KIKUSUI

Part No. IB034212
Oct 2019

Communication Interface Manual

Regurated DC Power Supply PWR-01 Series

400W model

PWR401L PWR401MH
PWR401ML PWR401H

800W model

PWR801L PWR801MH
PWR801ML PWR801H

1200W model

PWR1201L PWR1201MH
PWR1201ML PWR1201H

2000W model

PWR2001L
PWR2001ML

Contents

Command List

*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, serial number, and firmware version of the PWR-01.

*LRN

Queries the command that can restore the current panel settings.

*OPC

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

*OPT

Queries the option that are installed in the PWR-01.

*PSC

Sets whether to clear the event status enable register and the service request enable register when the POWER switch is turned on (power-on status).

*RST

Resets the panel settings.

*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 PWR-01 from executing subsequent commands until all operations in standby are complete.

ABOR

Aborts operations such as change and execute sequence in all sequence groups.

ABOR:PROG

Aborts the sequence trigger function.

ABOR:TRAN

Aborts the setting change trigger function.

DISP:BRIG

Adjusts the screen brightness.

GLOB:*RST

Resets the panel settings of all the channels in the same multichannel domain.

GLOB:*TRG

Trigger command for all the channels in the same multichannel domain.

GLOB:ABOR

Aborts modifications and sequence execution on all trigger subsystems (TRANsient/PROGram) of all the channels in the same multichannel domain.

GLOB:CURR

Sets the current of all the channels in the same multichannel domain.

GLOB:INIT:PROG

Starts the sequence trigger function of all the channels in the same multichannel domain.

GLOB:INIT:TRAN

Starts the setting change (TRANsient) trigger function of all the channels in the same multichannel domain.

GLOB:OUTP

Turns the output on and off of all the channels in the same multichannel domain.

GLOB:OUTP:PROT:CLE

Clears alarms of all the channels in the same multichannel domain.

GLOB:VOLT

Sets the voltagee of all the channels in the same multichannel domain.

INIT:PROG

Starts the sequence trigger function.

INIT:TRAN

Starts the setting change (TRANsient) trigger function.

INST

Specifies the channel to configure.

INST:CAT

Queries the list of channels that can be configured with the INST command.

INST:INFO

Queries the information of the channel currently being controlled.

FETC:ALL/ MEAS:ALL

Queries the current and voltage.

FETC:CURR/ MEAS:CURR

Queries the measured value of the current.

FETC:VOLT/ MEAS:VOLT

Queries the measured value of the voltage.

MEM:REC

Recalls the voltage, current, OVP, UVL, and OCP values saved in the preset memory.

MEM:REC:PREV

Queries the settings that are stored in the preset memory.

MEM:SAVE

Saves the present voltage, current, OVP, UVL, and OCP values in the preset memory.

OUTP

Turns the output on and off.

OUTP:DEL:ON

Sets the output-on delay.

OUTP:DEL:OFF

Sets the output-off delay.

OUTP:EXT

This is an old style command.

OUTP:EXT:ADV

Sets whether output will be turned on and off externally.

OUTP:EXT:LOG

Sets the logic used to control the turning of output on and off using an external contact.

OUTP:PON

Sets the output state at power-on.

OUTP:PROT:BRKT

Sets whether to activate the circuit breaker trip function when alarms occur.

OUTP:PROT:CLE

Clears alarms.

OUTP:PROT:WDOG

Sets the Communication monitor timer.

OUTP:TRIG:STAT

Sets whether to output a trigger signal when the output is turned on.

PROG:CRE

Deletes the existing program and creates a new program.

PROG:LOOP

Changes the number of times that the program will repeat.

PROG:REM:LOOP

Queries the number of repetitions remaining in the sequence that is running.

PROG:REM:TIME

Queries the time remaining in the sequence that is running.

PROG:STEPS

Queries the number of program steps.

PROG:STEPS:LOOP:ADD

Sets the starting step and ending step of an interval loop to execute in the program and the number of interval loops.

PROG:STEPS:LOOP:LIST

Queries all interval loops set in the program.

PROG:STEP<n>:CURR/ PROG:STEP_T:CURR

Sets the current and transition to use in the step.

PROG:STEP<n>:DWEL/ PROG:STEP_T:DWEL

Sets the step execution time.

PROG:STEP<n>:TRIGIN/ PROG:STEP_T:TRIGIN

Sets the step's trigger signal input.

PROG:STEP<n>:TRIGOUT/ PROG:STEP_T:TRIGOUT

Sets the step's trigger signal output.

PROG:STEP<n>:VOLT/ PROG:STEP_T:VOLT

Sets the voltage and transition to use in the step.

PROG:UCOD

Sets the user code to identify sequences.

CURR

Sets the current.

CURR:EXT:RANG

Sets the CC and CV control range that is used during external control.

CURR:EXT:SOUR

Sets whether constant current will be controlled externally.

CURR:LIM:AUTO

Enables or disables the limit on the current setting.

CURR:PROT

Sets the overcurrent protection (OCP) value.

CURR:PROT:DEL

Sets the detection time of OCP activation.

CURR:SST:FALL

Sets the soft stop time when the output is turned off.

CURR:SST:RISE

Sets the soft start time when the output is turned on.

CURR:TRIG

Sets the current value that is applied when a trigger is sent.

RES
Sets the internal resistance.

VOLT
Sets the voltage.

VOLT:EXT:RANG
Sets the CC and CV control range that is used during external control.

VOLT:EXT:SOUR
Sets whether constant voltage will be controlled externally.

VOLT:LIM:AUTO
Set whether to limit the voltage setting so that it does not exceed the OVP setting or become lower than the UVP setting.

VOLT:LIM:LOW
Sets the undervoltage limit (UVL) trip point.

VOLT:PROT
Sets the overvoltage protection (OVP).

VOLT:SST:FALL
Sets the soft stop time when the output is turned off.

VOLT:SST:RISE
Sets the soft start time when the output is turned on.

VOLT:TRIG
Sets the voltage that is applied when a trigger is sent.

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 of the OPERation status register.

STAT:OPER:PTR
Sets the positive transition of the OPERation status register.

STAT:OPER:INST

Queries the event of the OPERation:INSTrument subregister.

STAT:OPER:INST:COND

Queries the condition of the OPERation:INSTrument subregister.

STAT:OPER:INST:ENAB

Sets the enable register of the OPERation:INSTrument subregister.

STAT:OPER:INST:NTR

Sets the negative transition of the OPERation:INSTrument subregister.

STAT:OPER:INST:PTR

Sets the positive transition of the OPERation:INSTrument subregister.

STAT:OPER:INST:ISUM<n>

Queries the event of the OPERation:INSTrument:ISUMmary<n> subregister.

STAT:OPER:INST:ISUM<n>:COND

Queries the condition of the OPERation:INSTrument:ISUMmary<n> subregister.

STAT:OPER:INST:ISUM<n>:ENAB

Sets the enable register of the OPERation:INSTrument:ISUMmary<n> subregister.

STAT:OPER:INST:ISUM<n>:NTR

Sets the negative transition of the OPERation:INSTrument:ISUMmary<n> subregister.

STAT:OPER:INST:ISUM<n>:PTR

Sets the positive transition of the OPERation:INSTrument:ISUMmary<n> subregister.

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 of the QUEStionable status register.

STAT:QUES:PTR

Sets the positive transition of the QUEStionable status register.

STAT:QUES:INST

Queries the event of the QUEStionable:INSTrument subregister.

STAT:QUES:INST:COND

Queries the condition of the QUEStionable:INSTrument subregister.

STAT:QUES:INST:ENAB

Sets the enable register of the QUEStionable:INSTrument subregister.

STAT:QUES:INST:NTR

Sets the negative transition of the QUEStionable:INSTrument subregister.

STAT:QUES:INST:PTR

Sets the positive transition of the QUEStionable:INSTrument subregister.

STAT:QUES:INST:ISUM<n>

Queries the event of the QUEStionable:INSTrument:ISUMmary<n> subregister.

STAT:QUES:INST:ISUM<n>:COND

Queries the condition of the QUEStionable:INSTrument:ISUMmary<n> subregister.

STAT:QUES:INST:ISUM<n>:ENAB

Sets the enable register of the QUEStionable:INSTrument:ISUMmary<n> subregister.

STAT:QUES:INST:ISUM<n>:NTR

Sets the negative transition of the QUEStionable:INSTrument:ISUMmary<n> subregister.

STAT:QUES:INST:ISUM<n>:PTR

Sets the positive transition of the QUEStionable:INSTrument:ISUMmary<n> subregister.

STAT:PRES

Resets the ENABle, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values. 32 bits registers are used by the multichannel function.

SYST:BEEP:STAT

Turns the buzzer on and off.

SYST:COMM:RLST

Sets the operation of the PWR-01 to local or remote.

SYST:CONF:BLE

Sets the bleeder circuit operation.

SYST:CONF:MAST

Queries the number of units in master-slave parallel operation.

SYST:CONF:MON:RANG

Sets the range for monitoring the voltage or current externally.

SYST:CONF:PROT:REC

Sets the output state when a low AC input protection (AC-FAIL) is released.

SYST:CONF:SC1/ SYST:CONF:SC2/ SYST:CONF:SC3

Registers the CONFIG setting item to the panel's SC key.

SYST:CONF:SLAV:AMM

Sets whether or not the current or power on slave units is displayed on the panel during master-slave parallel operation.

SYST:CONF:STAR:PRI

Sets the operation mode to be prioritized when the output is turned on.

SYST:ERR

Reads the oldest error information or event information from the error queue. The error queue can store up to 16 errors.

SYST:ERR:COUN

Returns the number of unread errors in the error queue.

SYST:ERR:TRAC

Sets whether to display communication errors by performing a debug trace.

SYST:EXT:STATOUT:CC:POL

Sets the polarity of the status output signal for constant-current mode.

SYST:EXT:STATOUT:CV:POL

Sets the polarity of the status output signal for constant-voltage mode.

SYST:EXT:STATOUT:OUTP:POL

Sets the polarity of the status output signal for output.

SYST:EXT:STATOUT:PROT:POL

Sets the polarity of the status output signal for protection operation.

SYST:EXT:TRIGIN:POL

Sets the polarity of the trigger signal input.

SYST:EXT:TRIGOUT:POL

Sets the polarity of the trigger signal output.

SYST:KLOC

Sets and releases the panel operation lock (keylock).

SYST:LOC/ SYST:REM/ SYST:RWL

This is an old style command.

SYST:SEC:IMM

Sanitizes all contents stored in memory and initializes the panel settings to their factory default conditions.

SYST:VERS

Queries the version of the SCPI specifications to which the PWR-01 conforms.

TRIG:PROG

Executes a software trigger for the PROGram trigger subsystem.

TRIG:PROG:EXEC

Queries the sequence states (execution state, present repetition count, present step number, elapsed time, estimated time).

TRIG:PROG:SOUR

Sets the condition (trigger source) that determines when the PROGram trigger subsystem actually executes a sequence operation after the PWR-01 receives the INIT:PROG command.

TRIG:TRAN

Executes a software trigger for the TRANsient trigger subsystem.

TRIG:TRAN:SOUR

Sets the condition (trigger source) that determines when the TRANsient trigger subsystem actually changing the setting after the PWR-01 receives the INIT:TRAN command.

Introduction

The PWR-01 series Communication Interface Manual explains the settings and commands for remotely controlling the PWR-01 series.

- RS232C interface
- USB interface
- LAN interface

When the PWR-01 series is operating under remote control, the RMT LED on the display on the front panel lights. To switch from remote mode to local mode from the panel, press LOCAL.

For the safety precautions, installation, operation, and specifications of the PWR-01, read the accompanying PWR-01 series User's Manual.

■ Reading environment

This manual is in PDF format. The manual can be viewed by the following environment.

PDF Reader: Adobe Reader

■ Intended readers

This manual is written for readers with sufficient basic knowledge of how to control 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
- Overview of messages
- Command
- Appendix
- Tutorial

■ Trademarks

Internet Explorer and Visual Basic are registered trademarks of Microsoft Corporation in the United States and/or other countries.

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

■ Firmware version of the product to which this manual applies

This manual applies to products with the following firmware version:
Ver.1.2x

■ Instrument Interface Standards

The PWR-01 conforms to 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 Protcol Specification VXI-11 Rev 1.0 1995
- TCP/IP-IEEE488.2 Interface Specification VXI-11.3 Draft 0.3 1995
- LXI Device Specification 2011 rev 1.4
- LXI HiSLIP Extended Function Rev 1.01
- IVI-6.1 IVI High-Speed LAN Instrument Protocol (HiSLIP) Rev 1.0
- VPP-4.3 The VISA Library 2015 Rev 5.5

■ Copyright and publication

VISA Library

VISA (Virtual Instrument Software Architecture) is a specification for standard software that is used to connect instruments. VISA was defined by the IVI Foundation.

A VISA library is required to use the software application. The VISA library (NI-VISA, Keysight VISA, or KI-VISA) must be installed on the controller (Windows).

If you are controlling the instrument using RS232C or LAN communication from a PLC or microcomputer board, a VISA library is not required.

One of the VISA libraries (driver software implemented in compliance with the VISA specifications) below is necessary.

- NI-VISA by National Instruments (Ver. 5.1.1 or later)
- Keysight VISA by Keysight Technologies (Keysight IO Libraries Suite16.0 or later)
- KI-VISA Ver. 5.0.4 or later

- Note -
- If your VISA library is an older version than that specified, you may not be able to use it depending on the interface.
- Do not install multiple VISA libraries on the same PC. Doing so may cause errors.

Interface Setup

The PWR-01 is equipped with LAN, RS232C, and USB interfaces as standard. In addition to a PC, remote control is possible from a PLC, microcomputer board, or the like that support non-procedural communication.

There is no need to switch interfaces. All interfaces can be used simultaneously.

LAN Interface can be set to OFF in CONFIG settings.

RS232C

USB

LAN

Accessing and Operating the PWR-01 from a Web Browser (LAN interface)

■ RS232C connection

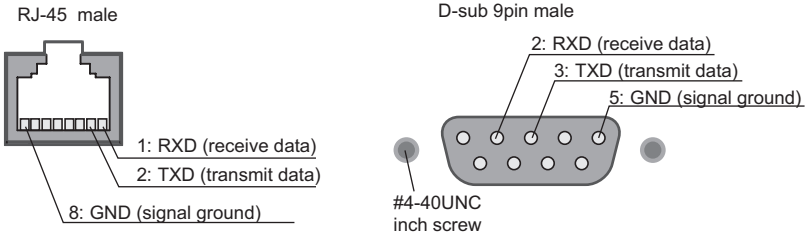
This product's RS232C/TRIG IN connector (RS232C connector) is RJ-45. Check that the PWR-01 and your PC are off before connecting them.

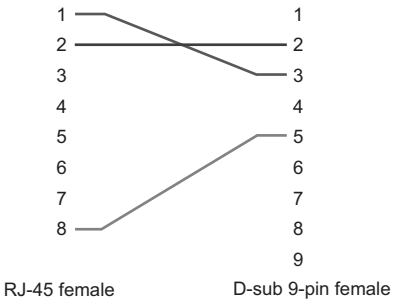
The RS232C port of this product comes with a cover. Remove the cover to use the port. Store the cover in a safe place so that you can use it to cover the RS232C port when the port is no longer in use. For safety reasons, when not using the RS232C port, be sure to attach to cover. For safety reasons, when not using the sensing terminals, be sure to attach to terminal cover.

Port cover
[P1-000-131]

The PWR-01 LAN port is the same shape as a RS232C port. Check the connector name marked on the rear panel to identify the appropriate one to use.

The optional RJ-45 - DB9 adapter cable can be used to connect the PWR-01 to a PC.





RJ-45-DB9 adapter cable

TRIG-IN is assigned to pin 7 of the this product's RS232C connector. TRIG-IN is used for synchronized operation.

■ RS232C configuration

The factory default RS232C settings are RS232C enabled and 19 200 bps baudrate.

For datails of CONFIG settings, see the PWR-01 user's manual.

1 **Press CONFIG several times until CF40 is displayed.**

2 **Turn the VOLTAGE knob until CF42 (RS232C interface setting) is displayed.**

3 **Use the CURRENT knob to select ON.**
RS232C is enabled.

4 **Press CONFIG several times until CF70 is displayed.**

5 **Turn the VOLTAGE knob until CF72 (RS232C interface data rate setting) is displayed.**

6 **Use the CURRENT knob to set the baudrate.**
For the settings, see the table under Protocol below.

7 **Restart the PWR-01.**
The settings are applied.

■ Protocol

The following table shows the settings that correspond to the RS232C protocol.

Underlined values are factory default settings.

Item	Setting
Baudrate	1 200 bps/ 2 400 bps/ 4 800 bps/ 9 600 bps/ <u>19 200 bps/</u> 38 400 bps/ 57 600 bps/ 115 200 bps
Data	Fixed to 8 bit
Stop	Fixed to 1 bit
Flow (X-flow control)	Fixed to None

■ RS232C communication

The PWR-01 does not have flow control (this is fixed). If you send setting commands consecutively at a high speed, the PWR-01 reception buffer may overflow. Do not send setting commands consecutively. Send query commands and read the responses at a certain interval, or reduce the command transmission frequency.

■ Break signal

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

- Note -
The RS232C interface should be shifted remotely by the command. Use the "SYST:COMM:RLST REM" SCPI command to set the RS232C interface to the remote state. Be sure to include this command at the start of the program when you are performing remote programming.

USB

A device driver supporting USB T&M Class (USBTMC) is required to control the PWR-01 through the USB interface. The USBTMC driver is automatically installed by the VISA library.

■ USB connection

Use a standard USB cable to connect the PWR-01 to the PC.

The USB port of this product comes with a cover. Remove the cover to use the port. Store the cover in a safe place so that you can use it to cover the USB port when the port is no longer in use. For safety reasons, when not using the USB port, be sure to attach to cover. For safety reasons, when not using the sensing terminals, be sure to attach to terminal cover.

Port cover
[P1-000-132]

■ USB setting

The factory default USB setting is “USB enabled.”

For datails of CONFIG settings, see the PWR-01 user's manual.

1 **Press CONFIG several times until CF40 is displayed.**

2 **Turn the VOLTAGE knob until CF41 (USB interface setting) is displayed.**

3 **Use the CURRENT knob to select ON.**
USB is enabled.

4 **Restart the PWR-01.**
The settings are applied.

■ Service request

The PWR-01 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

Data rate: 480 Mbps maximum (High speed)

VID (Vender ID)
  0x0B3E

PID (Product ID)
  PWR-01 400W model: 0x1049
  PWR-01 800W model: 0x104A
  PWR-01 1200W model: 0x104B
  PWR-01 2000W model: 0x1055

LAN

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 PWR-01 from a distance, install a Web camera or take other measures to monitor the status.

To use the LAN interface to control the PWR-01, middleware that supports the SCPI-Telnet/ VXI-11/ HiSLIP/ SCPI-RAW protocol must be installed on the controller. The middleware is installed automatically by the VISA library.

There is a Web browser interface to the PWR-01 embedded in the LAN interface board. 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.

Socket communication is possible with a PLC, microcomputer board, or the like that can communicate using the Telnet protocol.

■ LAN connection

Use a standard LAN cable (category 5 and straight) to connect the PWR-01 to a network hub or router.

The LAN port of this product comes with a cover. Remove the cover to use the port. Store the cover in a safe place so that you can use it to cover the LAN port when the port is no longer in use. For safety reasons, when not using the LAN port, be sure to attach to cover. For safety reasons, when not using the sensing terminals, be sure to attach to terminal cover.

Port cover
[P1-000-131]

The PWR-01 RS232C port is the same shape as a LAN port. Check the port name marked on the rear panel to identify the appropriate one to use.

■ LAN setting

For normal use, we recommend using the factory default settings.

Setting	Description (Factory default setting)	CONFIG setting
LAN setting	Use LAN	CF40: ON
IP address allocation method	DHCP: on AUTO IP: on MANUAL IP: off	CF61: 110

When connecting directly, for the IP address allocation method, set DHCP to off, Auto-IP to on, and MANUAL to off (CF61: 010) to automatically set the IP address.

To set the IP address manually, for the IP address allocation method, set DHCP to off, Auto-IP to off, MANUAL to on (CF61: 001), and set the IP address (CF62 to CF65).

If you change the interface settings (CF61 to CF66), apply the changes (CF60: APPL) or restart the power supply to make the new settings take effect.

For datails of CONFIG settings, see the PWR-01 user's manual.

1 **Press CONFIG several times until CF40 (LAN interface setting) is displayed.**

2 **Use the CURRENT knob to select ON.**
LAN is enabled.

3 **Press CONFIG several times until CF60 is displayed.**

4 **Turn the VOLTAGE knob until CF61 (IP address assignment method) is displayed.**

5 **Use the CURRENT knob to set 110.**
DHCP and AUTO IP are set to on, and MANUAL IP is set to off.

6 **Turn the POWER switch off and then back on to enable the settings.**

⚠WARNING
Possible damage to the equipment and electric shock. The LAN interface can be accessed from any place on the network. If necessary, configure the security settings. You can apply password protection for security, and you can restrict the IP addresses to limit the hosts.

