 ASRL1::INSTR
USB		USB[*board*]::*VendorID*::*ProductID*::*SerialNumber*[::*InterfaceNumber*][::INSTR] Example: A USBTMC measuring instrument whose vendor ID (VID) is 2878, product ID (PID) is 4162, and serial number is 00000001 USB0::0x0B3E::0x1042::00000001::INSTR
LAN[1]	VXI-11	TCPIP[*board*]::*hostname*[::inst*0*][::INSTR] Example: Measuring instrument whose IP address (hostname) is 169.254.7.8 TCPIP::169.254.7.8::INSTR You can also specify the host name for the hostname parameter.
	HiSLIP	TCPIP[*board*]::*hostname*::hislip0[::INSTR] Example: Measuring instrument whose IP address (hostname) is 169.254.7.8 TCPIP::169.254.7.8::hislip0::INSTR You can also specify the host name for the hostname parameter.
	SCPI-RAW	TCPIP[*board*]::*hostname*::*portno*::SOCKET Example: Measuring instrument whose IP address (hostname) is 169.254.7.8 (the product’s port number is fixed to 5025) TCPIP::169.254.7.8::5025::SOCKET You can also specify the host name for the hostname parameter.

1. The hostname must be a valid mDNS hostname (a Bonjour hostname that ends in ".local") or a DNS hostname that is managed by an external DNS server (a full-qualified domain name—FQDN). If you are using an mDNS hostname, Apple Bonjour (alternatively, iTunes or Safari) must be installed on your PC.

In VISA, you can use aliases for I/O resources.

If you use an alias for an I/O resource, even if the alias name is hard-coded in the application, the I/O resource name can still be changed to an appropriate value when the application runs.

Example: Using an alias (MYDEV1) for an I/O resource

```
Set msg = rm.Open("MYDEV1", NO_LOCK, 0, "")
```

When you use aliases, specify the actual I/O resources through an external configuration table or similar tool. Refer to the VISA manual.

Controlling the instrument

Next, we will use commands such as read and write commands to control the instrument. You must include line-feed codes in the command strings.

Examples:

```
msg.WriteString("CURRent 0.3" & vbLf) 'Set the current to 0.3 A
```

```
msg.WriteString("CURRent:PULSe:FREQuency 10" & vbLf) 'Set the
    pulse frequency to 10 Hz
```

```
msg.WriteString("INPut ON" & vbLf) 'Turn the load on
```

Closing VISA

Finally, close VISA.

In a sequence of operations, you only have to open and close VISA once.

```
msg.Close
```