- Note -
The LAN interface should be shifted remotely by the command. Use the "SYST:COMM:RLST REM" SCPI command to set the RS232C interface to the remote state. Be sure to include this command at the start of the program when you are performing remote programming.

■ Service request

The PWR-01 is equipped with service request and serial polling functions.

■ LAN function

Complies with the LXI 1.4 Core 2011

Complies with the SCPI-Telnet/ VXI-11/ HiSLIP/ SCPI-RAW protocol

Communication speed: Maximum 100 Mbps (Auto negotiation)

AUTO MDIX function

Web browser access

Instrument information, network information, display of VISA resource information, checking the connected PWR-01, remote control from browser, changing network settings, system status information, license information, password setting

■ Restarting the LAN interface (APPL)

You can use the CONFIG settings to restart the LAN interface (CF60: APPL). However, the setting condition of LAN interface will not be changed.

This operation does not affect the PWR-01's panel settings.

■ Resetting the LAN interface (LCI/ DEF)

You can use the CONFIG settings to reset the LAN interface (CF60: LCI/ DEF).

When reset, network settings are changed as follows.

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

LCI	DEF	Item	Default Value
X	X	Assignment Method	DHCP:ON, Auto-IP:ON, Static:OFF
	X	DNS Server Assignment	0.0.0.0
	X	WINS Server Assignment	0.0.0.0
	X	Desired Hostname	<Model name> - <Last 5 digits of serial number>
	X	Desired Description	KIKUSUI <Model name> Electronic Load - <Serial number>
X	X	Enable Dynamic DNS	Enable
X	X	Enable mDNS	Enable
X	X	Enable NetBIOS Over TCP/IP	Enable
X	X	Password Security	Not set
	X	VMCB Setting	0

Accessing and Operating the PWR-01 from a Web Browser (LAN interface)

You can use the LAN interface to configure detailed settings from a Web browser on your PC. Use latest version of browser. (Recommended browsers: Internet Explore, Chrome, or Safari)

The Web site's URL is defined by adding "http://" in front of the PWR-01's IP address.

You can enter the URL directly in the address bar of your Web browser by using the CONFIG settings (CF50 to CF53) to view the IP address.

When you are using a VISA library, a function is available that enables the application program (such as National Instruments NI-MAX, Keysight Connection Expert, and Kikusui KI-VISA Instrument Explorer) to retrieve the VXI-11 measuring instrument. This function is provided by VISA vendors. You can access the PWR-01 by clicking on the hyperlink that is provided in the retrieval results.

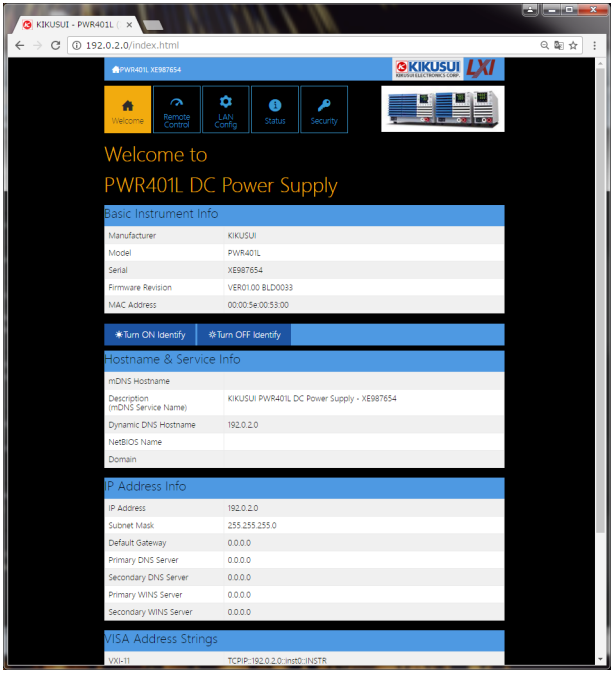
CONFIG	Display	Description
CF50	0 to 255	Display the 1st number of the IP address
CF51	0 to 255	Display the 2nd number of the IP address
CF52	0 to 255	Display the 3rd number of the IP address
CF53	0 to 255	Display the 4th number of the IP address

(Example) When the IP address is 169.254.7.8
http://169.254.7.8

■ WELCOME page

When you access the PWR-01 from a Web browser, the WELCOME page is displayed first.

The instrument information, network information, and VISA resource (I/O resource) information appear on the display. Click items in the navigation menu to move to the other pages.



Turn ON Identify: The LAN LED on the front panel of the connected PWR-01 blinks so that you can identify it.

Turn OFF Identify: The LAN LED blinking stops.

■ Remote Control page

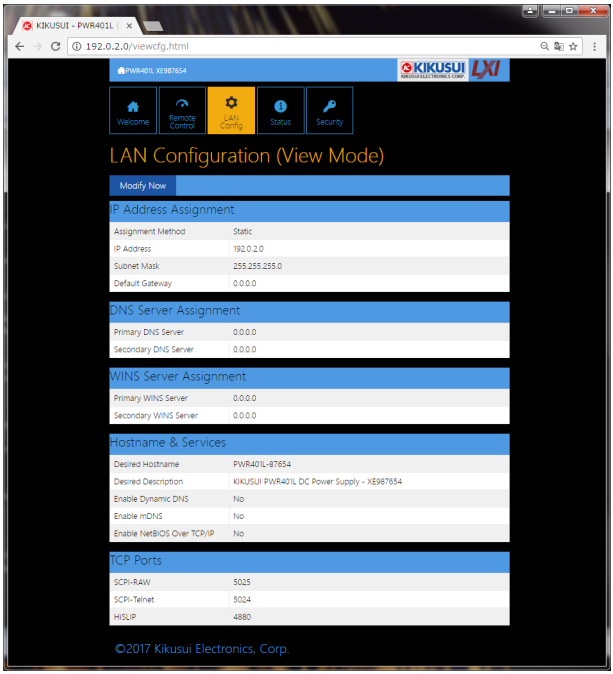
You can remotely control the PWR-01 from a browser. The various buttons have the same functions as those on the front panel of the PWR-01.

⚠WARNING

Should network communication issues occur, the instrument settings shown in this Browser Web Control page may not represent the actual state of the instrument. This may result in unexpected hazardous voltages on the output that could result in personal injury, death or damage to a device under test. Before touching the output or connecting a device under test always verify the state of the istrument.

■ LAN Config page

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



Navigation (View Mode)

Modify Now: Goes to the network setting item editing 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: Goes to the network setting item viewing 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.

Reset and factory default settings

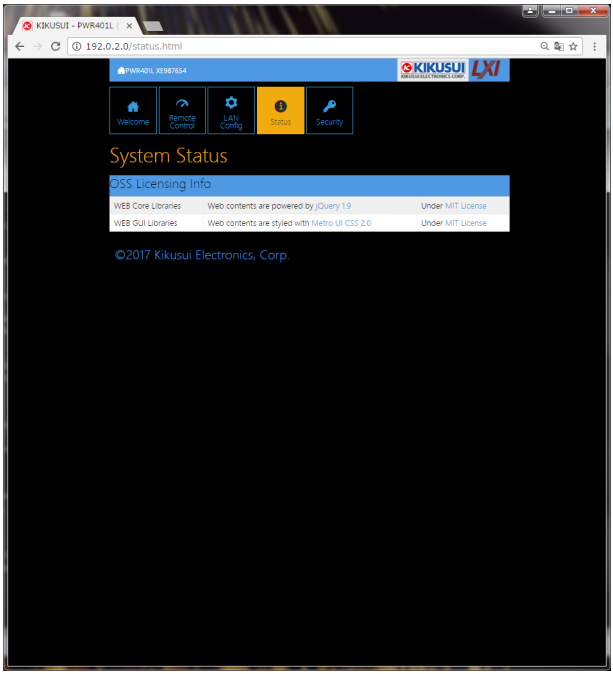
Clicking Reset and Default changes the network settings change as follows.

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

Reset	Default	Item	Default Value
X	X	Assignment Method	DHCP:ON, Auto-IP:ON, Static:OFF
X	X	DNS Server Assignment	0.0.0.0
X	X	WINS Server Assignment	0.0.0.0
	X	Desired Hostname	<Model name> - <Last 5 digits of serial number>
	X	Desired Description	KIKUSUI <Model name> Electronic Load - <Serial number>
X	X	Enable Dynamic DNS	Enable
X	X	Enable mDNS	Enable
X	X	Enable NetBIOS Over TCP/IP	Enable

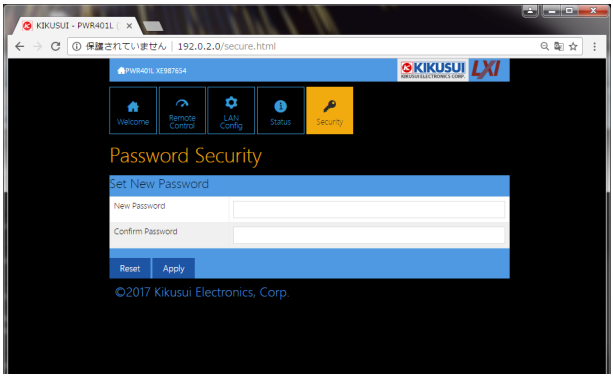
■ Status page

Displays the license information of the open-source software.



■ 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. 15 characters maximum. The first character should be an alphabetical character. The maximum password length is 15 characters.

Changing/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 CONFIG settings (CF60: LCI/DEF) or initialize the PWR-01 to its factory default settings.

For datails of CONFIG settings, see the PWR-01 user's manual.

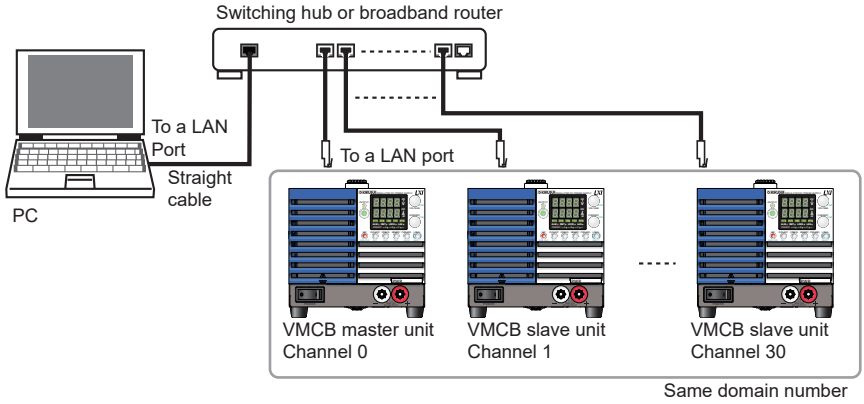
Multichannel Setup

If you use the multichannel (VMCB) function, you can connect one PC to up to 31 PWR-01s to construct a virtual multichannel power supply system. This is useful when you want to synchronize the operation of multiple PWR-01s or minimize the number of communication ports that are required.

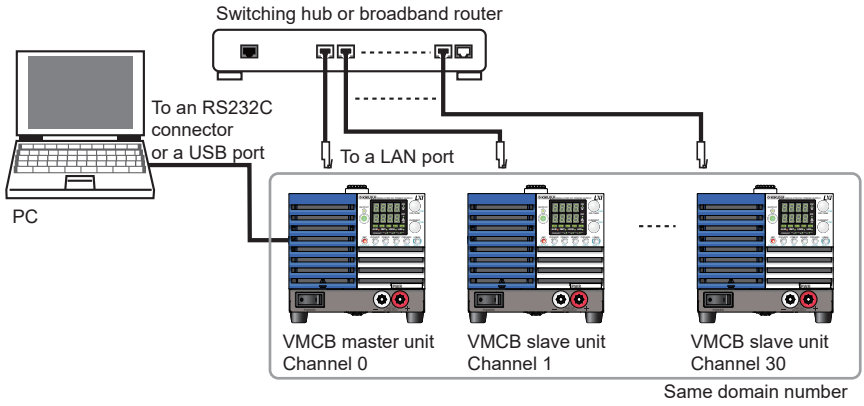
■ Multichannel connections

Set one PWR-01 as the master unit (VMCB master unit), and then connect this PWR-01 to the PC through the LAN, RS232C, or USB interface. The other PWR-01s are slave units (VMCB slave units). Use a switching hub or broadband router to connect the slave units to the master unit through the LAN interface. Use standard LAN cables (category 5).

When the master unit is connected to the PC through the LAN interface



When the master unit is connected to the PC through the RS232C or USB interface



■ Multichannel settings

On the master unit, configure the settings for the connection with the PC, and set the channel and domain.

On the slave units, configure the LAN settings, and set the channels and domains.

PWR-01s that have the same domain number perform multichannel operations as a single VMCB network. You can connect up to 31 PWR-01s to a single VMCB network.

Multichannel uses Multicast DNS (mDNS). Under Hostname / Service on the LAN CONFIG page accessed through a Web browser, the Enable mDNS check box must be selected. By default, this check box is selected.

For datails of CONFIG settings, see the PWR-01 user's manual.

- Note -
- Within the VMCB network, channel 0 is assigned to the master unit and all other channel numbers are assigned to slave units. Do not assign the same channel to two different PWR-01s on the same VMCB network.
- Configure all PWR-01s so that they have the same LAN settings (CF61: IP address assignment method, CF67: VMCB domain number).

Slave unit settings

Normally, you should set DHCP and AUTO IP to ON to set the IP address automatically. For details on the LAN settings, see LAN under Interface Setup.

Configure all PWR-01s on the same VMCB network so that they have the same CONFIG settings (CF61: IP address assignment method, CF67: VMCB domain number).

The remote interface setting (CF40) is not valid on slave units.

1. **Press CONFIG several times until CF60 is displayed.**

2. **Turn the VOLTAGE knob until CF61 (IP address assignment method) is displayed.**

3. **Use the CURRENT knob to select 110.**
   DHCP and AUTO IP are set to on, and MANUAL IP is set to off.
   If you do not want to use a DHCP server or the AUTO IP function, refer to the user's manual and set them accordingly.

4. **Turn the VOLTAGE knob until CF67 (multichannel (VMCB) domain number) is displayed.**

5. **Use the CURRENT knob to set the domain number.**
   PWR-01s that have the same domain number perform multichannel operations. Set this value to a number other than 0.

6. **Turn the VOLTAGE knob until CF68 (multichannel (VMCB) channel number) is displayed.**

7. **Use the CURRENT knob to set the channel number.**
   Specify a unique channel number on the VMCB network. For a slave unit, set this value to a number other than 0.

8. **Turn the PWR-01 off.**

Master unit settings

1 **Configure the settings for the connection with the PC.**
For details on settings for the connection with the PC, see Interface Setup.

2 **Press CONFIG several times until CF60 is displayed.**

3 **Turn the VOLTAGE knob until CF67 (multichannel (VMCB) domain number) is displayed.**

4 **Use the CURRENT knob to set the domain number.**
PWR-01s that have the same domain number perform multichannel operations. Set this value to a number other than 0.

5 **Turn the VOLTAGE knob until CF68 (multichannel (VMCB) channel number) is displayed.**

6 **Use the CURRENT knob to select 0.**
Because this is the master unit, set this value to 0.

7 **Turn the PWR-01 off.**

⚠Caution
The LAN interface can be accessed from any place on the network. If necessary, configure the security settings. You can apply password protection for security, and you can restrict the IP addresses to limit the hosts.

Turning power on

First turn the slave units on, and then turn the master unit on. If the PWR-01s are organized in a rack system, turn all the units on at the same time. If you turn the master unit on first, the slave units will not be detected correctly.

When a slave unit turns on, all its LEDs light, and then the following sequence is displayed: the rated voltage and rated current, the firmware version number, and then the build number. Each item is displayed for approximately 1 second. Then, the slave unit displays "I-F SLAV" and enters the wait state. When the PWR-01 is designated as a slave unit by the master unit, the slave unit enters the operation wait state.

The slave unit will not stop displaying "I-F SLAV" until it receives the slave designation from the master unit. Check the CONFIG parameters (CF61, CF67, and CF68) and the LAN connection. If you change the CONFIG parameters, reboot the PWR-01.

When the master unit turns on, all its LEDs light, and then the following sequence is displayed: the rated voltage and rated current, the firmware version number, and then the build number. Each item is displayed for approximately 1 second. Then, the master unit displays "FIND CH" and searches for slave units. When the master unit finishes searching, it enters the operation wait state.

Checking the multichannel configuration

To check the multichannel configuration, send the INSTrument:CATalog? query. You cannot check the configuration from the panel.

Overview of Command

The information that is exchanged between the controller (PC) and the device (PWR-01 series) is called a message.

The PWR-01 uses the SCPI language for the messages.

There are two types of messages, commands that are sent from the PC to the PWR-01 and responses that are sent from the PWR-01 to the PC.

Command Hierarchy

SCPI commands are ASCII-based commands designed for test and measurement devices. The command hierarchy is structured around the common root or node, which is the construction block of the SCPI subsystem. A command consists of a program header, parameters, and punctuation.

The hierarchy is explained using the SOURce subsystem as an example.

Program header	Parameter	Hierarchy of node
SOUR:		Root node
CURR		Second level
:LIM		Third level
:AUTO	<boolean>	Fourth level
VOLT		Second level
:EXT		Third level
:RANG	<character>	Fourth level
:SOUR	<character>	Fourth level

A higher node is separated from a lower node using a colon (:).

Command Syntax

This manual denotes SCPI commands using the following format.

```
[SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude] {<numeric>|MINi-
    mum|MAXimum}
```

SCPI commands can be issued using the short form. The short form of a SCPI command is the section of the command written in uppercase.

SCPI commands can be sent in the long form or short form. Since SCPI commands are not case-sensitive, CURR, Curr, and curr are all accepted as the short form of CURRent. In the long form, CURRENT, Current, and current are all acceptable.

- A space is required between the program header section and the parameter section.
- Multiple parameters, when available, are concatenated using commas.
- Commands are concatenated using semicolons (compound command).

```
CURRent:SSTart:FALL 25MS;RISE 25MS
```

In the second command, CURRent:SSTart: is omitted. This is because the path is set to CURRent:SSTart: by the first command CURRent:SSTart:FALL.

This compound command is the same as entering the following commands.

```
CURRent:SSTart:FALL 25MS
CURRent:SSTart:RISE 25MS
```

An error occurs if a node that is not defined in the current path is designated.

Commands of different subsystems can be concatenated using a colon and a semicolon together.

```
SYSTem:CONFigure:STARtup:PRIority CV;:VOLTage:SSTart:FALL 25MS
```

This compound command contains two root nodes, SYSTem and VOLTage.

When the second or subsequent command starts with a colon, the path specified by the previous command is cleared.

- The maximum number of characters that can be transmitted in a single line is 512.

■ Special symbols

Special symbols used in this manual to describe SCPI commands are defined below.

- Characters and numbers delimited by "|" in braces indicate that one of the items is to be selected.
  Do not include the braces in the actual program.
- The characters <> indicate program data.
  Do not write <> in the actual program.
- Brackets indicate option data.
  When option data is not sent with the program, the default value is applied.
  Do not write [ ] in the actual program.

■ Queries

The device settings or status can be queried.

To make a query, add a question mark at the end of the program header section.

If a query has parameters, enter a space after the question mark followed by the parameters.

```
CURRent? MIN
```

Response

A response returned as an answer to a query. It is a message that is always sent from the device to the PC. The status of the device or measured values are transmitted to the PC.

- Note -
When transmitting two queries in separate lines, read the response to the first query before transmitting the second line.

■ Program terminator

All commands must be terminated using a valid terminator.

	RS232C	USB	LAN	
			VXI-11, HiSLIP	SCPI-RAW
reception	LF	LF or EOM	LF or END	LF
transmission	LF or CR+LF	LF+EOM	LF+END	LF

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

- Note -
CR (ASCII 0x0D) is not a terminator.

■ Common commands

The IEEE-488.2 and SCPI standards contain a set of common commands for reset, self-test, and other functions. These common commands always start with an asterisk. The commands may have one or multiple parameters.

Parameters

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

The representation system of the program data that is used on the PWR-01 is indicated below.

■ Non-numeric parameters

Character string data (String)

Used when a series of ASCII characters are requested.

Be sure to enclose a string in single or double quotation marks. The start and end quotation marks must match.

```
SYSTem:LANGuage "SCPI"
```

If you wish to use a quotation mark as a part of the string, enter two quotation marks consecutively (with no characters in between).

Character data

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

```
TRIGger:TRANSient:SOURce {BUS|IMMediate}
```

Boolean data

Boolean data expresses a 1 or 0 condition or an ON or OFF condition. Responses are returned as 1 or 0

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

■ Numeric parameters

NR1

Represents an integer.

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 (floating point).

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

NR3

Represents a real number (exponential).

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

The value +3.80000+E02 is returned for the response data 380. The number of digits to the right of the decimal is 5.

NRf

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

Numeric

A numeric parameter such as a decimal point, optional prefix, or measurement unit.

The syntax as a numeric representation is the same as NRf.

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

Units such as V, A, and W can also be used in a numeric parameter.

If a value that cannot be assigned is entered, the device rounds the value to the closest possible value.

```
OUTP:DEL:ON 1.22
```

The setting resolution of the output-on delay is 0.1, so +1.20000E+00 is returned in response to the OUTP:DEL:ON? query.

■ Special form numeric parameters

The special form numeric parameters MINimum, MAXimum and DEFault can be used as substitutes for limit values when the parameter is numeric. In the example below, the current limit is set to the minimum value.

```
SOURce:CURRent MINimum
```

Queries can be used to inquire 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) •A (current) •HR (hour) MIN (minute)
•S (second)

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

Returns the value of the event status enable register in NR1 format in response to *ESE?.

*ESR

Queries the event status register.

Registers that are read are cleared.

Command

```
*ESR?
```

Response

Returns the value of the event status register in NR1 format in response to *ESR? and clears the register.

*IDN

Queries the model name, serial number, and firmware version of the PWR-01.

Command

```
*IDN?
```

Response

The response to *IDN? is indicated below.

(Example) For a PWR401L with serial number AB1234 and firmware version VER01.01 BLD0001, this returns:

```
KIKUSUI,PWR401L,AB1234,VER01.01 BLD0001
```

*LRN

Queries the command that can restore the current panel settings.

Values to be restored are those affected by the *RST command.

Command

```
*LRN?
```

Response

Returns the panel setting restore command as an ASCII character string in response to *LRN?.

*OPC

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

See section 12.5.3 in IEEE 488.2-1992.

Command

```
*OPC
*OPC?
```

Response

Returns 1 in response to *OPC? when all the commands in standby have been completed.

*OPT

Queries the option that are installed in the PWR-01.

Command

```
*OPT?
```

Response

Returns "VMCB","VIR" in response to *OPT?.

"VMCB" Virtual multichannel function (standard equipped on all models)

"VIR" Variable internal resistance (standard equipped on all models)

*PSC

Sets whether to clear the event status enable register and the service request enable register when the POWER switch is turned on (power-on status).

Command

```
*PSC <boolean>
*PSC?
```

Parameter <boolean>

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

(Example) To enable the power-on SRQ function

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

Response

Returns the power-on status setting in response to *PSC?.

*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.

Setting Items	When *RST performed
OUTPut	OFF
OUTPut:DELay:ON	0.0[s]
OUTPut:DELay:OFF	0.0[s]
OUTPut:EXTernal	OFF
[SOURce:]CURRent	MAXimum
[SOURce:]CURRent:EXTernal:SOURce	NONE
[SOURce:]CURRent:LIMit:AUTO	ON
[SOURce:]CURRent:PROTection	MAXimum
[SOURce:]CURRent:SST:FALL	0.0[s]
[SOURce:]CURRent:SST:RISE	0.0[s]
[SOURce:]CURRent:TRIGgered	MAXimum
[SOURce:]VOLTage	0.0[V]
[SOURce:]VOLTage:EXTernal:SOURce	NONE
[SOURce:]VOLTage:LIMit:AUTO	ON
[SOURce:]VOLTage:LIMit:LOW	MINimum
[SOURce:]VOLTage:PROTection	MAXimum
[SOURce:]VOLTage:SST:FALL	0.0[s]
[SOURce:]VOLTage:SST:RISE	0.0[s]
[SOURce:]VOLTage:TRIGgered	0.0[V]
TRIGger:TRANsient:SOURce	IMMediate
TRIGger:PROGram:SOURce	IMMediate
RES	MINimum

Command

```
*RST
```

*SRE

Sets the service request enable register.

The service request enable register is used to select the summary messages in the status byte register that will be able to perform service requests.

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

Returns the value of the service request enable register in NR1 format in response to *SRE?.

*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

Returns the value of the status byte register and the MSS message (bit 6) in NR1 format in response to *STB?.

*TRG

Trigger command.

Executes triggers on the TRANsient trigger group and PROGram trigger group.

This is a substitute command for the IEEE488.1 get (Group Execute Trigger) command.

If the PWR-01 is not in a condition to accept triggers, an SCPI error (-211,"Trigger ignored") occurs.

See section 10.37 in IEEE 488.2-1992.

Command

```
*TRG
```

*TST

Executes a self-test.

Use SYST:ERR? to query the errors that occurred. See section 10.38 in IEEE 488.2-1992.

Command

```
*TST?
```

Response

Returns +0 if there are no errors in response to *TST?. Returns the error code, if there are errors.

*WAI

Prevents the PWR-01 from executing subsequent commands until all operations in standby are complete.

Command

```
*WAI
```

ABORt Command

PWR-01 has two different trigger subsystems -TRANsient, and PROGram.

The TRANsient group is a trigger subsystem that changes the output voltage and current settings.

The PROGram group is used to execute sequences on the PWR-01.

ABOR

Aborts operations such as change and execute sequence in all sequence groups.

The trigger status immediately after the power is turned on is the same as the trigger status when the ABOR command is received.

A specific sequence group cannot be specified with the ABOR command. It is always interpreted as ALL.

Command

```
ABORt[:ALL]
```

ABOR:PROG

Aborts the sequence trigger function.

Command

```
ABORt:PROGram
```

ABOR:TRAN

Aborts the setting change trigger function.

Command

```
ABORt:TRANsient
```

DISP:BRIG

Adjusts the screen brightness.

Command

```
DISPlay:BRIGhtness <NRf>
DISPlay:BRIGhtness?
```

Parameter

Value: 1 to 7 (7 by default)

(Example)

```
DISP:BRIG 4
```

Response

Returns the screen brightness in NR1 format in response to DISP:BRIG?

GLOB:*RST

Resets the panel settings of all the channels in the same multichannel domain.

For the settings that are reset, see *RST.

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.

Insert a wait of at least 200 ms after sending the GLOB:*RST command.

Command

```
GLOBal:*RST
```

GLOB:*TRG

Trigger command for all the channels in the same multichannel domain.

Executes triggers on the TRANsient trigger group and PROGram trigger group.

This is a substitute command for the IEEE488.1 get (Group Execute Trigger) command.

Insert a wait of at least 200 ms after sending the GLOB:*TRG command.

Command

```
GLOBal:*TRG
```

GLOB:ABOR

Aborts modifications and sequence execution on all trigger subsystems (TRANsient/PROGram) of all the channels in the same multichannel domain.

Insert a wait of at least 200 ms after sending the GLOB:ABOR command.

Command

```
GLOBal:ABORt
```

GLOB:CURR

Sets the current of all the channels in the same multichannel domain.

Insert a wait of at least 200 ms after sending the GLOB:CURR command.

Command

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

Parameter

Value: 0 % to 105 % of the rated output current (the default value is MAXimum)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: A

Rated output current

PWR401L	40 A
PWR401ML	20 A
PWR401MH	5 A
PWR401H	1.85 A
PWR801L	80 A
PWR801ML	40 A
PWR801MH	10 A
PWR801H	3.70 A

PWR1201L	120 A
PWR1201ML	60 A
PWR1201MH	15 A
PWR1201H	5.55 A
PWR2001L	200 A
PWR2001ML	100 A

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

(Example)

```
GROB:CURR 2.5
```

GLOB:INIT:PROG

Starts the sequence trigger function of all the channels in the same multichannel domain.

When the trigger source is set to IMM, the sequence is executed immediately. When the trigger source is set to BUS, the PRW01 waits for a software trigger, and then executes the sequence.

This command cannot be executed simultaneously with the TRANsient subsystem.

This command is not valid when the output is on.

Insert a wait of at least 200 ms after sending the GLOB:INIT:PROG command.

Command

```
GLOBal:INITiate[:IMMediate]:PROGram
```

Related Command

```
TRIG:PROG:SOUR
```

GLOB:INIT:TRAN

Starts the setting change (TRANsient) trigger function of all the channels in the same multichannel domain.

If trigger source is set to IMM, the change starts immediately. If set to BUS, the change starts after waiting for a software trigger. If set to TRIGIN, the change starts after waiting for a hardware trigger.

This command cannot be executed simultaneously with the PROGram subsystem.

Insert a wait of at least 200 ms after sending the GLOB:INIT:TRAN command.

Command

```
GLOBal:INITiate[:IMMediate]:TRANsient
```

Related Command

```
TRIG:TRAN:SOUR
```

GLOB:OUTP

Turns the output on and off of all the channels in the same multichannel domain.

This command is invalid when a protection function is activated.

OUTP ON is invalid while a sequence is running. OUTP OFF terminates the sequence and turns the output off.

Insert a wait of at least 200 ms after sending the GLOB:OUTP command.

Command

```
GLOBal:OUTPut[:STATe][:IMMediate] <boolean>
```

Parameter

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

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

(Example)

```
GLOB:OUTP 1
```

GLOB:OUTP:PROT:CLE

Clears alarms of all the channels in the same multichannel domain.

Insert a wait of at least 200 ms after sending the GLOB:OUTP:PROT:CLE command.

Command

```
GLOBal:OUTPut:PROTection:CLEar
```

GLOB:VOLT

Sets the voltagee of all the channels in the same multichannel domain.

Insert a wait of at least 200 ms after sending the GLOB:VOLT command.

Command

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

Parameter

Value: 0 % to 105 % of the rated output voltage (the default value is MAXimum)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: V

Rated output voltage

PWR401L	40 V
PWR401ML	80 V
PWR401MH	240 V
PWR401H	650 V
PWR801L	40 V
PWR801ML	80 V
PWR801MH	240 V
PWR801H	650 V

PWR1201L	40 V
PWR1201ML	80 V
PWR1201MH	240 V
PWR1201H	650 V
PWR2001L	40 V
PWR2001ML	80 V

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

(Example)

```
GLOB:VOLT 120
```

INIT:PROG

Starts the sequence trigger function.

When the trigger source is set to IMM, the sequence is executed immediately.
When the trigger source is set to BUS, the PWR-01 waits for a software trigger, and then executes the sequence.

This command cannot be executed simultaneously with the TRANsient subsystem.

This command is not valid when the output is off.

Command

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

Related Command

```
TRIG:PROG:SOUR
```

INIT:TRAN

Starts the setting change (TRANsient) trigger function.

If trigger source is set to IMM, the change starts immediately. If set to BUS, the change starts after waiting for a software trigger. If set to TRIGIN, the change starts after waiting for a hardware trigger.

This command cannot be executed simultaneously with the PROGram subsystem.

Command

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

Related Command

```
TRIG:TRAN:SOUR
```

This product supports virtual multichannels based on the VMCB architecture.

When the product is running as a VMCB master, detected VMCB slave units can be selected. VMCB slave units cannot be controlled directly from a PC.

INST

Specifies the channel to configure.

Insert a wait of at least 200 ms after sending the INST command.

Command

```
INSTrument[:SELect] <NRf>
INSTrument[:SELect]?
INSTrument[:NSELect] <NRf>
INSTrument[:NSELect]?
```

Parameter

Value:	0	VMCB master unit (default)
	1 to 30	VMCB slave unit

(Example)

```
INST 7
```

Response

Returns the channel that is being configured in NR1 format in response to INST?.

INST:CAT

Queries the list of channels that can be configured with the INST command.

Command

```
INSTrument:CATalog?
```

Response

Returns the channels that can be configured in NR1[,NR1...] format in response to INST:CAT?. If the multichannel function is not in use, this query returns "+0."

+0	Master unit
+1 to +30	VMCB slave unit

For example, in a system consisting of three PWR-01s assigned to channels 0, 1, and 4, this query returns +0,+1,+4.

INST:INFO

Queries the information of the channel currently being controlled.

The channel under control is set with the INST command.

Command

```
INSTrument:INFO?
```

Response

Returns the maximum voltage, maximum current, maximum power, and model name of the channel under control in response to INST:INFO?.

(Example) If the channel under control is a PWR401ML (maximum voltage: 80 V, maximum current: 20 A, maximum power: 400 W)

```
+8.0000E+01, +2.0000E+01, +4.0000E+02, PWR401ML
```

is returned.

This product does not have the INITiate or FETCh measurement function.

The measured voltage and the measured current are updated alternately at 25 ms intervals.

MEASure and FETCh are aliases and operate in the same manner. Queries the latest measured value when the command is sent.

-> Tutorial “Settings and Measurement” (p.201)

FETC:ALL/ MEAS:ALL

Queries the current and voltage.

When you are using the multichannel function, use <n> to specify the channel number.

Command

```
FETCh[<n>][:SCALar]:ALL?
MEASure[<n>][:SCALar]:ALL?
```

(Example)

```
MEAS:ALL?
MEAS2:ALL?
```

Response

Returns the current <NR3> and then the voltage <NR3> in a comma-separated format in response to FETC:ALL?/ MEAS:ALL?.

When multichannel is in use, this command returns the current <NR3> and then the voltage <NR3> of the specified channel in a comma-separated format in response to FETC<n>:ALL?/ MEAS<n>:ALL?. If you omit <n>, the measured value of the channel that was specified by the INST command is returned. The measured voltage and the measured current are updated alternately at 25 ms intervals.

Unit:	Current	A
	Voltage	V

FETC:CURR/ MEAS:CURR

Queries the measured value of the current.

When you are using the multichannel function, use <n> to specify the channel number.

Command

```
FETCh[<n>][:SCALar]:CURRent[:DC]?
MEASure[<n>][:SCALar]:CURRent[:DC]?
```

(Example)

```
MEAS:CURR?
MEAS2:CURR?
```

Response

Returns the measured value of the current in NR3 format in response to FETC:CURR?/ MEAS:CURR?. The measured voltage and the measured current are updated alternately at 25 ms intervals.

When you are using the multichannel function, this query returns the measured current of the specified channel in NR3 format in response to FETC<n>:CURR?/ MEAS<n>:CURR?. If you omit <n>, the measured value of the channel that was specified by the INST command is returned.

Unit: A

FETC:VOLT/ MEAS:VOLT

Queries the measured value of the voltage.

When you are using the multichannel function, use <n> to specify the channel number.

Command

```
FETCh[<n>][:SCALar]:VOLTage[:DC]?
MEASure[<n>][:SCALar]:VOLTage[:DC]?
```

(Example)

```
MEAS:VOLT?
MEAS2:VOLT?
```

Response

Returns the measured value of the voltage in NR3 format in response to FETC:-VOLT?/ MEAS:VOLT?. The measured voltage and the measured current are updated alternately at 25 ms intervals.

When you are using the multichannel function, this query returns the measured voltage of the specified channel in NR3 format in response to FETC<n>:VOLT?/ MEAS<n>:VOLT?. If you omit <n>, the measured value of the channel that was specified by the INST command is returned.

Unit: V

MEM:REC

Recalls the voltage, current, OVP, UVL, and OCP values saved in the preset memory.

This command is invalid when a sequence is running.

You can view the settings that are stored in memory by using the MEM:REC:PREV command.

Command

```
MEMory:RECall[:IMMediate] <NRf>
```

Parameter

Value:	1	Memory A
	2	Memory B
	3	Memory C

(Example)

```
MEM:REC 2
```

MEM:REC:PREV

Queries the settings that are stored in the preset memory.

Command

```
MEMory:RECall:PREView? <NRf>
```

Parameter

Value:	1	Memory A
	2	Memory B
	3	Memory C

Response

Returns the current <NR3>, voltage <NR3>, OCP<NR3>, OVL<NR3>, and UVL<NR3> that are stored in the specified preset memory in a comma-separated format in response to MEM:PREV? <NRf>.

MEM:SAVE

Saves the present voltage, current, OVP, UVL, and OCP values in the preset memory.

Command

```
MEMory:SAVE[:IMMediate] <NRf>
```

Parameter

Value:	1	Memory A
	2	Memory B
	3	Memory C

(Example)

```
MEM:SAVE 3
```

OUTP

Turns the output on and off.

This command is invalid when a protection function is activated.

OUTP ON is invalid while a sequence is running. OUTP OFF terminates the sequence and turns the output off.

Command

```
OUTPut[:STATe][:IMMediate] <boolean>
OUTPut[:STATe][:IMMediate]?
```

Parameter

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

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

(Example)

```
OUTP 1
```

Response

Returns whether the output is on in NR1 format in response to OUTP?. This product's output status is updated at 50 ms intervals.

OUTP:DEL:ON

Sets the output-on delay.

Command

```
OUTPut:DELay:ON <numeric>
OUTPut:DELay:ON?
```

Parameter

Value: 0 No delay (default)
0.5 to 99.9
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: S

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

(Example)

```
OUTP:DEL:ON 1.2
```

Response

Returns the output-on delay setting in NR3 format in response to OUTP:DEL:ON?.

OUTP:DEL:OFF

Sets the output-off delay.

Command

```
OUTPut:DELay:OFF <numeric>
OUTPut:DELay:OFF?
```

Parameter

Value: 0 No delay (default)
0.5 to 99.9
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: S

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

(Example)

```
OUTP:DEL:OFF 1.2
```

Response

Returns the output-off delay setting in NR3 format in response to OUTP:DEL:OFF?.

OUTP:EXT

This is an old style command.

When creating new programs, use OUTP:EXT:ADV (p.91).

Command

```
OUTPut:EXTernal[:STATe] <boolean>
OUTPut:EXTernal[:STATe]?
```

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

Response

NR1 format

OUTP:EXT:ADV

Sets whether output will be turned on and off externally.

Set the external control logic with OUTP:EXT:LOG.

This command is invalid while a sequence is running.

Command

```
OUTPut:EXTernal[:STATe]:ADVance <character>
OUTPut:EXTernal[:STATe]:ADVance?
```

Parameter

Value:	ON	External control is performed. Disables the output-on/ off delay function and soft start and soft stop functions.
	ENHancedon	External control is performed. Enables the output-on/ off delay function and soft start and soft stop functions.
	OFF	External control is not performed (default).

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

(Example)

```
OUTP:EXT:ADV ON
```

Response

Returns whether output will be turned on and off externally in CHAR format in response to OUTP:EXT:ADV?.

OUTP:EXT:LOG

Sets the logic used to control the turning of output on and off using an external contact.

This command is invalid while a sequence is running.

Command

```
OUTPut:EXTernal:LOGic <character>
OUTPut:EXTernal:LOGic?
```

Parameter

Value:	LOW	The output is turned on with a low signal (default).
	HIGH	The output is turned on with a high-level signal or open circuit.

(Example)

```
OUTP:EXT:LOG HIGH
```

Response

Returns the logic used to control the turning of output on and off using an external contact in CHAR format in response to OUTP:EXT:LOG?.

OUTP:PON

Sets the output state at power-on.

Command

```
OUTPut:PON[:STATe] <character>
OUTPut:PON[:STATe]?
```

Parameter

Value:	SAFE	The output is off. (default)
	AUTO	The PWR-01 turns on in the previous power-off state.
	FORCe	The output is on.

(Example)

```
OUTP:PON AUTO
```

Response

Returns the output state at power-on in character format in response to OUTP:PON:STAT?.

OUTP:PROT:BRKT

Sets whether to activate the circuit breaker trip function when alarms occur.

This is available only for the PWR2001L or PWR2001ML.

Command

```
OUTPut:PROTection:BRKTrip <boolean>
```

```
OUTPut:PROTection:BRKTrip?
```

Parameter

設定値： ON(1) Activate (default)
OFF(0) Not activate

(Example)

```
OUTP:PROT:BRKT ON
```

Response

Returns whether the circuit breaker trip function is to be activated in NR1 format in response to OUTP:PROT:BRKT?

OUTP:PROT:CLE

Clears alarms.

Command

```
OUTPut:PROTection:CLEar
```

OUTP:PROT:WDOG

Sets the Communication monitor timer.

When an alarm occurs, set the value to zero first and then clear the alarm.

Command

```
OUTPut:PROTection:WDOG[:DELay] <numeric>
```

```
OUTPut:PROTection:WDOG[:DELay]?
```

Parameter

Value: 0 Communication monitoring function off (default)

1, 3, 10, 30, 100, 300, 1000, 3000

A SCPI error (-222, “Data out of range”) occurs if outside the range.

If a value that cannot be set is specified, the next higher value is set.

(Example) If OUTP:PROT:WDOG 31 is specified, 100 is set.

Unit: S

(Example)

```
OUTP:PROT:WDOG 30
```

Response

Returns the Communication monitor timer in NR1 format in response to OUTP:PROT:WDOG?.

Related Command

```
OUTP:PROT:CLE
```

OUTP:TRIG:STAT

Sets whether to output a trigger signal when the output is turned on.

Command

```
OUTPut:TRIGger:STATe <boolean>
OUTPut:TRIGger:STATe?
```

Parameter

Value:	ON(1)	Output
	OFF(0)	Not output (default)

(Example)

```
OUTP:TRIG:STAT ON
```

Response

Returns whether a trigger signal is output when the output is turned on in NR1 format in response to OUTP:TRIG:STAT?.

PROGram Command

The sequence function is mapped to the PROGram trigger subsystem.

A single program consisting of up to 64 steps (indexes 0 to 63) can be created.

The sequence content is retained even when the power is turned off.

Sequence creation and editing commands are invalid while a sequence is running.

Program template

The contents of a program template can be reflected in the initial step of a newly created program (PROG:CRE).

The contents of a template can be changed while a program is running.

The settings of a program template cannot be saved.

The settings do not change even when you send a *RST or *RCL command. The contents returned to their default values when the PWR-01 is restarted.

-> Tutorial “Sequence (PROGram)” (p.206)

PROG:CRE

Deletes the existing program and creates a new program.

The repetition count and user code are set to default values.

Specify also the number of steps to use in the program. This number cannot be changed later.