■ Sample program

```
Imports Ivi.Visa.Interop

Public Class Form1
    Dim rm As IResourceManager3 = New ResourceManager()
    Dim msg As IMessage

    'Open the VISA resource.
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        'When using serial port COM1
        msg = rm.Open("ASRL1::INSTR", AccessMode.NO_LOCK, 0, "")
        'When using USB
        'msg = rm.Open("USB0::0x0B3E::0x1042::AB012345::INSTR", AccessMode.NO_LOCK, 0, "")
        'When using LAN (VXI-11)
        'msg = rm.Open("TCPIP0::192.168.1.23::inst0::INSTR", AccessMode.NO_LOCK, 0, "")
        'When using a VISA alias
        'msg = rm.Open("MYDEV1", AccessMode.NO_LOCK, 0, "")
        msg.TerminationCharacterEnabled = True

        'When using a serial port, set the parameters with the following code.
        Dim seri As ISerial = msg
        seri.BaudRate = 19200
        seri.DataBits = 8
        seri.StopBits = SerialStopBits.ASRL_STOP_ONE
        seri.Parity = SerialParity.ASRL_PAR_NONE
        seri.FlowControl = SerialFlowControl.ASRL_FLOW_NONE
    End Sub

    'Query the ID
    Private Sub CommandIdn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CommandIdn.Click
        msg.WriteString("SYST:COMM:RLST REM" & vbLf)
        msg.WriteString("*IDN?" & vbLf)
        TextBox1.Text = msg.ReadString(256)
        msg.WriteString("*RST" & vbLf)
    End Sub

    'Set the current and pulse function.
    Private Sub CommandCurr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CommandCurr.Click
        msg.WriteString("*RST" & vbLf)
        msg.WriteString("FUNCtion CC" & vbLf)
        msg.WriteString("CURRent 0.3" & vbLf)
        msg.WriteString("CURRent:SLEWrate 0.1" & vbLf)
        msg.WriteString("CURRent:PULSe:LEVel 0.15" & vbLf)
        msg.WriteString("CURRent:PULSe:FREQuency 10" & vbLf)
        msg.WriteString("CURRent:PULSe:DCYCle 10" & vbLf)
        msg.WriteString("POWer:PROTection  30" & vbLf)
```

```
    msg.WriteString("POWer:PROTection:STATe ON" & vbLf)
    msg.WriteString("INITiate:PULSe" & vbLf)
    msg.WriteString("INPut ON" & vbLf)
End Sub

'Editing the program
Private Sub CommandProg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CommandProg.Click
    msg.WriteString("PROG """" & vbLf)
    msg.WriteString("PROG:CRE ""/Program1"" & vbLf)
    msg.WriteString("PROG ""/Program1"" & vbLf)
    msg.WriteString("PROG:LOOP 2" & vbLf)
    msg.WriteString("PROG:STEPS:COUN 4" & vbLf)
    msg.WriteString("PROG:STEP1:LEV 8" & vbLf)
    msg.WriteString("PROG:STEP2:LEV 4" & vbLf)
    msg.WriteString("PROG:STEP3:LEV 0" & vbLf)
    msg.WriteString("PROG:STEP4:LEV 3" & vbLf)
    msg.WriteString("PROG:STEP1:DWEL 10" & vbLf)
    msg.WriteString("PROG:STEP2:DWEL 12" & vbLf)
    msg.WriteString("PROG:STEP3:DWEL 7" & vbLf)
    msg.WriteString("PROG:STEP4:DWEL 15" & vbLf)
    msg.WriteString("PROG:STEP1:INP ON" & vbLf)
    msg.WriteString("PROG:STEP2:INP ON" & vbLf)
    msg.WriteString("PROG:STEP3:INP OFF" & vbLf)
    msg.WriteString("PROG:STEP4:INP ON" & vbLf)
    msg.WriteString("PROG:STEP1:TRAN RAMP" & vbLf)
    msg.WriteString("PROG:STEP2:TRAN IMM" & vbLf)
    msg.WriteString("PROG:STEP4:TRAN IMM" & vbLf)
    msg.WriteString("PROG:STEP1:TRIG:GEN NONE" & vbLf)
    msg.WriteString("PROG:STEP2:TRIG:GEN NONE" & vbLf
    msg.WriteString("PROG:STEP3:TRIG:GEN NONE" & vbLf)
    msg.WriteString("PROG:STEP4:TRIG:GEN TRIGOUT" & vbLf)
    msg.WriteString("PROG:SAVE" & vbLf)
    msg.WriteString("INIT:TRAN:PROG" & vbLf)
End Sub

'Queries the measured current
Private Sub CommandMeas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CommandMeas.Click
    msg.WriteString("MEAS:CURR?" & vbLf)
    TextBox2.Text = msg.ReadString(256)
End Sub

'Close the VISA resource.
Private Sub Form1_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Disposed
    msg.Close()
End Sub
```

List of errors

■ Command errors

An error in the range [-199, -100] indicates that an IEEE 488.2 syntax error has been detected by the instrument's parser. The occurrence of any error in this class causes the Command Error bit (bit 5) in the event status register to be set.

Error code		Error message description
-101	Invalid character	An invalid character exists. A data element different than those allowed was recognized.
-102	Syntax error	Syntax error. An unrecognized command or data type was encountered.
-103	Invalid separator	Invalid separator. The parser was expecting a separator and encountered an illegal character.
-108	Parameter not allowed	Parameter not allowed. More parameters were received than expected for the header.
-109	Missing parameter	Missing parameter. Fewer parameters were received than required for the header.
-110	Command header error	Command header error. An error was detected in the header.
-112	Program mnemonic too long	Mnemonic too long. The number of characters in the command header exceeds 12 characters.
-113	Undefined header	Undefined header. Inappropriate for the product.
-114	Header suffix out of range	Invalid suffix exists in the header.
-115	Unexpected number of parameters	Unexpected parameters were received in the header.
-120	Numeric data error	Numeric data error. Generated when parsing a data element which appears to be numeric, including the non-decimal numeric types.
-128	Numeric data not allowed	Numeric data is not allowed.
-131	Invalid suffix	A suffix is invalid. The suffix does not follow the syntax, or the suffix is inappropriate for the product.
-138	Suffix not allowed	A suffix was encountered after a numeric parameter that does not allow suffixes.
-140	Character data error	Character data error. Generated when parsing a character data element.
-141	Invalid character data	Either the character data element contains an invalid character, or the element is not valid.
-144	Character data too Long	Character data too long. The character data element contains too many characters.
-150	String data error	String data error. Generated when parsing a string data element.
-151	Invalid string data	Invalid string data.
-158	String data not allowed	String data is not allowed.
-161	Invalid block data	The block data is invalid.