Command

```
PROGram:CREate <NRf>[,<character>]
```

Parameter <NRf>

Value: 1 to 64 Number of steps

Parameter <character>

Value: DEFault Default values are copied into all the steps in this program (default)

TEMPlate Values from the template are copied into all the steps in this program.

DEFault values

Step time	1 s
Voltage	0 V
Voltage transition	IMMediate
Current	MAXimum
Current transition	IMMediate
Trigger output	OFF
Trigger input	OFF

(Example)

```
PROG:CRE 30,TEMP
```

PROG:LOOP

Changes the number of times that the program will repeat.

Command

```
PROGram[:SELected]:LOOP[:COUNt] {<NRf>|INFinity}
PROGram[:SELected]:LOOP[:COUNt]?
```

Parameter

Value:	1 to 99998	Repetition count (1 by default)
	99999 or higher or INFinity	Repeated indefinitely

Settings are reset to default when the PROG:CLE command is sent.

(Example)

```
PROG:LOOP 100
```

Response

Returns the number of program repetitions in NR1 format in response to PROG:LOOP?. If the number of program repetitions is set to infinity, INF is returned.

PROG:REM:LOOP

Queries the number of repetitions remaining in the sequence that is running.

Command

```
PROGram:REMaining:LOOP?
```

Response

Returns the remaining number of sequence repetitions in NR1 format in response to PROG:REM:LOOP?.

Returns 0 if a sequence is not running.

If the number of program repetitions is set to infinity, INF is returned.

PROG:REM:TIME

Queries the time remaining in the sequence that is running.

Command

```
PROGram:REMaining:TIME?
```

Response

Returns the time remaining in the sequence in NR1 format in response to PROG:REM:TIME?.

Returns 0 if a sequence is not running.

If the number of program repetitions is set to infinity, INF is returned.

Unit: S

PROG:STEPS

Queries the number of program steps.

Command

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

Response

Returns the number of program steps in NR1 format in response to PROG:-STEPS?.

PROG:STEPS:LOOP:ADD

Sets the starting step and ending step of an interval loop to execute in the program and the number of interval loops.

The starting step must be set to a smaller number than the ending step.

Interval loops can be set in up to 16 locations in a single program.

Once you set them, you cannot edit or delete them.

An interval loop cannot overlap with another interval loop.

Interval loop cannot be nested.

You cannot edit or delete interval loops that have already been set.

Command

```
PROGram[:SELected]:STEPS:LOOP:ADD <NRf_begin>,<NRf_end>,<NRf_
    count>
```

Parameter <NRf_begin>

Value: 0 to number of steps-2 Starting step

Parameter <NRf_end>

Value: 1 to number of steps-1 Ending step

Parameter <NRf_count>

Value: 2 to 99998 Number of interval loops

(Example) To repeat step 0 to step 2 twice

```
PROG:STEPS:LOOP:ADD 0,2,2
```

PROG:STEPS:LOOP:LIST

Queries all interval loops set in the program.

Command

```
PROGram[:SELected]:STEPS:LOOP:LIST?
```

Response

Returns the starting step <NR1>, ending step <NR1>, number of interval loops <NR1> for all interval loops in order in response to PROG:STEPS:LOOP:LIST?

(Example) When an interval loop with starting step set to 0, ending step set to 2, and number of interval loops set to 2 and an interval with starting step set to 5, ending step set to 7 and number of interval loops set to 3 are set

```
+0,+2,+2,+5,+7,+3
```

is returned.

PROG:STEP<n>:CURR/ PROG:STEP_T:CURR

Sets the current and transition to use in the step.

PROG:STEP<n>:CURR sets the current and transition to use in step n. Use <n> to specify the step number. Step numbers start at 0.

PROG:STEP_T:CURR sets the current and transition of the program template.

Substitutes MAXimum and MINimum cannot be used in queries.

Command

```
PROGram[:SELected]:STEP<n>:CURRent <numeric>[,<character>]
PROGram[:SELected]:STEP<n>:CURRent?
```

Command (Template)

```
PROGram[:SELected]:STEP_T:CURRent <numeric>[,<character>]
PROGram[:SELected]:STEP_T:CURRent?
```

Parameter <numeric>

Value: See the CURR command.
Unit: A

Parameter <character>

Value:	IMMediate	Transition in steps (default)
	RAMP	Transition with a slope

(Example)

```
PROG:STEP0:CURR 10,RAMP
```

Response

Returns the current <NR3> and transition <character> in a comma-separated format in response to PROG:STEP<n>:CURR?/ PROG:STEP_T:CURR?.

PROG:STEP<n>:DWEL/ PROG:STEP_T:DWEL

Sets the step execution time.

PROG:STEP<n>:DWEL sets the execution time of step n of the applicable program. Use <n> to specify the step number. Step numbers start at 0.

PROG:STEP_T:DWEL sets the step execution time of the program template.

Command

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

Command (Template)

```
PROGram[:SELected]:STEP_T:DWELl <numeric>
PROGram[:SELected]:STEP_T:DWELl?
```

Parameter

Value:	0.1 to 360000	Step execution time (1 s by default)
	A SCPI error (-222, “Data out of range”) occurs if outside the range.	
Unit:	S	Seconds (when omitted)
	MIN	Minutes
	HR	Hours

(Example)To set the execution time of step 3 to 5 minutes

```
PROG:STEP3:DWEL 5MIN
```

Response

Returns the step time in NR3 format in response to PROG:STEP<n>:DWEL?/ PROG:STEP_T:DWEL?.

Unit: S

PROG:STEP<n>:TRIGIN/ PROG:STEP_T:TRIGIN

Sets the step's trigger signal input.

Steps with this set to on pauses before starting the step and waits for a trigger.

PROG:STEP<n>:TRIGIN sets the trigger signal input of step n. Use <n> to specify the step number. Step numbers start at 0.

PROG:STEP_T:TRIGIN sets the trigger signal input of the program template.

Command

```
PROGram[:SELected]:STEP<n>:TRIGIN <boolean>
PROGram[:SELected]:STEP<n>:TRIGIN?
```

Command (Template)

```
PROGram[:SELected]:STEP_T:TRIGIN <boolean>
PROGram[:SELected]:STEP_T:TRIGIN?
```

Parameter

Value:	ON(1)	Wait for the trigger signal input
	OFF(0)	Do not wait for the trigger signal input (default)

(Example)

```
PROG:STEP0:TRIGIN ON
```

Response

Returns the trigger signal input setting in NR1 format in response to PROG:-STEP<n>:TRIGIN?/ PROG:STEP_T:TRIGIN?.

PROG:STEP<n>:TRIGOUT/ PROG:STEP_T:TRIGOUT

Sets the step's trigger signal output.

PROG:STEP<n>:TRIGOUT sets the trigger signal output of step n. Use <n> to specify the step number. Step numbers start at 0.

PROG:STEP_T:TRIGOUT sets the trigger signal output of the program template.

Command

```
PROGram[:SELected]:STEP<n>:TRIGOUT <boolean>
PROGram[:SELected]:STEP<n>:TRIGOUT?
```

Command (Template)

```
PROGram[:SELected]:STEP_T:TRIGOUT <boolean>
PROGram[:SELected]:STEP_T:TRIGOUT?
```

Parameter

Value:	ON(1)	Enable trigger signal output
	OFF(0)	Disable trigger signal output (default)

(Example)

```
PROG:STEP0:TRIGOUT ON
```

Response

Returns the trigger signal output setting in NR1 format in response to PROG:-STEP<n>:TRIGOUT?/ PROG:STEP_T:TRIGOUT?.

PROG:STEP<n>:VOLT/ PROG:STEP_T:VOLT

Sets the voltage and transition to use in the step.

PROG:STEP<n>:VOLT sets the voltage and transition to use in step n. Use <n> to specify the step number. Step numbers start at 0.

PROG:STEP_T:VOLT sets the voltage and transition of the program template.

Command

```
PROGram[:SELected]:STEP<n>:VOLTage <numeric>[,<character>]
PROGram[:SELected]:STEP<n>:VOLTage?
```

Command (Template)

```
PROGram[:SELected]:STEP_T:VOLTage <numeric>[,<character>]
PROGram[:SELected]:STEP_T:VOLTage?
```

Parameter <numeric>

Value: See the VOLT command.

Unit: V

Parameter <character>

Value:	IMMediate	Transition in steps (default)
	RAMP	Transition with a slope

(Example)

```
PROG:STEP0:VOLT 100,RAMP
```

Response

Returns the voltage <NR3> and transition <character> in a comma-separated format in response to PROG:STEP<n>:VOLT?/ PROG:STEP_T:VOLT?.

PROG:UCOD

Sets the user code to identify sequences.

You can view the user code on the panel. To prevent unintentional sequence execution, we recommend that you set unique user codes for each sequence.

Command

```
PROGram[:SELected]:UCODe <NRf>
PROGram[:SELected]:UCODe?
```

Parameter

Value: 0 to 9999 (0 by default)

Settings are reset to default when the PROG:CLE command is sent.

(Example)

```
PROG:UCOD 7890
```

Response

Returns the user code in NR1 format in response to PROG:UCOD?.

CURR

Sets the current.

When the PWR-01 is set so that constant current is controlled externally (CURR:EXT:SOUR CURR), settings performed by this command are invalid.

This command is invalid while a sequence is running.

Command

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

Parameter

Valut: 0 % to 105 % of the rated output current (Maximum by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: A

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

Rated output current

PWR401L	40 A
PWR401ML	20 A
PWR401MH	5 A
PWR401H	1.85 A
PWR801L	80 A
PWR801ML	40 A
PWR801MH	10 A
PWR801H	3.70 A

PWR1201L	120 A
PWR1201ML	60 A
PWR1201MH	15 A
PWR1201H	5.55 A
PWR2001L	200 A
PWR2001ML	100 A

(Example)

```
CURR 2.5
```

Response

eturns the current setting in NR3 format in response to CURR?.

CURR:EXT:RANG

Sets the CC and CV control range that is used during external control.

This command is invalid while a sequence is running.

CURR:EXT:RANG and VOLT:EXT:RANG are aliases.

Command

```
[SOURce:]CURRent:EXTernal:RANGe <character>
[SOURce:]CURRent:EXTernal:RANGe?
```

Parameter

Value:	LOW	A range of 0 V to 5 V is used (default)
	HIGH	A range of 0 V to 10 V is used.

When you send the VOLT:EXT:RANG command, this setting is set to the VOLT:EXT:RANG value.

(Example)

```
CURR:EXT:RANG HIGH
```

Response

Returns the CC and CV control range in character format in response to CURR:EXT:RANG?.

CURR:EXT:SOUR

Sets whether constant current will be controlled externally.

This command is invalid while a sequence is running.

Command

```
[SOURce:]CURRent:EXTernal:SOURce <character>
```

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

Parameter

Value:	NONE	External control is not performed (default).
	VOLTage	External control is performed.

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

(Example)

```
CURR:EXT:SOUR VOLT
```

Response

Returns whether constant current is being controlled externally in character format in response to CURR:EXT:SOUR?.

CURR:LIM:AUTO

Enables or disables the limit on the current setting.

If you enable the limit when the current setting is higher than 95 % of the OCP trip point, the OCP trip point will be set to 105 % of the current setting.

This command is invalid while a sequence is running.

Command

```
[SOURce:]CURRent:LIMit:AUTO <boolean>
```

```
[SOURce:]CURRent:LIMit:AUTO?
```

Parameter

Value:	ON(1)	The current setting limit is enabled (default).
	OFF(0)	The current setting limit is disabled.

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

(Example)

```
CURR:LIM:AUTO ON
```

Response

Returns whether the limit on the current setting is enabled in NR1 format in response to CURR:LIM:AUTO?

Related Command

```
CURR:PROT
```

```
CURR:PROT:DEL
```

CURR:PROT

Sets the overcurrent protection (OCP) value.

This command is invalid while a sequence is running.

Command

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

Parameter

Value: 10 % to 112 % of the rated output current (Maximum by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: A

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

If you enable the current limit (CURR:LIM:AUTO ON) when the current setting is higher than 95 % of the OCP trip point, the OCP trip point setting is changed to 105 % of the voltage setting.

(Example)

```
CURR:PROT 48
```

Response

Returns the OCP value in NR3 format in response to CURR:PROT?.

Related Command

```
OUTP:PROT:CLE
CURR:LIM:AUTO
CURR:PROT:DEL
```

CURR:PROT:DEL

Sets the detection time of OCP activation.

This command is invalid while a sequence is running.

Command

```
[SOURce:]CURRent:PROTection:DELay <numeric>
```

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

Parameter

Value: 0.0 to 2.0 Detection time of OCP activation. (0 s by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: S

(Example)

```
CURR:PROT:DEL 1.5
```

Response

Returns the detection time of OCP activation in NR3 format in response to CURR:PROT:DEL?.

Related Command

```
CURR:PROT
```

CURR:SST:FALL

Sets the soft stop time when the output is turned off.

This command is valid when the startup state when the output is turned on is set to CC priority (SYST:CONF:STAR:PRI CC).

Command

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

Parameter

Value:	0	Disable soft stop (default)
	0.5 to 10.0	Soft stop time
	A SCPI error (-222, “Data out of range”) occurs if outside the range.	
Unit:	S	

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

(Example)

```
CURR:SST:FALL 2500MS
```

Response

Returns the soft stop time in NR3 format in response to CURR:SST:FALL?.

CURR:SST:RISE

Sets the soft start time when the output is turned on.

This command is valid when the startup state when the output is turned on is set to CC priority (SYST:CONF:STAR:PRI CC).

Command

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

Parameter

Value:	0	Disable soft start (default)
	0.5 to 99.9	Soft start time
	A SCPI error (-222, “Data out of range”) occurs if outside the range.	
Unit:	S	

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

(Example)

```
CURR:SST:RISE 2500MS
```

Response

Returns the soft start time in NR3 format in response to CURR:SST:RISE?.

CURR:TRIG

Sets the current value that is applied when a trigger is sent.

When the PWR-01 is set so that constant current is controlled externally (CURR:EXT:SOUR VOLT), settings performed by this command are invalid.

This command is invalid while a sequence is running.

Command

```
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <numeric>
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]?
```

Parameter

Value: See CURR command

Unit: A

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

When you send the CURR command, this setting is set to the CURR value.

(Example)

```
CURR:TRIG 2.5
```

Response

Returns the current setting to be changed with a trigger in NR3 format in response to CURR:TRIG?.

RES

Sets the internal resistance.

If the variable internal resistance feature (factory options) is in use, this command is valid.

If you are not using the variable internal resistance feature, set this to 0.

Command

```
[SOURce:]RESistance <numeric>
[SOURce:]RESistance?
```

Parameter

Value: L/ ML type 0 to rated output voltage/ rated output current
MH/ H type 0 to rated output voltage/ rated output current x 0.75
(0 by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: OHM

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

(Example)

```
RES 0.16
```

Response

Returns the internal resistance setting in NR3 format in response to RES?.

VOLT

Sets the voltage.

When the PWR-01 is set so that constant voltage is controlled externally (VOLT:EXT:SOUR VOLT), settings performed by this command are invalid.

This command is invalid while a sequence is running.

Command

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

Parameter

Value: 0 % to 105 % of the rated output voltage (MAXimum by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: V

Rated output voltage

PWR401L	40 V
PWR401ML	80 V
PWR401MH	240 V
PWR401H	650 V
PWR801L	40 V
PWR801ML	80 V
PWR801MH	240 V
PWR801H	650 V

PWR1201L	40 V
PWR1201ML	80 V
PWR1201MH	240 V
PWR1201H	650 V
PWR2001L	40 V
PWR2001L	80 V

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

```
VOLT 120
```

Response

Returns the voltage setting in NR3 format in response to VOLT?.

VOLT:EXT:RANG

Sets the CC and CV control range that is used during external control.

This command is invalid while a sequence is running.

CURR:EXT:RANG and VOLT:EXT:RANG are aliases.

Command

```
[SOURce:]VOLTage:EXTernal:RANGe <character>
[SOURce:]VOLTage:EXTernal:RANGe?
```

Parameter

Value:	LOW	A range of 0 V to 5 V is used. (default)
	HIGH	A range of 0 V to 10 V is used.

When you send the CURR:EXT:RANG command, this setting is set to the CURR:EXT:RANG value.

(Example)

```
VOLT:EXT:RANG HIGH
```

Response

Returns the CC and CV control range in CHAR format in response to VOLT:EXT:RANG?.

VOLT:EXT:SOUR

Sets whether constant voltage will be controlled externally.

This command is invalid while a sequence is running.

Command

```
[SOURce:]VOLTage:EXTernal:SOURce <character>
```

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

Parameter

Value:	NONE	External control is not performed (default).
	VOLTage	External control is performed.

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

(Example)

```
VOLT:EXT:SOUR VOLT
```

Response

Returns whether constant voltage is being controlled externally in CHAR format in response to VOLT:EXT:SOUR?.

VOLT:LIM:AUTO

Set whether to limit the voltage setting so that it does not exceed the OVP setting or become lower than the UVP setting.

If set to off (not limit), the UVP is disabled.

If you enable the limit when the voltage setting is higher than 95 % of the OVP trip point, the OVP trip point will be set to 105 % of the voltage setting.

If you enable the limit when the UVP setting is higher than the voltage setting, the UVP setting will be equal to the voltage setting.

This command is invalid while a sequence is running.

Command

```
[SOURce:]VOLTage:LIMit:AUTO <boolean>
```

```
[SOURce:]VOLTage:LIMit:AUTO?
```

Parameter

Value:	ON(1)	The voltage setting limit is enabled (default).
	OFF(0)	The voltage setting limit is disabled.

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

(Example)

```
VOLT:LIM:AUTO ON
```

Response

Returns whether the voltage setting limit is enabled in NR1 format in response to VOLT:LIM:AUTO?.

Related Command

```
VOLT:PROT
```

```
VOLT:LIM:LOW
```

VOLT:LIM:LOW

Sets the undervoltage limit (UVL) trip point.

This command is invalid while a sequence is running.

Command

```
[SOURce:]VOLTage:LIMit:LOWer <numeric>
[SOURce:]VOLTage:LIMit:LOWer?
```

Parameter

Value: 0 % to 105 % of the rated output voltage (UVL; 0 V by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: V

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

If you enable the voltage setting limit (VOLT:LIM:AUTO ON) when the UVP trip point is lower than the voltage setting, the UVP trip point setting is changed to the voltage setting.

(Example)

```
VOLT:LIM:LOW 4.8
```

Response

Returns the UVL trip point in NR3 format in response to VOLT:LIM:LOW?.

Related Command

```
VOLT:LIM:AUTO
```

VOLT:PROT

Sets the overvoltage protection (OVP).

This command is invalid while a sequence is running.

Command

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

Parameter

Value: 10 % to 112 % of the rated output voltage (OVP; MAXimum by default)
A SCPI error (-222, “Data out of range”) occurs if outside the range.

Unit: V

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

If you enable the voltage limit (VOLT:LIM:AUTO ON) when the voltage setting is higher than 95 % of the OVP trip point, the OVP trip point setting is changed to 105 % of the voltage setting.

(Example)

```
VOLT:PROT 48
```

Response

Returns the OVP value in NR3 format in response to VOLT:PROT?.

Related Command

```
OUTP:PROT:CLE
VOLT:LIM:AUTO
```

VOLT:SST:FALL

Sets the soft stop time when the output is turned off.

This command is invalid when the startup state when the output is turned on is set to CC priority (SYST:CONF:STAR:PRI CC).

Command

```
[SOURce:]VOLTage:SSTart:FALL <numeric>
[SOURce:]VOLTage:SSTart:FALL?
```

Parameter

Value:	0	Disable soft stop (default)
	0.5 to 10.0	Soft stop time
	A SCPI error (-222, “Data out of range”) occurs if outside the range.	
Unit:	S	

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

(Example)

```
VOLT:SST:FALL 2500MS
```

Response

Returns the soft stop time in NR3 format in response to VOLT:SST:FALL?.

VOLT:SST:RISE

Sets the soft start time when the output is turned on.

This command is invalid when the startup state when the output is turned on is set to CC priority (SYST:CONF:STAR:PRI CC).

Command

```
[SOURce:]VOLTage:SSTart:RISE <numeric>
[SOURce:]VOLTage:SSTart:RISE?
```

Parameter

Value:	0	Disable soft start (default)
	0.5 to 10.0	Soft start time
	A SCPI error (-222, “Data out of range”) occurs if outside the range.	
Unit:	S	

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

(Example)

```
VOLT:SST:RISE 2500MS
```

Response

Returns the soft start time in NR3 format in response to VOLT:SST:RISE?.

VOLT:TRIG

Sets the voltage that is applied when a trigger is sent.

When the PWR-01 is set so that constant voltage is controlled externally (VOLT:EXT:SOUR VOLT), settings performed by this command are invalid.

This command is invalid while a sequence is running.

Command

```
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <numeric>
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
```

Parameter

Value: See VOLT command

Unit: V

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

When you send the VOLT command, this setting is set to the VOLT value.