■ Execution errors

An error in the range [-299, -200] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class causes the Execution Error bit (bit 4) in the event status register to be set.

Error code		Error message description
-203	Command protected	Password-protected program or query command cannot be executed.
-211	Trigger ignored	A trigger was received but ignored.
-213	Init ignored	A measurement initiate operation was ignored because measurement is in progress.
-214	Trigger deadlock	A deadlock occurred because a query was received before the software trigger.
-220	Parameter error	Invalid parameter.
-221	Settings conflict	A command was received that the product cannot execute in its present condition.
-222	Data out of range	Parameter was out of range.
-224	Illegal parameter value	Received invalid parameter data.
-230	Data corrupt or stale	Received a data query before the measurement completed.
-241	Hardware missing	Cannot be executed because the optional hardware is not installed.

■ Product-specific errors

An error in the range [-399, -300] indicates that an error other than command error, query error, or execution error was detected. The occurrence of any error in this class causes the Device Dependent Error bit (bit 3) in the event status register to be set.

Error code		Error message description
-310	System error	System error.
-311	Memory error	Memory error.
-313	Calibration memory lost	The calibration data has been lost.
-314	Save/recall memory lost	The data that was saved with the *SAV command has been lost.
-315	Configuration memory lost	The system settings data has been lost.
-330	Self-test failed	An error occurred during the self-test.
-350	Queue overflow	The error could not be placed in the queue because the queue is full.
-363	Input buffer overrun	Buffer overflow error. This is an error when using RS232C.

■ Query errors

An error in the range [-499, -400] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol described in IEEE 488.2, chapter 6. The occurrence of any error in this class causes the Query Error bit (bit 2) in the event status register to be set.

Error code		Error message description
-410	Query INTERRUPTED	Received a new command after the query was received and before the response was read.
-420	Query UNTERMINATED	The controller attempted to read the response after the device received an unsupported query or did not received a query. The “-100 COMMAND ERROR” and this error are stored in the error queue. The controller will time out.
-430	Query DEADLOCKED	The error queue, input buffer, and output buffer are full when sending large binary data as a response, and the transmission timing is off.

■ Product-dependent errors

The occurrence of any error in this class causes the Device Dependent Error bit (bit 3) in the event status register to be set.

Configuration conflict errors and configuration change rejection errors

These errors occur when the specified configuration changes cannot be permitted.

Error code		Error message description
+101	Setting conflicts due to PROTection state	Cannot be set because a protection function is activated.
+102	Setting conflicts while INPut is ON	Cannot be set because the load is on.
+103	Setting conflicts while INPut:SHORt is ON	Cannot be set because the short function is in operation.
+104	Setting conflicts while PROGram is running	Cannot be set because a program is running.
+105	Setting conflicts while PULSe is running	Cannot be set because pulse operation is in progress.
+106	Setting conflicts due to invalid FUNCtion:MODE	Cannot be set because an invalid mode is set.
+107	Setting conflicts while in CVOPtion is OFF	Cannot be set because the +CV function is disabled.
+108	Wrong password given	The password is not correct.
+109	Illegal password format	The password format is not correct.
+110	Mutual-sync is not established	Mutual-sync is not established.
+121	ARBitrary map index out of range	The number of data points of the arbitrary I-V characteristics is out of range.
+122	ARBitrary map voltage out of range	The voltage of the arbitrary I-V characteristics is out of range.
+123	ARBitrary map current out of range	The current of the arbitrary I-V characteristics is out of range.
+124	ARBitrary map data conflict	The arbitrary I-V characteristics is invalid.
+125	Too less ARBitrary map data	The number of data of the arbitrary I-V characteristics is too low.
+126	Too many ARBitrary map data	The number of data of the arbitrary I-V characteristics is too large.

Out of range setting errors

These errors occur when invalid or incorrect settings are specified.

Error code		Error message description
+201	Illegal buffer size due to not in 2^N	The buffer size is not 2^N.
+202	Same item is specified more than once	The same item is specified multiple times.
+203	NONE is invalid in multiple item settings	NONE is invalid if multiple items are set.
+204	IMMediate is invalid in multiple item settings	IMMediate is invalid if multiple items are set.
+205	BUS is invalid in multiple item settings	Connections for parallel operation are illegal.