(Example)

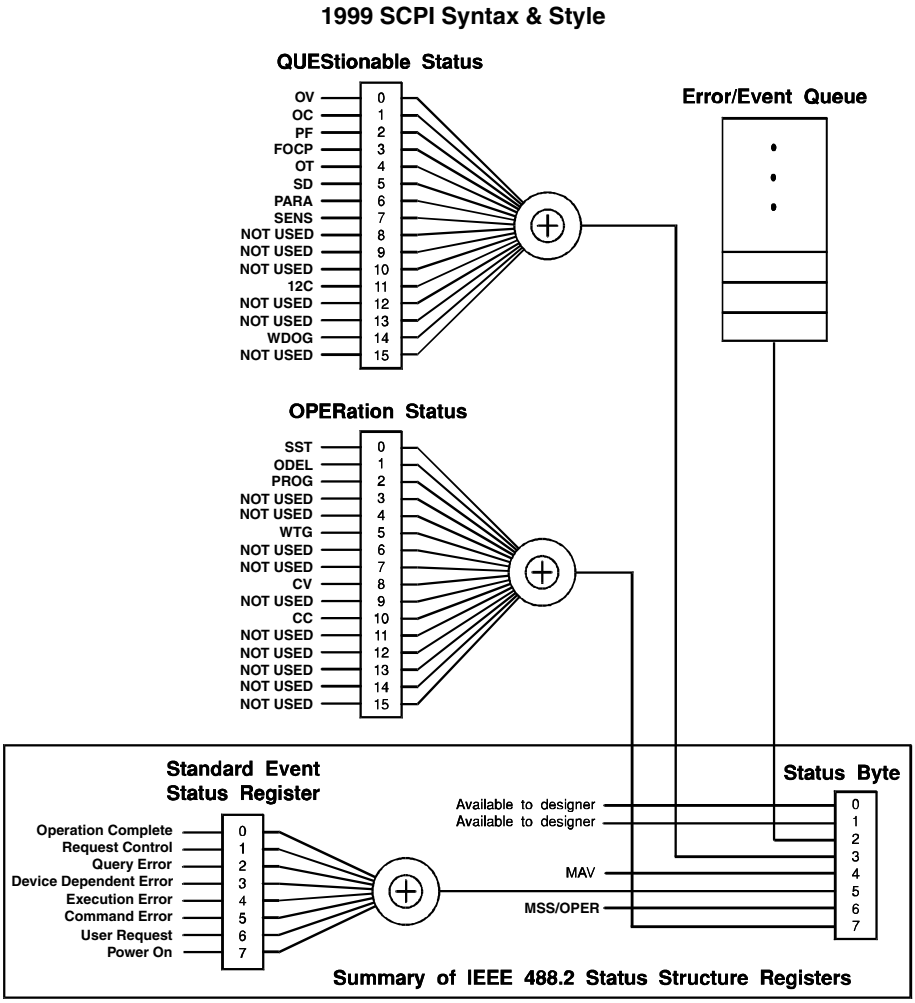
```
VOLT:TRIG 120
```

Response

Returns the voltage value that is applied when a trigger is received in NR3 format in response to VOLT:TRIG?.

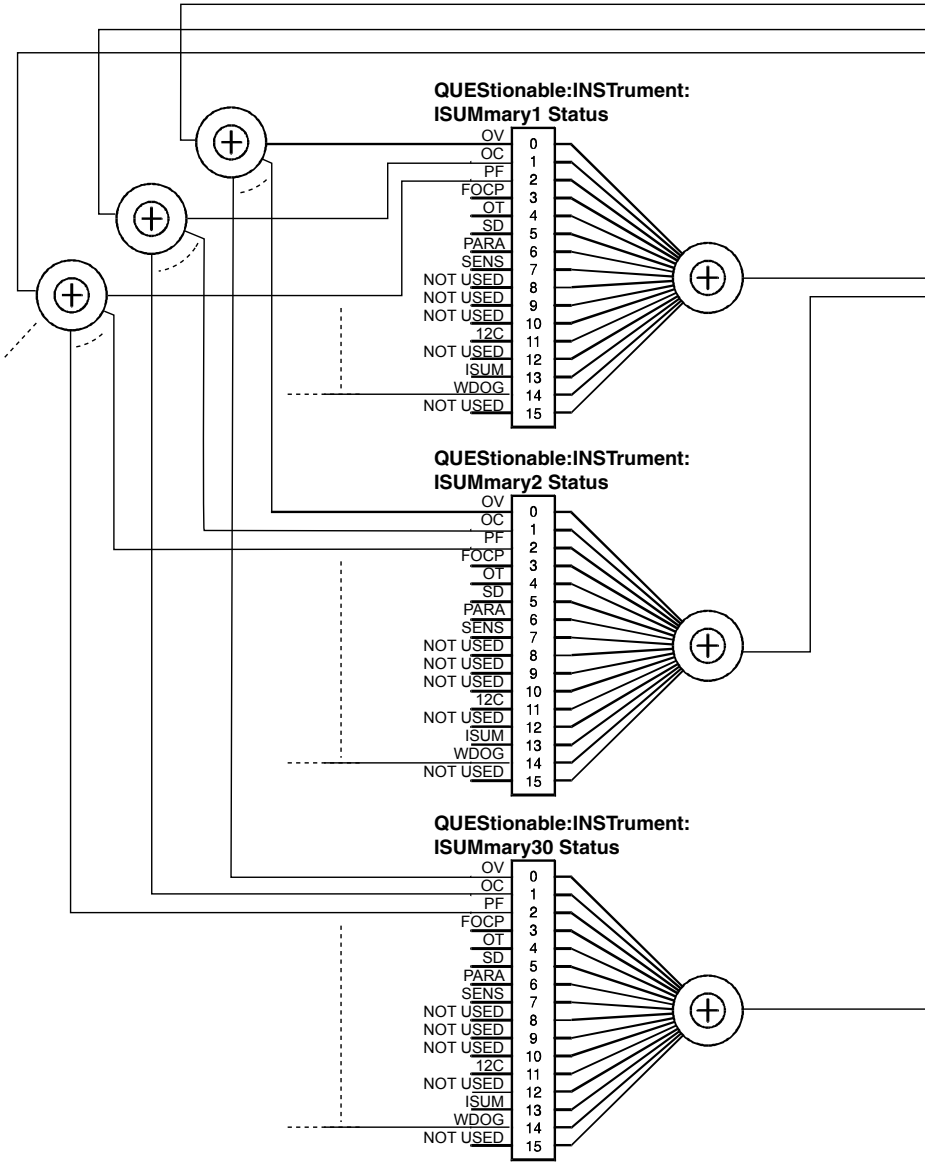
Register Structure

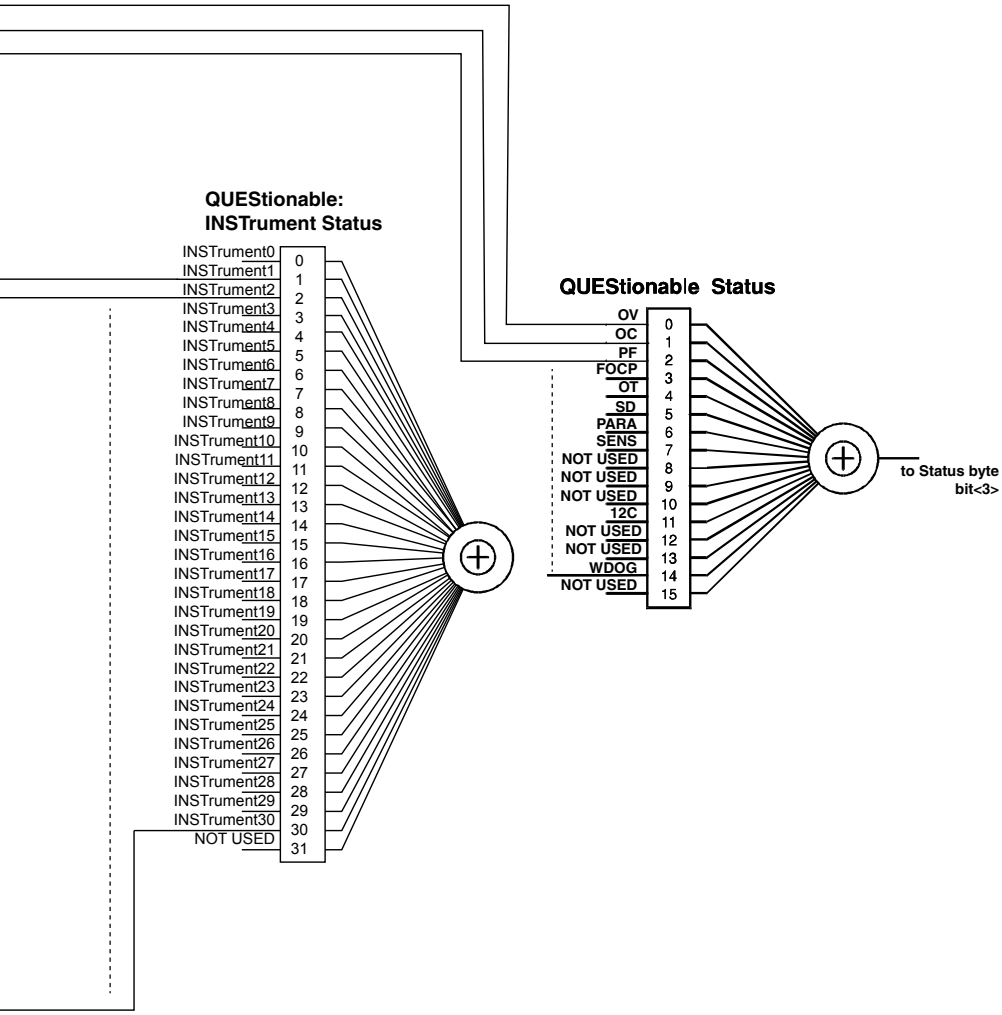
The SCPI status register structure is shown in the figure below. The character "+" represents the logical sum of the register bits.



■ When multichannel is in use

The SCPI status register structure is shown in the figure below. The character "+" represents the logical sum of the register bits. The OPERation Status, OPERation:INSTrument Status, and OPERation:INSTrument:ISUMmary Status registers all operate the same way.




QUEStionable:
INSTrument Status
INSTrument0
INSTrument1
INSTrument2
INSTrument3
INSTrument4
INSTrument5
INSTrument6
INSTrument7
INSTrument8
INSTrument9
INSTrument10
INSTrument11
INSTrument12
INSTrument13
INSTrument14
INSTrument15
INSTrument16
INSTrument17
INSTrument18
INSTrument19
INSTrument20
INSTrument21
INSTrument22
INSTrument23
INSTrument24
INSTrument25
INSTrument26
INSTrument27
INSTrument28
INSTrument29
INSTrument30
NOT USED
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
QUEStionable Status
OV
OC
PF
FOCP
OT
SD
PARA
SENS
NOT USED
NOT USED
NOT USED
12C
NOT USED
NOT USED
WDOG
NOT USED
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
to Status byte
bit<3>

Architecture

IEEE488.2 and SCPI registers are used for the status reports.

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

-> Tutorial “Status Monitoring” (p.221)

CONDition register

The transition of the CONDition register is automatic and reflects the condition of the PWR-01 in real-time. Reading this register does not affect the contents.

EVENt register

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

ENABle register

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

Transition filter

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

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

If both the positive filter and the 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 IEEE488.1 standard. The status byte register can be read using IEEE488.1 serial polling or the IEEE488.2 common command *STB?.

When serial polling is carried out, bit 6 responds with the 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, or RQS.

Bit	Bit weight	Bit name	Description
0	1	Reserved	Reserved for future use by the IEEE488. 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 PWR-01 is ready to output 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 a 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 PWR-01 operation. All bits of the event status register are set by the error event queue.

The register is defined by the IEEE488.2 standard and is controlled by the IEEE488.2 common commands *ESE, *ESE?, and *ESR?.

See SYST:ERR? for the descriptions of the errors.

Bit	Bit weight	Bit name	Description	Error code
0	1	Operation Complete(OPC)	Set when an *OPC command is received and all operations in standby are complete.	-800 to -899
1	2	Request Control (RQC)	Not used	--
2	4	Query Error(QYE)	Set when an attempt is made to read data from the output queue when there is no output or the error queue is in wait status. Indicates that there is no data in the error 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 PWR-01 evaluates that the program data following the header is outside the formal input range or does not match the performance of the PWR-01. This indicates that a valid SCPI command may not be executed correctly depending on the conditions of the PWR-01.	-200 to -299
5	32	Command Error(CME)	ISet when an IEEE 488.2 syntax error is detected, 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)	Not used	--
7	128	Power On(PON)	Set when the power is turned on.	--
8-15		Reserved	Not used	--

OPERation status register

The OPERation status register is a 16-bit register that contains conditions that are part of the PWR-01 normal operations.

Bit	Bit weight	Bit name	Description
0	1	Soft Start(SST)	Indicates whether a soft start or soft stop is in use.
1	2	OUTPut DELAY(ODEL)	Indicates whether the output delay is in use.
2	4	Sequnce program(PROG)	Indicates whether a sequence is being performed.
3	8	NOT USED	--
4	16	NOT USED	--
5	32	wating for trigger(WTG)	Indicates whether the PWR-01 is in a trigger (TRIG) wait state after an INIT is sent.*1
6	64	NOT USED	--
7	128	NOT USED	--
8	256	CV	CV output
9	512	NOT USED	--
10	1024	CC	CC output
11	2048	NOT USED	--
12	4096	NOT USED	--
13	8192	NOT USED	--
14	16384	NOT USED	--
15	32768	NOT USED	Always zero

*1: If the PWR-01 is waiting for a hardware trigger in a sequence step, the PWR-01 will not enter a trigger wait state.

STAT:OPER

Queries the event of the OPERation status register.

A query clears the contents of the register.

Command

```
STATus:OPERation[:EVENt]?
```

Response

Returns the event of the OPERation status register in NR1 format.

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

Returns the condition of the OPERation status register in NR1 format.

STAT:OPER:ENAB

Sets the enable register of the OPERation status register.

Command

```
STATus:OPERation:ENABle <NRf>
STATus:OPERation:ENABle?
```

Parameter

Value: 0 to 65535

Response

Returns the enable register setting of the OPERation status register in NR1 format.

STAT:OPER:NTR

Sets the negative transition of the OPERation status register.

Command

```
STATus:OPERation:NTRansition <NRf>
STATus:OPERation:NTRansition?
```

Parameter

Value: 0 to 65535

Response

Returns the negative transition of the OPERation status register in NR1 format.

STAT:OPER:PTR

Sets the positive transition of the OPERation status register.

Command

```
STATus:OPERation:PTRansition <NRf>
STATus:OPERation:PTRansition?
```

Parameter

Value: 0 to 65535

Response

Returns the positive transition of the OPERation status register in NR1 format.

OPERation:INSTrument subregister

This is the subregister (32 bits) of bit 13 of the OPERation status register.

Bit	Bit weight	Bit name	Description
0	1	INSTrument0	Summary bit of channel 0(OPER:INST:ISUM0)
1	2	INSTrument1	Summary bit of channel 1(OPER:INST:ISUM1)
2	4	INSTrument2	Summary bit of channel 2(OPER:INST:ISUM2)
3	8	INSTrument3	Summary bit of channel 3(OPER:INST:ISUM3)
4	16	INSTrument4	Summary bit of channel 4(OPER:INST:ISUM4)
5	32	INSTrument5	Summary bit of channel 5(OPER:INST:ISUM5)
6	64	INSTrument6	Summary bit of channel 6(OPER:INST:ISUM6)
7	128	INSTrument7	Summary bit of channel 7(OPER:INST:ISUM7)
8	256	INSTrument8	Summary bit of channel 8(OPER:INST:ISUM8)
9	512	INSTrument9	Summary bit of channel 9(OPER:INST:ISUM9)
10	1024	INSTrument10	Summary bit of channel 10(OPER:INST:ISUM10)
11	2048	INSTrument11	Summary bit of channel 11(OPER:INST:ISUM11)
12	4096	INSTrument12	Summary bit of channel 12(OPER:INST:ISUM12)
13	8192	INSTrument13	Summary bit of channel 13(OPER:INST:ISUM13)
14	16384	INSTrument14	Summary bit of channel 14(OPER:INST:ISUM14)
15	32768	INSTrument15	Summary bit of channel 15(OPER:INST:ISUM15)
16	65536	INSTrument16	Summary bit of channel 16(OPER:INST:ISUM16)
17	131072	INSTrument17	Summary bit of channel 17(OPER:INST:ISUM17)
18	262144	INSTrument18	Summary bit of channel 18(OPER:INST:ISUM18)
19	524288	INSTrument19	Summary bit of channel 19(OPER:INST:ISUM19)
20	1048576	INSTrument20	Summary bit of channel 20(OPER:INST:ISUM20)
21	2097152	INSTrument21	Summary bit of channel 21(OPER:INST:ISUM21)
22	4194304	INSTrument22	Summary bit of channel 22(OPER:INST:ISUM22)
23	8388608	INSTrument23	Summary bit of channel 23(OPER:INST:ISUM23)
24	16777216	INSTrument24	Summary bit of channel 24(OPER:INST:ISUM24)
25	33554432	INSTrument25	Summary bit of channel 25(OPER:INST:ISUM25)
26	67108864	INSTrument26	Summary bit of channel 26(OPER:INST:ISUM26)
27	134217728	INSTrument27	Summary bit of channel 27(OPER:INST:ISUM27)
28	268435456	INSTrument28	Summary bit of channel 28(OPER:INST:ISUM28)
29	536870912	INSTrument29	Summary bit of channel 29(OPER:INST:ISUM29)
30	1073741824	INSTrument30	Summary bit of channel 30(OPER:INST:ISUM30)
31	2147483648	NOT USED	Always zero

STAT:OPER:INST

Queries the event of the OPERation:INSTrument subregister.

A query clears the contents of the subregister.

Command

```
STATus:OPERation:INSTrument[:EVENt]?
```

Response

Returns the event of the OPERation:INSTrument subregister in NR1 format.

STAT:OPER:INST:COND

Queries the condition of the OPERation:INSTrument subregister.

A query does not clear the contents of the subregister.

Command

```
STATus:OPERation:INSTrument:CONDition?
```

Response

Returns the condition of the OPERation:INSTrument subregister in NR1 format.

STAT:OPER:INST:ENAB

Sets the enable register of the OPERation:INSTrument subregister.

Command

```
STATus:OPERation:INSTrument:ENABle <NRf>
STATus:OPERation:INSTrument:ENABle?
```

Parameter

Value: 0 to 2147483647

Response

Returns the enable register setting of the OPERation:INSTrument subregister in NR1 format.

STAT:OPER:INST:NTR

Sets the negative transition of the OPERation:INSTrument subregister.

Command

```
STATus:OPERation:INSTrument:NTRansition <NRf>
STATus:OPERation:INSTrument:NTRansition?
```

Parameter

Value: 0 to 2147483647

Response

Returns the negative transition of the OPERation:INSTrument subregister in NR1 format.

STAT:OPER:INST:PTR

Sets the positive transition of the OPERation:INSTrument subregister.

Command

```
STATus:OPERation:INSTrument:PTRansition <NR1>
STATus:OPERation:INSTrument:PTRansition?
```

Parameter

Value: 0 to 2147483647

Response

Returns the positive transition of the OPERation:INSTrument subregister in NR1 format.

OPERation:INSTrument:ISUMmary<n> subregister

This is the subregister of bit <n> of the OPERation:INSTrument subregister. This is a 16-bit register that contains information about the normal operating conditions of the PWR-01 whose channel you specified.

Use <n> to specify the channel number.

For example, channel 2 is specified as "OPER:INST:ISUM2".

Bit	Bit weight	Bit name	Description
0	1	Soft Start(SST)	Indicates whether a soft start or soft stop is in use.
1	2	OUTPut DELEY(ODEL)	Indicates whether the output delay is in use.
2	4	Sequnce program(PROG)	Indicates whether a sequence is being performed.
3	8	NOT USED	--
4	16	NOT USED	--
5	32	wating for trigger(WTG)	Indicates whether the PWR-01 is in a trigger (TRIG) wait state after an INIT is sent. *1
6	64	NOT USED	--
7	128	NOT USED	--
8	256	CV	CV output
9	512	NOT USED	--
10	1024	CC	CC output
11	2048	NOT USED	--
12	4096	NOT USED	--
13	8192	NOT USED	--
14	16384	NOT USED	--
15	32768	NOT USED	Always zero

*1: If the PWR-01 is waiting for a hardware trigger in a sequence step, the PWR-01 will not enter a trigger wait state.

STAT:OPER:INST:ISUM<n>

Queries the event of the OPERation:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:OPER:INST:ISUM2" command.

A query clears the contents of the subregister.

Command

```
STATus:OPERation:INSTrument:ISUMmary<n>[:EVENt]?
```

Response

Returns the event of the OPERation:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:OPER:INST:ISUM<n>:COND

Queries the condition of the OPERation:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:OPER:INST:ISUM2:COND" command.

A query does not clear the contents of the subregister.

Command

```
STATus:OPERation:INSTrument:ISUMmary<n>:CONDition?
```

Response

Returns the condition of the OPERation:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:OPER:INST:ISUM<n>:ENAB

Sets the enable register of the OPERation:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:OPER:INST:ISUM2:ENAB" command.

Command

```
STATus:OPERation:INSTrument:ISUMmary<n>:ENABle <NRf>
STATus:OPERation:INSTrument:ISUMmary<n>:ENABle?
```

Parameter

Value: 0 to 65535

Response

Returns the enable register setting of the OPERation:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:OPER:INST:ISUM<n>:NTR

Sets the negative transition of the OPERation:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:OPER:INST:ISUM2:NTR" command.

Command

```
STATus:OPERation:INSTrument:ISUMmary<n>:NTRansition <NRf>
STATus:OPERation:INSTrument:ISUMmary<n>:NTRansition?
```

Parameter

Value: 0 to 32767

Response

Returns the negative transition of the OPERation:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:OPER:INST:ISUM<n>:PTR

Sets the positive transition of the OPERation:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:OPER:INST:ISUM2:PTR" command.

Command

```
STATus:OPERation:INSTrument:ISUMmary<n>:PTRansition <NRf>
STATus:OPERation:INSTrument:ISUMmary<n>:PTRansition?
```

Parameter

Value: 0 to 32767

Response

Returns the positive transition of the OPERation:INSTrument:ISUMmary<n> subregister in NR1 format.

QUEStionable status register

The QUEStionable status register is a 16-bit register that stores information related to the questionable events and status during PWR-01 operation.These register bits may indicate problems with the measured data of the PWR-01.

Bit	Bit weight	Bit name	Description
0	1	OVP (OV)	Overvoltage protection has been activated
1	2	OCP (OC)	Overcurrent protection has been activated
2	4	AC (PF)	Low AC input protection has been activated
3	8	Front Panel OCP (FOCP)	Front panel overcurrent protection has been activated
4	16	Over Temperature Protection (OT)	Overheat protection has been activated
5	32	Shut Down protection (SD)	Shut down
6	64	PARAllel protection (PARA)	Incorrect parallel operation connection has been activated
7	128	SENSing protection (SENS)	Incorrect sensing connection protection has been activated
8	256	NOT USED	--
9	512	NOT USED	--
10	1024	NOT USED	--
11	2048	Internal commuication failure (12C)	--
12	4096	NOT USED	--
13	8192	NOT USED	--
14	16384	WatchDOG protection (WDOG)	Communication monitor has been activated
15	32768	NOT USED	Always zero

STAT:QUES

Queries the event of the QUEStionable status register.

A query clears the contents of the register.

Command

```
STATus:QUEStionable[:EVENt]?
```

Response

Returns the event of the QUEStionable status register in NR1 format.

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

Returns the condition of the QUEStionable status register in NR1 format.

STAT:QUES:ENAB

Sets the enable register of the QUEStionable status register.

Command

```
STATus:QUEStionable:ENABle <NRf>
STATus:QUEStionable:ENABle?
```

Parameter

Value:0 to 65535

Response

Returns the enable register setting of the QUEStionable status register in NR1 format.

STAT:QUES:NTR

Sets the negative transition of the QUEStionable status register.

Command

```
STATus:QUEStionable:NTRansition <NRf>
STATus:QUEStionable:NTRansition?
```

Parameter

Value:0 to 65535

Response

Returns the negative transition of the QUEStionable status register in NR1 format.

STAT:QUES:PTR

Sets the positive transition of the QUEStionable status register.

Command

```
STATus:QUEStionable:PTRansition <NRf>
STATus:QUEStionable:PTRansition?
```

Parameter

Value:0 to 65535

Response

Returns the positive transition of the QUEStionable status register in NR1 format.

QUEStionable:INSTrument subresigter

This is the subregister (32 bits) of bit 13 of the QUEStionable status register.

Bit	Bit weight	Bit name	Description
0	1	INSTrument0	Summary bit of channel 0(OPER:INST:ISUM0)
1	2	INSTrument1	Summary bit of channel 1(OPER:INST:ISUM1)
2	4	INSTrument2	Summary bit of channel 2(OPER:INST:ISUM2)
3	8	INSTrument3	Summary bit of channel 3(OPER:INST:ISUM3)
4	16	INSTrument4	Summary bit of channel 4(OPER:INST:ISUM4)
5	32	INSTrument5	Summary bit of channel 5(OPER:INST:ISUM5)
6	64	INSTrument6	Summary bit of channel 6(OPER:INST:ISUM6)
7	128	INSTrument7	Summary bit of channel 7(OPER:INST:ISUM7)
8	256	INSTrument8	Summary bit of channel 8(OPER:INST:ISUM8)
9	512	INSTrument9	Summary bit of channel 9(OPER:INST:ISUM9)
10	1024	INSTrument10	Summary bit of channel 10(OPER:INST:ISUM10)
11	2048	INSTrument11	Summary bit of channel 11(OPER:INST:ISUM11)
12	4096	INSTrument12	Summary bit of channel 12(OPER:INST:ISUM12)
13	8192	INSTrument13	Summary bit of channel 13(OPER:INST:ISUM13)
14	16384	INSTrument14	Summary bit of channel 14(OPER:INST:ISUM14)
15	32768	INSTrument15	Summary bit of channel 15(OPER:INST:ISUM15)
16	65536	INSTrument16	Summary bit of channel 16(OPER:INST:ISUM16)
17	131072	INSTrument17	Summary bit of channel 17(OPER:INST:ISUM17)
18	262144	INSTrument18	Summary bit of channel 18(OPER:INST:ISUM18)
19	524288	INSTrument19	Summary bit of channel 19(OPER:INST:ISUM19)
20	1048576	INSTrument20	Summary bit of channel 20(OPER:INST:ISUM20)
21	2097152	INSTrument21	Summary bit of channel 21(OPER:INST:ISUM21)
22	4194304	INSTrument22	Summary bit of channel 22(OPER:INST:ISUM22)
23	8388608	INSTrument23	Summary bit of channel 23(OPER:INST:ISUM23)
24	16777216	INSTrument24	Summary bit of channel 24(OPER:INST:ISUM24)
25	33554432	INSTrument25	Summary bit of channel 25(OPER:INST:ISUM25)
26	67108864	INSTrument26	Summary bit of channel 26(OPER:INST:ISUM26)
27	134217728	INSTrument27	Summary bit of channel 27(OPER:INST:ISUM27)
28	268435456	INSTrument28	Summary bit of channel 28(OPER:INST:ISUM28)
29	536870912	INSTrument29	Summary bit of channel 29(OPER:INST:ISUM29)
30	1073741824	INSTrument30	Summary bit of channel 30(OPER:INST:ISUM30)
31	2147483648	NOT USED	Always zero

STAT:QUES:INST

Queries the event of the QUEStionable:INSTrument subregister.

A query clears the contents of the register.

Command

```
STATus:QUEStionable:INSTrument[:EVENt]?
```

Response

Returns the event of the QUEStionable:INSTrument subregister in NR1 format.

STAT:QUES:INST:COND

Queries the condition of the QUEStionable:INSTrument subregister.

A query does not clear the contents of the register.

Command

```
STATus:QUEStionable:INSTrument:CONDition?
```

Response

Returns the condition of the QUEStionable:INSTrument subregister in NR1 format.

STAT:QUES:INST:ENAB

Sets the enable register of the QUEStionable:INSTrument subregister.

Command

```
STATus:QUEStionable:INSTrument:ENABle <NRf>
STATus:QUEStionable:INSTrument:ENABle?
```

Parameter

Value:0 to 2147483647

Response

Returns the enable register setting of the QUEStionable:INSTrument subregister in NR1 format.

STAT:QUES:INST:NTR

Sets the negative transition of the QUEStionable:INSTrument subregister.

Command

```
STATus:QUEStionable:INSTrument:NTRansition <NRf>
STATus:QUEStionable:INSTrument:NTRansition?
```

Parameter

Value:0 to 2147483647

Response

Returns the negative transition of the QUEStionable:INSTrument subregister in NR1 format.

STAT:QUES:INST:PTR

Sets the positive transition of the QUEStionable:INSTrument subregister.

Command

```
STATus:QUEStionable:INSTrument:PTRansition <NRf>
STATus:QUEStionable:INSTrument:PTRansition?
```

Parameter

Value:0 to 2147483647

Response

Returns the positive transition of the QUEStionable:INSTrument subregister in NR1 format.

QUEStionable:INSTrument:ISUMmary<n> subregister

This is the subregister of bit <n> of the QUEStionable:INSTrument subregister. This is a 16-bit register that stores information related to the status of and the questionable events that occur during operation of the PWR-01 whose channel you specified.

Use <n> to specify the channel number.

For example, channel 2 is specified as "QUES:INST:ISUM2".

The QUEStionable status register bits may indicate that there are problems with the PWR-01's measured data.

Bit	Bit weight	Bit name	Description
0	1	OVP (OV)	Overvoltage protection has been activated
1	2	OCP (OC)	Overcurrent protection has been activated
2	4	AC (PF)	Low AC input protection has been activated
3	8	Front Panel OCP (FOCP)	Front panel overcurrent protection has been activated
4	16	Over Temperature Protection (OT)	Overheat protection has been activated
5	32	Shut Down protection (SD)	Shut down
6	64	PARAllel protection (PARA)	Incorrect parallel operation connection has been activated
7	128	SENSing protection (SENS)	Incorrect sensing connection protection has been activated
8	256	NOT USED	--
9	512	NOT USED	--
10	1024	NOT USED	--
11	2048	Internal commuication failure (12C)	--
12	4096	NOT USED	--
13	8192	NOT USED	--
14	16384	WatchDOG protection (WDOG)	Communication monitor has been activated
15	32768	NOT USED	Always zero

STAT:QUES:INST:ISUM<n>

Queries the event of the QUEStionable:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:QUES:INST:ISUM2" command.

A query clears the contents of the register.

Command

```
STATus:QUEStionable:INSTrument:ISUMmary<n>[:EVENt]?
```

Response

Returns the event of the QUEStionable:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:QUES:INST:ISUM<n>:COND

Queries the condition of the QUEStionable:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:QUES:INST:ISUM2:COND" command.

A query does not clear the contents of the register.

Command

```
STATus:QUEStionable:INSTrument:ISUMmary<n>:CONDition?
```

Response

Returns the condition of the QUEStionable:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:QUES:INST:ISUM<n>:ENAB

Sets the enable register of the QUEStionable:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:QUES:INST:ISUM2:ENAB" command.

Command

```
STATus:QUEStionable:INSTrument:ISUMmary<n>:ENABle <NRf>
STATus:QUEStionable:INSTrument:ISUMmary<n>:ENABle?
```

Parameter

Value:0 to 65535

Response

Returns the enable register setting of the QUEStionable:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:QUES:INST:ISUM<n>:NTR

Sets the negative transition of the QUEStionable:INSTrument:ISUMmary<n> sub-register.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:QUES:INST:ISUM2:NTR" command.

Command

```
STATus:QUEStionable:INSTrument:ISUMmary<n>:NTRansition <NRf>
STATus:QUEStionable:INSTrument:ISUMmary<n>:NTRansition?
```

Parameter

Value:0 to 32767

Response

Returns the negative transition of the QUEStionable:INSTrument:ISUMmary<n> subregister in NR1 format.

STAT:QUES:INST:ISUM<n>:PTR

Sets the positive transition of the QUEStionable:INSTrument:ISUMmary<n> subregister.

Use <n> to specify the channel number.

For example, to specify channel 2, send the "STAT:QUES:INST:ISUM2:PTR" command.

Command

```
STATus:QUEStionable:INSTrument:ISUMmary<n>:PTRansition <NR1>
STATus:QUEStionable:INSTrument:ISUMmary<n>:PTRansition?
```

Parameter

Value:0 to 32767

Response

Returns the positive transition of the QUEStionable:INSTrument:ISUMmary<n> subregister in NR1 format.

Preset status

STAT:PRES

Resets the ENABle, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values. 32 bits registers are used by the multichannel function.

Default values (16 bits register):
STATus:ENABle = 0x0000
STATus:PTRansition = 0x7FFF
STATus:NTRansition = 0x0000

Default values (32 bits register):
STATus:ENABle = 0x7FFFFFFF
STATus:PTRansition = 0x7FFFFFFF
STATus:NTRansition = 0x00000000

Command

```
STATus:PRESet
```

SYST:BEEP:STAT

Turns the buzzer on and off.

Command

```
SYSTem:BEEPer:STATe <boolean>
SYSTem:BEEPer:STATe?
```

Parameter

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

(Example)

```
SYST:BEEP:STAT OFF
```

Response

Returns the buzzer setting in NR1 format in response to SYST:BEEP:STAT?.

SYST:COMM:RLST

Sets the operation of the PWR-01 to local or remote.

Command

```
SYSTem:COMMunicate:RLSTate <character>
SYSTem:COMMunicate:RLSTate?
```

Parameter

Value: LOCal Sets the PWR-01 to local mode (Remote Disable; the REMOTE LED turns off).

Disable state enables both panel operations and commands.

This is a substitute command for IEEE488.1 ren FALSE (Remote Disable).

If the communication monitoring timer is set, set the timer to off (OUTP:PROT:WDOG 0) before switching to local mode.

REMote Sets the PWR-01 operation to remote mode.

All panel keys except the LOCAL key are locked.

This is a substitute command for the IEEE488.1 REN (Remote Enable) command and address designation.

RWLock Sets the PWR-01 operation to remote mode.

All panel keys are locked (the LOCAL key is also locked). This is a substitute command for the IEEE488.1 llo (Local Lock Out) command.

(Example)

```
SYST:COMM:RLST REM
```

Response

Returns the operation status of the PWR-01 in character format in response to the SYST:COMM:RLST?.

SYST:CONF:BLE

Sets the bleeder circuit operation.

This command is invalid while a sequence is running.

Command

```
SYSTem:CONFigure:BLEeder <character>
SYSTem:CONFigure:BLEeder?
```

Parameter

Value:	DISable	The bleeder is off.
	NORMal	The normal bleeder is on
	HYPer	The hyper bleeder is on.

(Example)

```
SYST:CONF:BLE HYP
```

Response

Returns the bleeder circuit setting in character format in response to SYST:CONF:BLE?.

SYST:CONF:MAST

Queries the number of units in master-slave parallel operation.

Command

```
SYSTem:CONFigure:MASTer?
```

Response

Returns the number of units in master-slave parallel operation (the master unit is included in this number) in NR1 format in response to SYST:CONF:MAST?.

SYST:CONF:MON:RANG

Sets the range for monitoring the voltage or current externally.

Command

```
SYSTem:CONFigure:MONitor:RANGe <character>
SYSTem:CONFigure:MONitor:RANGe?
```

Parameter

Value:	LOW	A range of 0 V to 5 V is used. (default)
	HIGH	A range of 0 V to 10 V is used.

(Example)

```
SYST:CONF:MON:RANG HIGH
```

Response

Returns the range for voltage and current monitoring in character format in response to SYST:CONF:MON:RANG?.

SYST:CONF:PROT:REC

Sets the output state when a low AC input protection (AC-FAIL) is released.

Command

```
SYSTem:CONFigure:PROTection:RECovery <character>
SYSTem:CONFigure:PROTection:RECovery?
```

Parameter

Value:	SAFE	Output is not turned on automatically. (default)
	AUTO	Output is turned on automatically.

(Example)

```
SYST:CONF:PROT:REC SAFE
```

Response

Returns the output state when a low AC input protection is released in character format in response to SYST:CONF:PROT:REC?.

SYST:CONF:SC1/ SYST:CONF:SC2/ SYST:CONF:SC3

Registers the CONFIG setting item to the panel's SC key.

Command

SC1 key

```
SYSTem:CONFigure:SC1 <NRf>
SYSTem:CONFigure:SC1?
```

SC2 key

```
SYSTem:CONFigure:SC2 <NRf>
SYSTem:CONFigure:SC2?
```

SC3 key

```
SYSTem:CONFigure:SC3 <NRf>
SYSTem:CONFigure:SC3?
```

Parameter <NRf>

Value:	0	CONFIG parameter content not registered (default)
	1 to 99	CF01 to CF99 (invalid if an nonexistent CONFIG parameter number is specified.

(Example)To register CF40 to the SC1 key

```
SYST:CONF:SC1 40
```

Response

Returns the registered CONFIG parameter number in NR3 format in response to SYST:CONF:SC1?/ SYST:CONF:SC2?/ SYST:CONF:SC3?.

SYST:CONF:SLAV:AMM

Sets whether or not the current or power on slave units is displayed on the panel during master-slave parallel operation.

Command

```
SYSTem:CONFigure:SLAVe:AMMeter <boolean>
SYSTem:CONFigure:SLAVe:AMMeter?
```

Parameter

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

(Example)

```
SYST:CONF:SLAV:AMM OFF
```

Response

Returns whether or not the current or power is displayed on the panel in NR1 format in response to the SYST:CONF:SLAV:AMM?.

SYST:CONF:STAR:PRI

Sets the operation mode to be prioritized when the output is turned on.

This command is invalid while a sequence is running.

Setting the slave unit during master-slave parallel operation is invalid.

Command

```
SYSTem:CONFigure:STARtup:PRIority <character>
SYSTem:CONFigure:STARtup:PRIority?
```

Parameter

Value:	CC	CC (constant current) is prioritized.
	CV	CV (constant voltage) is prioritized.

(Example)

```
SYST:CONF:STAR:PRI CC
```

Response

Returns the state that the PWR-01 starts up in when the output state at power-on is set to on in character format in response to SYST:CONF:STAR:PRI?.

SYST:ERR

Reads the oldest error information or event information from the error queue. The error queue can store up to 16 errors.

-> Tutorial “Error Checking” (p.227)

Use the *CLS command to clear the error queue.

Command

```
SYSTem:ERRor[:NEXT]?
```

Response

Returns the oldest error or event information in the error/event queue in response to SYST:ERR? as follows:

(Example) When there are no errors or events

+0"No error"

(Example) When a command that cannot be executed under the current operating conditions has been received

-221,"Settings conflict"

SYST:ERR:COUN

Returns the number of unread errors in the error queue.

Command

```
SYSTem:ERRor:COUNt?
```

Response

Returns the number of unread errors in the error queue in NR1 format in response to SYST:ERR:COUN?.

SYST:ERR:TRAC

Sets whether to display communication errors by performing a debug trace.

If the communication error display is enabled, error numbers, such as Err-100, are shown in the PWR-01 display area.

Communication errors are displayed when the PWR-01 is set to remote mode.

Command

```
SYSTem:ERRor:TRACe <boolean>
SYSTem:ERRor:TRACe?
```

Parameter

Value:	ON(1)	Communication errors are displayed.
	OFF(0)	Communication errors are not displayed. (default)

(Example)

```
SYST:ERR:TRAC ON
```

Response

Returns whether communication errors are displayed in NR1 format in response to SYST:ERR:TRAC?.

SYST:EXT:STATOUT:CC:POL

Sets the polarity of the status output signal for constant-current mode.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:STATOUT:CC:POLarity <character>
SYSTem:EXTernal:STATOUT:CC:POLarity?
```

Parameter

Value:	POSitive	High in CC mode
	NEGative	Low in CC mode (default)

(Example)

```
SYST:EXT:STATOUT:CC:POL NEG
```

Response

Returns the polarity of the status output signal in character format in response to SYST:EXT:STATOUT:CC:POL?.

SYST:EXT:STATOUT:CV:POL

Sets the polarity of the status output signal for constant-voltage mode.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:STATOUT:CV:POLarity <character>
SYSTem:EXTernal:STATOUT:CV:POLarity?
```

Parameter

Value:	POSitive	High in CV mode
	NEGative	Low in CV mode (default)

(Example)

```
SYST:EXT:STATOUT:CV:POL NEG
```

Response

Returns the polarity of the status output signal in character format in response to SYST:EXT:STATOUT:CV:POL?.

SYST:EXT:STATOUT:OUTP:POL

Sets the polarity of the status output signal for output.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:STATOUT:OUTP:POLarity <character>
SYSTem:EXTernal:STATOUT:OUTP:POLarity?
```

Parameter

Value:	POSitive	High when the output is on
	NEGative	Low when the output is on (default)

(Example)

```
SYST:EXT:STATOUT:OUTP:POL NEG
```

Response

Returns the polarity of the status output signal in character format in response to SYST:EXT:STATOUT:OUTP:POL?.

SYST:EXT:STATOUT:PROT:POL

Sets the polarity of the status output signal for protection operation.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:STATOUT:PROTection:POLarity <character>
SYSTem:EXTernal:STATOUT:PROTection:POLarity?
```

Parameter

Value:	POSitive	High when a protection is activated
	NEGative	Low when a protection is activated (default)

(Example)

```
SYST:EXT:STATOUT:PROT:POL NEG
```

Response

Returns the polarity of the status output signal in character format in response to SYST:EXT:STATOUT:PROT:POL?.