Auto test execution and trigger function execution errors

Error code		Error message description
+301	Illegal PROGram name	The program name is illegal.
+302	PROGram name already exists	The program name is already used.
+303	PROGram not found	The program could not be found.
+304	PROGram not selected	No program is selected.
+305	Selected PROGram cannot be deleted	The selected program cannot be deleted.
+306	Recursive PROGram specified	A recursive program has been specified.
+307	STEP number out of range	The number of steps is too large.
+308	Execution error due to non-existing PROGram specified	An execution error occurred because you specified a non-existing program.
+309	Execution error due to value range conflict	An execution error occurred due to a value range conflict.
+310	TRANsient not operating	The TRANsient function is not operating.
+311	TRANsient not suspending	The TRANsient function is not paused.
+313	PROGram cannot be created during selection	Programs cannot be created when a program is being selected.

Sample of ARB:DATA

This is a sample program that sets ARB:DATA on Visual Basic 2017.

A programming example in which the voltage-current pairs are (0 V, 0 A), (1 V, 0.1 A), (1010 V, 0.1 A) are provided below. The first voltage and current are fixed to “0V, 0A,” and the last voltage is fixed to “1010V.” If you specify any other values, an error is returned.

■ Setting ARB:DATA

```
Private Sub CommandArbSet_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CommandArbSet.Click

  'Data you want to set in the ARB table
  Dim Dbl_Volt() As Double = {0, 1, 1010}
  Dim Dbl_Curr() As Double = {0, 0.1, 0.1}

  'Set the unit to µ order
  Dim Int_Data(5) As Integer
  Dim Int_a As Integer = 0
  For Int_b As Integer = 0 To 2
  Int_Data(Int_a) = CInt(Dbl_Volt(Int_b) * 1000000)
  Int_a = Int_a + 1
  Int_Data(Int_a) = CInt(Dbl_Curr(Int_b) * 1000000)
  Int_a = Int_a + 1
  Next

  Dim Byte_Data(41) As Byte
  'Convert the command and the LENGTH portion to byte type
  Dim Byte_Comm() As Byte = System.Text.Encoding.ASCII.GetBytes("ARB:DATA #40024")
  Dim Int_Count As Integer = 0

  For Int_Count = 0 To 14
  'Insert in front of the byte sequence to be transmitted
  Byte_Data(Int_Count) = Byte_Comm(Int_Count)
  Next

  Dim Byte_VoltCurr(4) As Byte
  For Int_m As Integer = 0 To 5
    'Convert the data you want to set in the ARB table to byte type in order
    Byte_VoltCurr = BitConverter.GetBytes(Int_Data(Int_m))
    For Int_n As Integer = 0 To 3
    'Fill in the byte sequence data to be transmitted in order
    Byte_Data(Int_Count) = Byte_VoltCurr(Int_n)
    Int_Count = Int_Count + 1
    Next
  Next
```

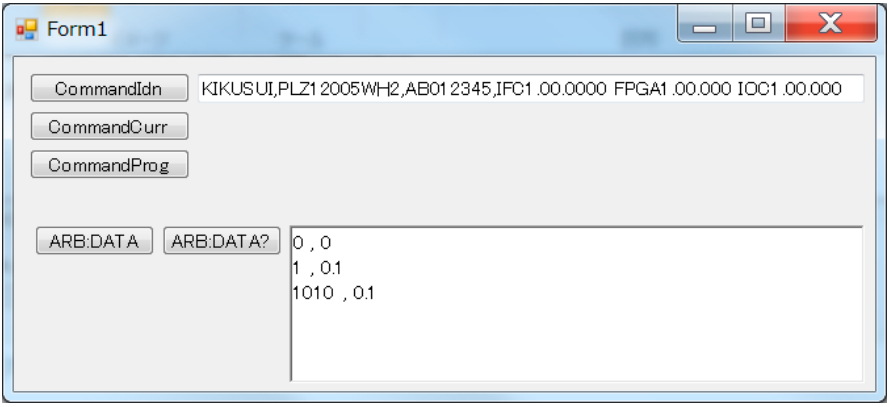
```
    'Insert a line feed at the end of the byte sequence
    Byte_Data(Int_Count + 1) = CByte(&HA)


    'Write in the PLZ12005WH2
    msg.Write(Byte_Data, 41)

    'Apply the ARB data
    msg.WriteString("ARB:APPL" & vbLf)
End Sub
```

■ Querying ARB:DATA

```
    Dim rm As IResourceManager3 = New ResourceManager()

    Dim msg As IMessage

    Dim seri As ISerial
Private Sub ARB_DATA_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.
Click

    'Initialize the RichTextBox that will display the ARB table values
    RichText.Clear()

    'Check the number of data pairs in the ARB table
    msg.WriteString("ARB:COUN?" & vbLf)

    Dim Int_ArbCoun As Integer = CInt(msg.ReadString(32))

    'Calculate the total number of bytes to receive
    Dim Int_ByteLength As Integer = Int_ArbCoun * 8 + 7


    'Disable the terminator
    msg.TerminationCharacterEnabled = False

    'To use 232C, set this also
    seri.EndIn = SerialEndConst.ASRL_END_NONE

    'Write a query command in the PLZ12005WH2
    msg.WriteString("ARB:DATA?" & vbLf)

    'Read the response from the PLZ12005WH2
    Dim Byte_ArbRead() As Byte = msg.Read(Int_ByteLength)


    'Calculate the number of data entries from the <Length> value and prepare a box
    Dim Int_Volt(Int_ArbCoun) As Integer

    Dim Int_Curr(Int_ArbCoun) As Integer

    Dim Dbl_Volt(Int_ArbCoun) As Double

    Dim Dbl_Curr(Int_ArbCoun) As Double

    Dim Int_Count As Integer = 6

    For Int_a = 0 To Int_ArbCoun - 1

        'Convert the byte type data into integers in order from the top
        Int_Volt(Int_a) = BitConverter.ToInt32(Byte_ArbRead, Int_Count)

        'Convert the unit from µV order to V order
        Dbl_Volt(Int_a) = Int_Volt(Int_a) / 1000000

        Int_Count = Int_Count + 4

        'Convert the byte type data into integers in order from the top
        Int_Curr(Int_a) = BitConverter.ToInt32(Byte_ArbRead, Int_Count)
```

```
        'Convert the unit from µA order to A order
        Dbl_Curr(Int_a) = Int_Curr(Int_a) / 1000000
        Int_Count = Int_Count + 4
        'Write the ARB data as voltage-current pairs
        RichText.Text = RichText.Text & CStr(Dbl_Volt(Int_a)) & " , " & CStr(Dbl_Curr(Int_a)) & vbCrLf
    Next

    'Set the Terminator back to enabled
    msg.TerminationCharacterEnabled = True
    'Set the value back to its default
    seri.EndIn = SerialEndConst.ASRL_END_TERMCHAR
End Sub
```

The program execution result is as follows.



Command Processing time

A certain amount of time is required before the commands shown in the following table are received by the product.

The processing times indicated here are standard values. They are not guaranteed.

The processing times vary depending on the settings and the measurement conditions.

The values shown below do not include hardware response times.

Commands	Processing time (ms)						
	USB	RS232C[1]	LAN			GPIB[2]	
			VXI-11	HiSLIP	SCPI-RAW	RS232C[3]	USB[4]
INP/OUTP	5.02	8.36	5.26	5.64	2.44	11.3	52.6
CURR	5.18	7.22	7.16	6.92	3.64	8.10	51.6
INIT:TRAN:PROG	3.54	3.32	12.4	8.22	2.66	12.1	51.4
MEAS?[5]	122	148	122	122	133	168	168

1. Baud rate setting: 115200 bps
2. When the EOI is used as the command transmission completion trigger.
3. When a connection to the product is made via RS232C using the optional GPIB converter (PIA5100) and the bitrate is set to 19200 bps.
4. When a connection to the product is made via USB using the optional GPIB converter (PIA5100).
5. When SENS:APER is set to 0.1 seconds and a query is made for the current, voltage, or power.

If you find any misplaced or missing pages in the manuals, they will be replaced. If the manual gets lost or soiled, a new copy can be provided for a fee. In either case, please contact your Kikusui agent or distributor. At that time, inform your agent or distributor of the “Part No.” written on the front cover of this manual.

Every effort has been made to ensure the accuracy of this manual. However, if you have any questions or find any errors or omissions, please contact your Kikusui agent or distributor.

After you have finished reading this manual, store it so that you can use it for reference at any time.

KIKUSUI ELECTRONICS CORP.

1-1-3 Higashiyamata, Tsuzuki-ku, Yokohama,
224-0023, Japan

Phone: +81-45-482-6353
Facsimile: +81-45-482-6261

www.kikusui.co.jp/en/