SYST:EXT:TRIGIN:POL

Sets the polarity of the trigger signal input.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:TRIGIN:POLarity <character>
```

```
SYSTem:EXTernal:TRIGIN:POLarity?
```

Parameter

Value:	PTRansition	Positive trigger (default)
	NTRansition	Negative trigger

(Example)

```
SYST:EXT:TRIGIN:POL NTR
```

Response

Returns the polarity of the trigger signal input in character format in response to SYST:EXT:TRIGIN:POL?.

SYST:EXT:TRIGOUT:POL

Sets the polarity of the trigger signal output.

This command is invalid while a sequence is running.

Command

```
SYSTem:EXTernal:TRIGOUT:POLarity <character>
SYSTem:EXTernal:TRIGOUT:POLarity?
```

Parameter

Value:	PTRansition	Positive trigger (default)
	NTRansition	Negative trigger

(Example)

```
SYST:EXT:TRIGOUT:POL NTR
```

Response

Returns the polarity of the trigger signal output in character format in response to SYST:EXT:TRIGOUT:POL?.

SYST:KLOC

Sets and releases the panel operation lock (keylock).

Command

```
SYSTem:KLOCk <boolean>
SYSTem:KLOCk?
```

Parameter

Value: ON(1) Key lock is set.
OFF(0) Key lock is released.

(Example)

```
SYSTem:KLOC ON
```

Response

Returns the key lock setting in NR1 format in response to SYST:KLOCk?.

SYST:LOC/ SYST:REM/ SYST:RWL

This is an old style command.

When creating new programs, use SYST:COMM:RLST (p.175).

Command

```
SYSTem:LOCal
SYSTem:REMote
SYSTem:RWLock
```

SYST:SEC:IMM

Sanitizes all contents stored in memory and initializes the panel settings to their factory default conditions.

Command

```
SYSTem:SECurity:IMMediate
```

SYST:VERS

Queries the version of the SCPI specifications to which the PWR-01 conforms.

Command

```
SYSTem:VERSion?
```

Response

Returns 1999.0 in response to SYST:VERS?.

TRIG:PROG

Executes a software trigger for the PROGram trigger subsystem.

Command

```
TRIGger:PROGram[:IMMediate]
```

TRIG:PROG:EXEC

Queries the sequence states (execution state, present repetition count, present step number, elapsed time, estimated time).

Command

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

Response

Returns the execution state (STOP: stopped, WTG: waiting for a trigger, RUN: running) <character>, present repetition count (1 to upper limit 2^{31}) <NR1>, present step number (1 to 64) <NR1>, elapsed time [s] (0 to upper limit 2^{31}) <NR1>, estimated time [s] (0 to upper limit 2^{31}, INF for infinite) <NR1/character> in a comma-separated format in response to TRIG:PROG:EXEC?.

Example when the sequence is stopped

```
STOP,0,0,0,0
```

Example when the sequence is waiting for *TRIG immediately after INIT

```
WTG,0,0,0,36000
```

Example after a 10-hour schedule has been executed, after about 5 s has elapsed, when step 0 is being executed

```
RUN,1,0,5,36000
```

Example after the completion of a 10-hour schedule execution

```
STOP,1,0,36000,36000
```

Example after an infinite schedule has been executed, after about 5 s has elapsed, when step 0 is being executed

```
RUN,1,0,5,INF
```

Example when 60 s has elapsed after execution and step 1 is waiting for a trigger

```
WTG,1,1,60,INF
```

Example when the sequence has been aborted when 10 m has elapsed after execution

```
STOP,1,2,600,INF
```

TRIG:PROG:SOUR

Sets the condition (trigger source) that determines when the PROGram trigger subsystem actually executes a sequence operation after the PWR-01 receives the INIT:PROG command.

Command

```
TRIGger:PROGram:SOURce <character>
```

```
TRIGger:PROGram:SOURce?
```

Parameter

Value:	IMMediate	The sequence starts immediately (default)
	BUS	The PWR-01 waits for a software trigger (use the *TRG, TRIG:PROG, or IEEE488.1 get (Group Execute Trigger), and then the sequence starts.

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

(Example)

```
TRIG:PROG:SOUR BUS
```

Response

Returns the PROGram trigger subsystem's trigger source in character format in response to TRIG:PROG:SOUR?.

TRIG:TRAN

Executes a software trigger for the TRANsient trigger subsystem.

Command

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

TRIG:TRAN:SOUR

Sets the condition (trigger source) that determines when the TRANsient trigger subsystem actually changing the setting after the PWR-01 receives the INIT:TRAN command.

Command

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

Parameter

Value:	IMMediate	The setting is changed immediately (default).
	BUS	The setting is changed when a software trigger is received (use the *TRG, TRIG:TRAN, or IEEE488.1 get (Group Execute Trigger) command to change the setting; default).
	TRIGIN	The setting is changed when a hardware trigger is received.

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

(Example)

```
TRIG:TRAN:SOUR BUS
```

Response

Returns the trigger source of the TRANsient trigger subsystem in character format in response to TRIG:TRAN:SOUR?.

Settings and Measurement

■ Voltage and current

The output voltage and output current are controlled by the VOLTage and CURRent commands. The output ON/OFF state is controlled by the OUTPut command.

```
VOLTage 80   'Set the voltage to 80 V
CURRent 5    'Set the current to 5 A
OUTPut ON    'Turn the output on
```

To set the current to the maximum value, set the value of the CURRent command to MAXimum.

The setting can be set to a value in the range from 0 to 105 % of the rated output current.

```
VOLTage 80   'Set the voltage to 80 V
CURRent MAXimum   'Set the current to the maximum value
```

To set the voltage to the maximum value, set the value of the VOLTage command to MAXimum.

The setting can be set to a value in the range from the value of the VOLTage:LIMit:LOW setting to 105 % of the rated output voltage.

```
CURRent 5    Set the current to 5 A
VOLTage MAXimum   'Set the voltage to the maximum value
```

You can confirm the maximum allowed value that can be specified in the VOLTage and CURRent commands by specifying the MAXimum parameter in the VOLTage? and CURRent? queries, respectively.

```
VOLTage? MAXimum
```

<Read the response>

```
CURRent? MAXimum
```

<Read the response>

The values that the queries above return may vary depending on the VOLTage:PROTection (OVP) and CURRent:PROTection (OCP) settings.

■ Setting the protection functions

The PWR-01 provides overvoltage protection (OVP) and overcurrent protection (OCP) functions that can be configured. The VOLTage:PROTection and CURRent:PROTection commands are used to set the OVP and OCP, respectively.

```
VOLTage:PROTection 50   'Set the OVP to 50 V
CURRent:PROTection 4.5   'Set the OCP to 4.5 A
```

If you want to set the OVP or OCP setting to the maximum or minimum value, you can specify MAXimum or MINimum for the parameter.

```
VOLTage:PROTection MAXimum  'Set the OVP to the maximum value
CURRent:PROTection MAXimum  'Set the OCP to the maximum value
```

The output turns off if an OVP or OCP function is activated.

Eliminate the cause of the alarm, and clear the alarm.

```
OUTPut:PROTection:CLEar
```

■ Measurement

After you have set the output and protection function settings, query the measured values. The PWR-01 can return the measured voltage and the measured current.

To measure the voltage and current, use the following queries.

```
MEASure:VOLTage?  'Query the voltage output
MEASure:CURRent?  'Query the current output
```

The measured voltage and the measured current are updated alternately at 25 ms intervals. The output status is updated alternately at 50 ms intervals. If you query the measured value or the output status at an interval shorter than 50 ms, the previous measured value will be returned.

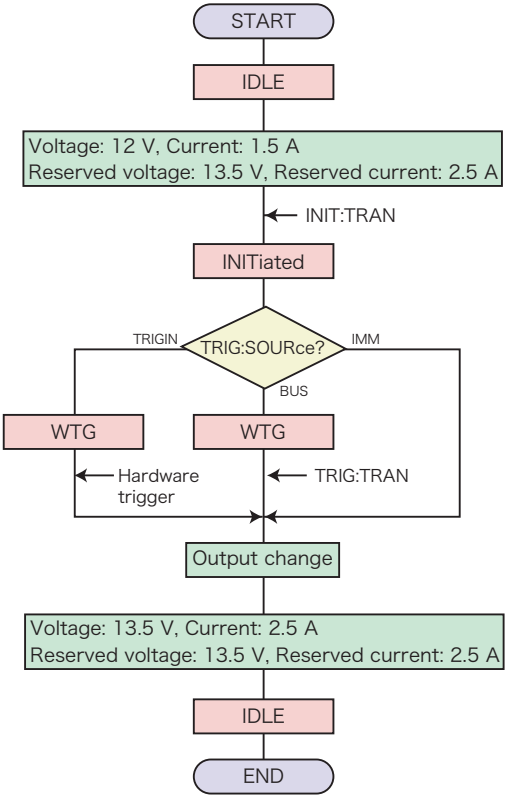
Using Triggers to Change Settings (TRANsient)

PWR-01 has two different trigger subsystems -TRANsient, and PROGram.

The TRANsient group changes the PWR-01 settings.

You can use triggers to synchronize the setting of the voltage and current. This is convenient if you want to synchronize output changes with the action of other instruments such as electronic loads. To reserve triggered settings, use the VOLTage:TRIGgered and CURRent:TRIGgered commands.

```
VOLTage 12.0  'Set the voltage to 12.0 V
CURRent 1.5  'Set the current to 1.5 A
VOLTage:TRIGgered 13.5  'Set the PWR-01 so that the voltage is 13.5 V when triggered
CURRent:TRIGgered 2.5  'Set the PWR-01 so that the current is 2.5 A when triggered
TRIGger:SOURce BUS  'Set the trigger source to BUS
INITiate:TRANsient  'Initiate
TRIGger:TRANsient  'Apply a software trigger
```



The TRIGger:TRANSient:SOURce command sets the trigger source to BUS, TRIGIN, or IMMediate.

The INITiate:TRANsient command makes the trigger subsystem leave the IDLE state and enter the initiated state. When the Trigger Source is set to IMMediate, the transient action is executed immediately, which causes the voltage or current or both to change to a new setting. When the Trigger Source is set to BUS or TRIGIN, the trigger subsystem enters the WTG (Waiting For Trigger) state.

The GLOB:INIT:TRAN command start the trigger function of all channels.

If the trigger source is set to BUS and the TRIGger subsystem is in a WTG state, when a software trigger is received, a change is executed.

TRIGger:TRANsient command applies a software trigger to the TRANsient subsystem.

The *TRG command or the IEEE488.1 get (Group Execute Trigger) command applies a software trigger to all trigger subsystems, if there are other trigger subsystems in the initiated state, their trigger functions will also be executed at the same time.

The GLOB:*TRG commands apply a software trigger to all channels, so if there are other channels that are in the INITiated state, their settings will be changed at the same time.

If the trigger source is set to TRIGIN and the TRIGger subsystem is in a WTG state, when a hardware trigger (pin 7 of the TRIG IN terminal on the rear panel) is received, a change is executed.

When the operation is completed, the TRANsient subsystem returns to the IDLE state again. When an ABORt command or an equivalent command is sent without executing the trigger, the transient action is canceled, and then the TRANsient subsystem returns to the IDLE state.

When the PWR-01 is turned on, all the TRIGger subsystems are in the IDLE state. In this state, the TRIGger subsystem ignores all triggers. If you send one of the following commands, the TRIGger subsystem is switched to the IDLE state, regardless of its current state.

`ABORt`

`*RST`

Device clear (USB interface) or break signal (RS232 interface)

Operation using a software trigger

When ABOR is sent, INIT:TRAN is cancelled. The VOLT:TRIG setting does not change.

The following table shows the responses when the voltage is set to 20 V (VOLT 20) and when the target value to which the voltage will change when a trigger is received is set to 10 V (VOLT:TRIG 10).

	Response	
	VOLT?	VOLT:TRIG?
Immediately after the setting is made.	20 V	10 V
After a trigger is sent.	10 V	10 V
After *RST is sent.	0 V	0 V
When "VOLT 30" (to change the voltage) is sent before sending a trigger	30 V	30 V (cancel)

PWR-01 behavior during synchronized operation

If all trigger sources are set to TRIGIN on all PWR-01s operating in sync and the TRIGger subsystems are in a WTG state, when all PWR-01s receive a hardware trigger (pin 7 of the TRIG IN terminal on the rear panel), the PWR-01 change is executed. If the trigger setting is already reserved in PWR-01 with a command (VOLT:TRIG or CURR:TRIG), changes can be executed with hardware triggers without a PC.

For details on the connection, see the user’s manual.

Sequence (PROGram)

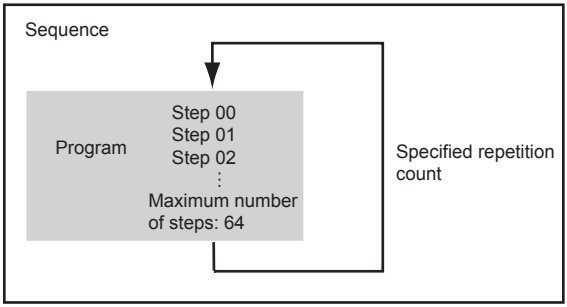
PWR-01 has two different trigger subsystems -TRANsient, and PROGram.

The PROGram group is used to execute sequences on the PWR-01.

A sequence is function that automatically executes operations set in advance one operation at a time.

You cannot configure sequence from the panel. When a sequence is executed, the sequence information is written in the PWR-01. Such sequences can be executed from the panel without a PC.

A program executes the configured step numbers one by one in ascending order from 0. If an interval loop is specified, the steps in the middle are executed repeatedly. When the last step is complete, the program is complete. When the program is executed for the specified repetition count, the sequence is complete.



■ Setting program conditions

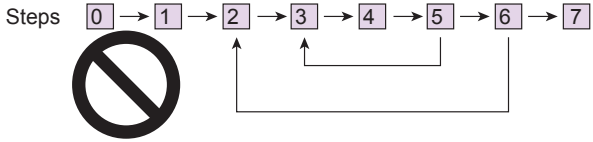
The parameters you set in a program are as follows:

Item	Description	Command	Default
Repetition count	Number of times that the program will repeat	PROG:LOOP	1
Interval loop	Interval loop to execute in the program	PROG:STEPS:LOOP:ADD	--
User code	User code that you can view on the panel	PROG:UCOD	--

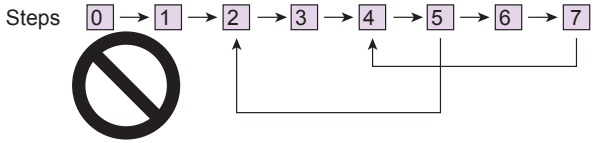
Interval loop

The steps in a sequence are normally executed in order from step 0. However, you can set interval loops to repeat intermediate steps.
Interval loops can be set in up to 16 locations in a single program.
Interval loop cannot be nested. An interval loop cannot overlap with another interval loop.

Nesting

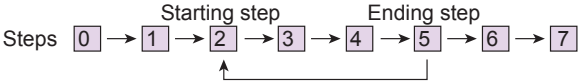


Overlap



You cannot delete or edit interval loops that have already been set. To do so, you need to create the sequence from scratch.

- Example of setting an interval loop
  The order in which the steps are executed when the starting step is set to 2, the ending step to 5, and the number of interval loops to 2 (PROG:STEPS:LOOP:ADD 2,5,2)
  0→1→2→3→4→5→2→3→4→5→6→7…



Starting from step 0, steps 2 to 5 are repeated twice, and step 6 and subsequent steps are executed.

User code

You can execute sequences that you configured from the panel. You cannot view the content of sequences from the panel, but you can view the user code.
To prevent unintentional sequence execution, we recommend that you set unique user codes for each sequence and check the user code before executing the sequence.

■ Setting step conditions

The parameters you set in a step are as follows: Use <n> of commands to specify the step number.

You cannot set the output to on or off in a step. If you want to turn off the output, set the voltage to 0 V and the current to 0 A.

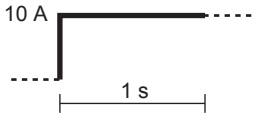
When all the processes of a program are complete, the output is in the last-step state.

Parameter	Description	Command	Default
Step time	Time to execute the step	PROG:STEP<n>:DWEL	1 s
Voltage	—	PROG:STEP<n>:VOLT	0 V
Voltage transition	Voltage step transition		IMMediate
Current	—	PROG:STEP<n>:CURR	MAXimum
Current transition	Current step transition		IMMediate
Trigger output	Trigger output on/off	PROG:STEP<n>:TRIGOUT	Off
Trigger input	Pause until a trigger signal is received	PROG:STEP<n>:TRIGIN	Off

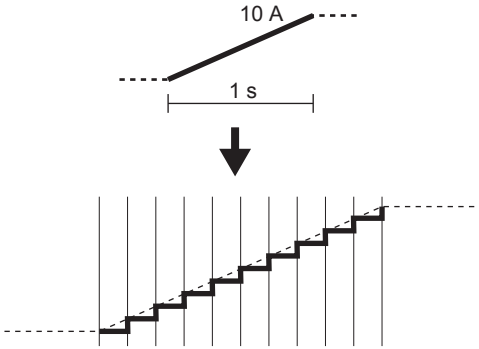
Transition

Set the step transition. If you select IMMediate, the step changes in a step. If you select RAMP, the step changes with a slope (actually, step transition at 100 ms intervals) over the step time.

Example: Current value: 10 A
Immediate transition (IMMediate)
Step time: 1 s

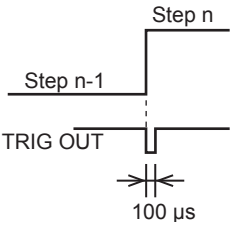


Example: Current value: 10 A
Ramp transition (RAMP)
Step time: 1 s



Trigger output

Set the trigger output to on or off. When set to on, a trigger signal (100 µs) is output from pin 7 of the TRIG OUT terminal on the rear panel simultaneously with execution of the step.

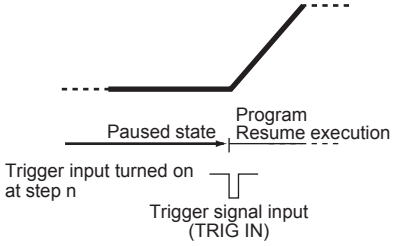


Example: Setting the trigger output at step n (TRIG ON)

Trigger input

Set the step pause. When set to on, the sequence operation pauses before executing the step. To release the pause state, enter a trigger signal (pin 7 of the TRIG IN terminal on the rear panel).

Example: Pausing a sequence by turning on the trigger input and releasing by applying a trigger input signal (TRIG IN)



■ Program template

If you set a program template, new programs are configured with steps set to the values from the template. This is effective when you want to set common step content.

The settings of a program template cannot be saved.

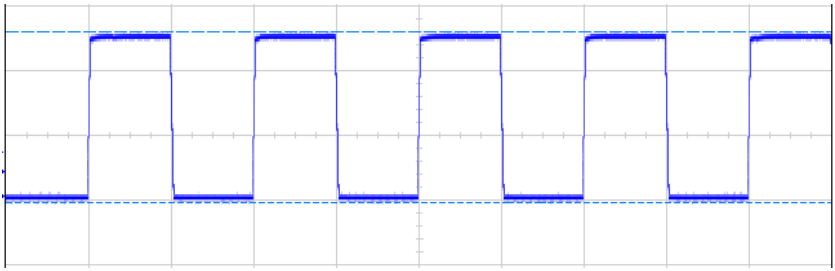
The settings do not change even when you send a *RST or *RCL command. The contents returned to their default values when the PWR-01 is restarted.

Before creating a program, set the settings that are common to multiple steps in the template.

Parameter	Description	Command	Default
Step time	Time to execute the step	PROG:STEP_T:DWEL	1 s
Voltage	—	PROG:STEP_T:VOLT	0 V
Voltage transition	Voltage step transition		IMMediate
Current	—	PROG:STEP_T:CURR	MAXimum
Current transition	Current step transition		IMMediate
Trigger output	Trigger output on/off	PROG:STEP_T:TRIGOUT	Off
Trigger input	Pause until a trigger signal is received	PROG:STEP_T:TRIGIN	Off

■ Sequence example1 (square wave)

This section describes a sequence that continuously outputs a rectangular wave.



Parameter	Value
Repetition count	Infinity
Interval loop	--
User code	1234

Parameter	Steps	
	0	1
Step time	0.5 s	0.5 s
Voltage	5.0 V	0.0 V
Voltage transition	IMM	IMM
Current	1.0 A	1.0 A
Current transition	IMM	IMM
Trigger output	Off	Off
Trigger input	Off	Off

First, reset.

```
*RST
```

Delete the current sequence, and create a new 2-step program.

```
PROG:CRE 2
```

You cannot change the number of steps later. If you want to add or delete them, use the PROG:CRE command to set the number again.

The steps will contain default values (Step time: 1 s, Voltage: 0 V, Voltage transition: IMM, Current: Max., Current transition: IMM, Trigger output: Off, Trigger input: Off).

For step 0, set the step time to 0.5 s, the voltage to 5.0 V, and the current to 1 A.

```
PROG:STEP0:DWELL 0.5
PROG:STEP0:VOLT 5
PROG:STEP0:CURR 1
```

For step 1, set the step time to 0.5 s and the current to 1 A.

```
PROG:STEP1:DWELL 0.5
PROG:STEP1:CURR 1
```

Set the user code to 1234. You can also view the user code on the panel.

```
PROG:UCOD 1234
```

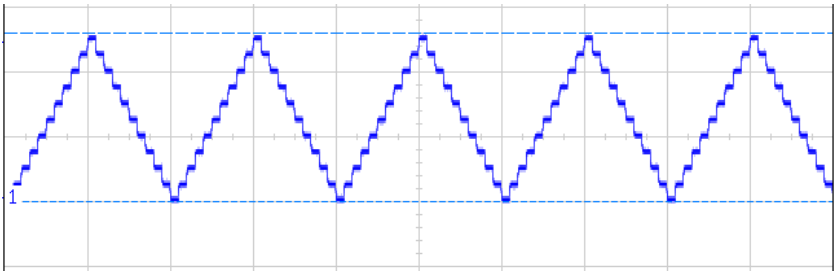
Set the repetition count to infinity.

```
PROG:LOOP INF
```

After you have finished configuring the sequence, execute it.

■ Sequence example2 (triangular wave)

This section describes a sequence that continuously outputs a triangular wave.



In a step with voltage transition set to RAMP, the voltage changes linearly (actually, step transition at 100 ms intervals) from the voltage setting of the previous step to the specified voltage.

Parameter	Value
Repetition count	Infinity
Interval loop	--
User code	5678

Parameter	Steps	
	0	1
Step time	1 s	1 s
Voltage	5.0 V	0.0 V
Voltage transition	RAMP	RAMP
Current	1.0 A	1.0 A
Current transition	IMM	IMM
Trigger output	Off	Off
Trigger input	Off	Off

First, reset.

```
*RST
```

Delete the current sequence, and create a new 2-step program.

```
PROG:CRE 2
```

You cannot change the number of steps later. If you want to add or delete them, use the PROG:CRE command to set the number again.

The steps will contain default values (Step time: 1 s, Voltage: 0 V, Voltage transition: IMM, Current: Max., Current transition: IMM, Trigger output: Off, Trigger input: Off).

For step 0, set the voltage to 5.0 V, the voltage transition to RAMP, and the current to 1 A.

```
PROG:STEP0:VOLT 5,RAMP
PROG:STEP0:CURR 1
```

For step 1, set the voltage transition to RAMP and the current to 1 A.

```
PROG:STEP1:VOLT 0,RAMP
PROG:STEP1:CURR 1
```

Set the user code to 5678. You can also view the user code on the panel.

```
PROG:UCOD 5678
```

Set the repetition count to infinity.

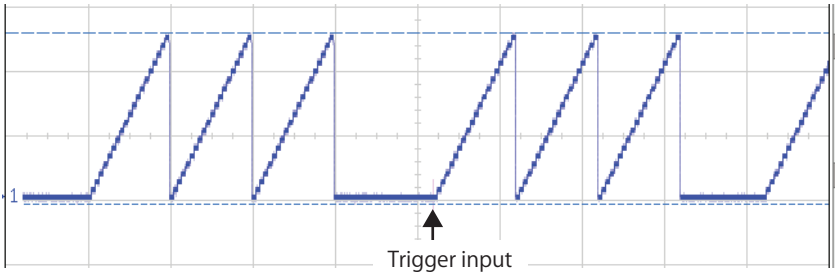
```
PROG:LOOP INF
```

After you have finished configuring the sequence, execute it.

■ Sequence example3 (sawtooth wave)

This section describes a sequence that repeats 10 000 times a pattern that outputs three sawtooth waves and then outputs another three sawtooth waves on an external trigger signal (CH2).

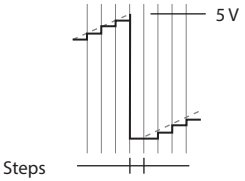
Create steps using a template, and then edit each step.



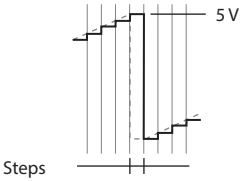
In a step with voltage transition set to RAMP, the voltage changes linearly (actually, step transition at 100 ms intervals) from the voltage setting of the previous step to the specified voltage.

Because we want to change the voltage from 0.0 V to 5.0 V, the step before the step that will be set to 5.0 V must be set to 0.0 V. For this purpose, we will set short starting voltage steps before steps that will be set to 5.0 V. In this example, steps 1, 3, and 5 are the starting voltage steps. The minimum step execution time that you can set is 0.1 s. Do not set the step execution time to 0.0 s.

Set the voltage transition to RAMP for the starting voltage steps. The sawtooth wave will not be clean if you specify IMM.



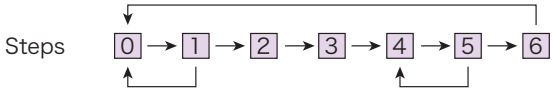
If the transition for the starting voltage steps is set to IMM



If the transition for the starting voltage steps is set to RAMP

Parameter	Value
Repetition count	10000
Interval loop	Starting step: 0, ending step: 1, loop count: 3 Starting step: 4, ending step: 5, loop count: 2
User code	7890

Parameter	Steps						
	0	1	2	3	4	5	6
Step time	1.9 s	0.1 s	1.9 s	0.1 s	1.9 s	0.1 s	2.0 s
Voltage	5.0 V	0.0 V	5.0 V	0.0 V	5.0 V	0.0 V	0.0 V
Voltage transition	RAMP	RAMP	RAMP	RAMP	RAMP	RAMP	IMM
Current	1.0 A	1.0 A	1.0 A	1.0 A	1.0 A	1.0 A	1.0 A
Current transition	IMM	IMM	IMM	IMM	IMM	IMM	IMM
Trigger output	Off	Off	Off	Off	Off	Off	Off
Trigger input	Off	Off	On	Off	Off	Off	Off



First, reset.

```
*RST
```

Next, set the program template values.

Set the current.

Use default values for all other items (Step time: 1 s, Voltage: 0 V, Voltage transition: IMM, Current transition: IMM, Trigger output: Off, Trigger input: Off).

```
PROG:STEP_T:CURR 1
```

Delete the current sequence, and create a new 7-step program using the set template.

```
PROG:CRE 7,TEMP
```

You cannot change the number of steps later. If you want to add or delete them, use the PROG:CRE command to set the number again.

A 7-step program with the current set to 1 A and the remaining parameters set to default values is created.

For step 0, set the step time to 1.9 s, the voltage to 5.0 V, and the voltage transition to RAMP.

```
PROG:STEP0:DWELL 1.9
PROG:STEP0:VOLT 5,RAMP
```

For step 1, set the step time to 0.1 s and the voltage transition to RAMP.

```
PROG:STEP1:DWELL 0.1
PROG:STEP1:VOLT 0,RAMP
```

For step 2, set the step time to 1.9 s and the voltage to 5 V. Because we want to start this step with an external trigger, set the trigger input to on.

```
PROG:STEP2:DWELL 1.9
PROG:STEP2:VOLT 5,RAMP
PROG:STEP2:TRIGIN ON
```

For step 3, set the step time to 0.1 s and the voltage transition to RAMP.

```
PROG:STEP3:DWELL 0.1
PROG:STEP3:VOLT 0,RAMP
```

For step 4, set the step time to 1.9 s, the voltage to 5 V, and the voltage transition to RAMP.

```
PROG:STEP4:DWELL 1.9
PROG:STEP4:VOLT 5,RAMP
```

For step 5, set the step time to 0.1 s and the voltage transition to RAMP.

```
PROG:STEP5:DWELL 0.1
PROG:STEP5:VOLT 0,RAMP
```

For step 6, set the step time to 2.0 s and the voltage transition to RAMP.

```
PROG:STEP6:DWELL 2
PROG:STEP6:VOLT 0,RAMP
```

Repeat steps 0 to 1 three times and steps 4 to 5 two times.

```
PROG:STEPS:LOOP:ADD 0,1,3
PROG:STEPS:LOOP:ADD 4,5,2
```

Interval loops cannot be changed or deleted. If you want to change or delete them, use the PROG:CRE command to set the number again.

Set the user code to 7890. You can also view the user code on the panel.

```
PROG:UCOD 7890
```

Set the repetition count to 10000.

```
PROG:LOOP 10000
```

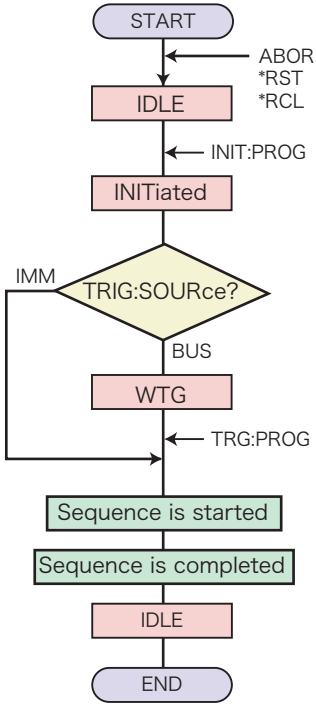
After you have finished configuring the sequence, execute it.

■ Execution using triggers

First, set the output to on.

```
OUTP ON
```

A sequence has three states: IDLE, INITiated, and WTG.



To start a sequence immediately, set the trigger source to IMMediate, and then use the INITiate:PROGram command.

```
TRIG:PROG:SOUR IMM   'Sets the trigger source to IMM
INIT:PROG   'Initiates PROGram subsystem. The sequence begins.
```

When multichannel is use, you can start the sequence operation on all channels by using the GLOB:INIT:PROG command.

To use software triggers to start a sequence, change the trigger source to BUS.

If you are using software triggers to start the sequence, change the trigger source to BUS.

```
TRIG:PROG:SOUR BUS   'Sets the trigger source to BUS
INIT:PROG   'Initiates PROGram subsystem.
TRIG:PROG   'Applies a software trigger to PROGram subsystem. The sequence
begins.
```

If you use INITiate:PROGram to bring the TRIGger subsystem out of the IDLE state and start (initiate) the trigger function, the TRIGger subsystem enters the WTG (Waiting For Trigger) state. When a software trigger is received (through the TRIGger:PROGram command or *TRG command), the sequence starts.

TRIGger:PROGram only applies a software trigger to the PROGram subsystem.

The *TRG command or the IEEE488.1 get (Group Execute Trigger) command applies a software trigger to all trigger subsystems, if there are other trigger subsystems in the initiated state, their trigger functions will also be executed at the same time.

When multichannel is use, you can send software triggers to all channels by using the GLOB:*TRIG command.

When the sequence finishes, the PROGram subsystem enters the IDLE state again. If the ABORt command or an equivalent command is received in the WTG state or when a sequence is being executed, the sequence is canceled, and the PROGram subsystem returns to the IDLE state.

You can view the remaining repetition count while a sequence is running.

```
PROG:REM:LOOP?
```

You can view the remaining execution time of a sequence.

```
PROG:REM:TIME?
```

You can collectively view the execution state, present repetition count, present step number, elapsed time, and estimated time of a sequence.

```
TRIG:PROG:EXEC
```

If you want to terminate a sequence, use the ABORt:PROGram command.

```
ABOR:PROG
```

When all the processes of a program are complete, the output state remains in the last-step state. In the sequence example, the sequence finishes with the 9 V output turned on.

In the end, turn the output off.

```
OUTP OFF
```

When the power is turned off, the template information is deleted.

When a sequence is executed, the sequence information is written in the PWR-01. Such sequences can be executed from the panel without a PC. For details how to execute sequences from the panel, see the user’s manual.

PWR-01 behavior during synchronized operation

Turn on the trigger input in a step of your choice on the PWR-01 operating in sync. When all PWR-01s receive a hardware trigger (pin 7 of the TRIG IN terminal on the rear panel) when the sequence is paused, the pause is released simultaneously on all the PWR-01s. If the sequence information is written in the PWR-01, hardware triggers can be used to release the pause state without a PC.

If PWR-01 are cascaded, turn on the trigger input and trigger output.

For details on the connection, see the user's manual.

Status Monitoring

The PWR-01 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 standard event/filter architecture, employing CONDition, EVENt, ENABle, and optionally PTRansition and NTRansition. CONDition and EVENt are read-only registers working as status indicators, and ENABle, PTRansition and NTRansition are read-write registers working as event and summary filters.

■ STATus:OPERation

The STATus:OPERation register records events or signals that occur during normal operation.

For example, to check if the PWR-01 is being regulated in CV state, check the CV bit (bit 8) on the STATus:OPERation register.

```
STATus:OPERation?  'Check whether the CV bit is set
```

When multichannel is in use, check the CV bit (bit 8) of the STATus:OPERation:INSTrument:ISUMmary<n> subregister.

```
STATus:OPERation:INSTrument:ISUMmary2?  'Check whether the CV bit of channel 2 is set.
```

■ STATus:QUEStionable

The STATus:QUEStionable register records events or signals that indicate abnormal operation.

To check if the protection function is working, check the OV bit (bit 0) on the STATus: QUEStionable register.

```
STATus:QUEStionable?  'Check whether the OV bit is set
```

When multichannel is in use, even if bit 0 is true, you cannot tell on which channel the overvoltage protection function has been activated. To check which channel is operating abnormally, check the STATus:QUEStionable:INSTrument subregister.

```
STATus:QUEStionable:INSTrument?  'Check which channel is operating abnormally.
```

All channels whose corresponding bits are true are operating abnormally. You can determine how the specified channel is operating abnormally by checking the STATus:QUEStionable:INStrument:ISUMmary<n> subregister of the channel.

```
STATus:QUEStionable:INSTrument:ISUMmary2?  'Check whether the OV bit of channel 2 is set.
```

■ PON (Power ON) bit

The PON bit (bit 7) in the event status register is set whenever the PWR-01 is turned on. The most common use for the PON bit is to generate an SRQ at power-on to keep track of unexpected loss of power or power line failure. To do this, follow the steps shown below.

1 **Set *PSC (Power-on Status Clear) to 0 (or OFF).**
Enable the backup function of the event status enable register and service request enable register (*PSC 0).

2 **Set the PON bit (bit 7) of the event status enable register.**
Permit the transmission of a power-on event to the upper layer (*ESE 128).

3 **Set the ESB bit (bit 5) of the status byte enable register.**
Permit the generation of an SRQ caused by a standard event (*SRE 32).

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

When using the RS232C interface, the PON bit cannot be assigned to the service request, because SRQs are not generated.

Though the SRQ feature itself is provided by communication protocol on the USB interface or LAN (VXI-11/ HiSLIP) interface, a Connection Lost error in the VISA I/O session occurs immediately before the power-on event. It may be difficult to handle PON events when using the USB interface.

Multichannel

When you are using the multichannel function, the PWR-01 that communicates with the PC is the VMCB master unit.

To configure a PWR-01 that you are using with the multichannel function, you first have to specify the channel of the PWR-01 that you want to configure.

At power on, the PWR-01 is set to channel 0.

First specify the channel, and then configure the other settings.

Setup example for a PWR-01 that you set to channel 2:

```
INST 2   'Specify channel 2.
VOLT 12.0   'Set the channel 2 voltage to 12.0 V.
CURR 1.5   'Set the channel 2 current to 1.5 A.
```

You can use the GLOBal subsystem to make changes to the settings (voltage, current, turning output on or off) of all channels at the same time. Insert a wait of at least 200 ms after sending a GLOBal subsystem command.

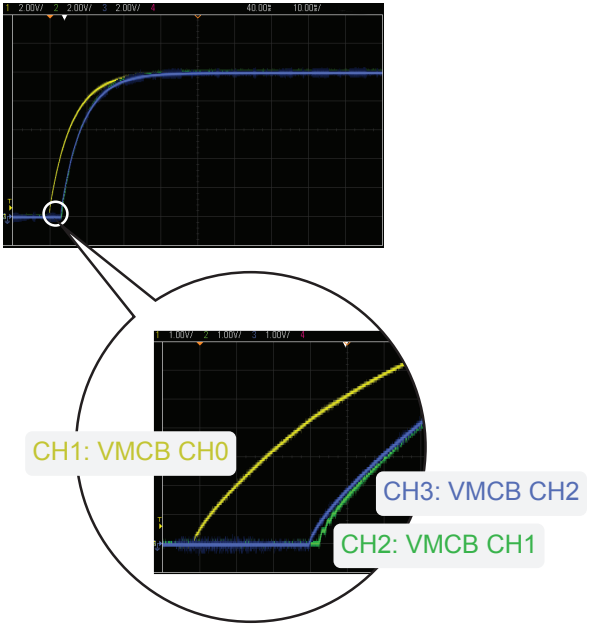
```
GLOB:VOLT 20   'Set the voltage of all channels to 20 V.
GLOB:CURR 2   'Set the current of all channels to 2 A.
GLOB:OUTP 1   'Turn output on for all channels.
```

The GLOB:OUTP command turns on/off the output of all configured channels, but they are not turned on/off at the same time. There is a slight time offset between channels.

- Note -
The time offset varies depending on the number of PWR-01 units in the multi-channel configuration and the network conditions.

As a reference, the waveforms are shown for the following measurement environment.

Measurement environment
Software: KI-VISA 5.5.0
PC: Intel(R) Core(TM) i7-47901 3.60GHz / Windows7 Professional(x64)
Interface: LAN
PWR401L: 3 units



If you specify the channel after the MEAS/FETC node, you can query measured values without using the INST command to specify the channel.

```
MEASure2:VOLTage?   'Query the channnel 2 voltage output
MEASure2:CURRent?   'Query the channnel 2 current output
```

If you do not specify the channel after the MEAS/FETC node, the measured value of the channel that was specified by the INST command is returned.

```
INST 3   'Specify channel 3.
MEASure:VOLTage?   'Query the channnel 3 voltage output
MEASure:CURRent?   'Query the channnel 3 current output
```

You can configure changes to settings on the PWR-01s in advance, and then use triggers to synchronize the changes on all units. The INIT and TRIG commands are sent to all channels.

Example in which channels 0 and 1 are changed simultaneously:

```
INST:NSEL 0   'Specify channel 0.
TRIG:SOUR BUS   'Set the trigger source of channel 0 to BUS.
VOLT:TRIG 10;:CURR:TRIG 10   'Configure channel 0 so that the voltage will be set to 10 V and the current to 10 A when a trigger occurs.
INST:NSEL 1   'Specify channel 1.
TRIG:SOUR BUS   'Set the trigger source of channel 1 to BUS.
VOLT:TRIG 20;:CURR:TRIG 20   'Configure channel 1 so that the voltage will
```

be set to 20 V and the current to 20 A when a trigger occurs.

```
GLOB:INIT:TRAN  'Initiate the measurement.
GLOB:*TRIG  'Apply a software trigger.
```

When you send the TRIG command, the settings of channel 0 and channel 1 are changed.

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 PWR-01 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 is empty when the *CLS common command is sent, when the last item in the queue is read, and when the PWR-01 is turned on. When the error/event queue is empty, the query returns the following:

0,"No error"

■ Displaying communication errors

The PWR-01 has a debug trace function. The oldest item among the errors and events (if they are present) can be displayed on the PWR-01. This function is convenient when you debug your remote applications.

While an error/event item is displayed on the panel, the normal voltmeter and ammeter are disabled.

If the error/event queue is empty, the debug trace function does not display any errors. Sending the *CLS command clears the communication error display.

If in local mode, the debug trace function is temporarily disabled.

The communication error display can be enabled or disabled with the SYSTem:ERRor:TRACe command.

```
SYSTem:ERRor:TRACe {ON|OFF}
```

■ Setting the "Project"

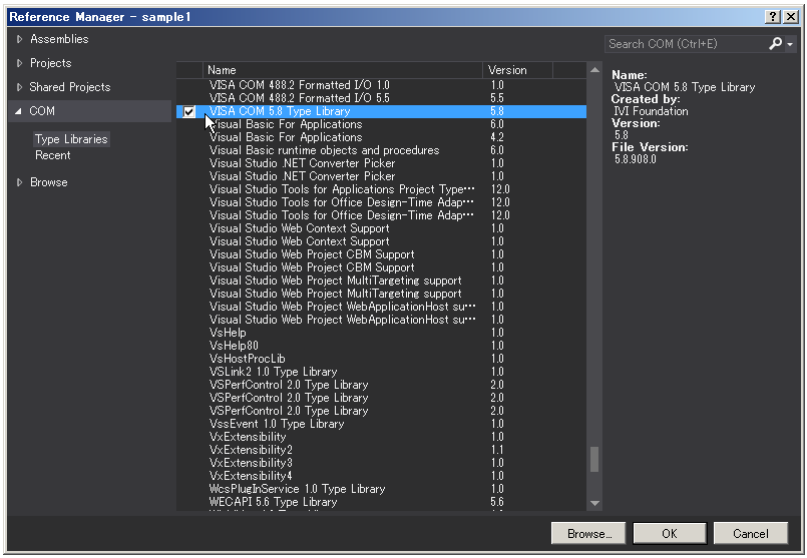
First, add the communication middleware (VISA library) to the project.

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.



■ Communication through RS232C, USB, or LAN

Open the VISA

To communicate with an RS232C, USB, or LAN device using VISA, you have to first open VISA. When you open VISA, specify the I/O resource.

Example : To open VISA by using USB

```
Set rm = CreateObject("VISA.GlobalRM")
Set msg = rm.Open("USB::0x0B3E::0x1049::00000001::INSTR", NO_LOCK, 0, "")
```

"USB::0x0B3E::0x1049::00000001::INSTR" is the I/O resource.

The I/O resource is specified by the following constructions. The part indicated with [ ] can be omitted. Enter the appropriate values in the parts specified in oblique characters.

Serial (RS232C)		ASRL[*board*][::INSTR] Example : The measuring instrument connected to the serial port COM1. ASRL1::INSTR
USB		USB[*board*]::*VendorID*::*ProductID*::*SerialNumber*[::*InterfaceNumber*][::INSTR] Example: The USNTMC measuring instrument having vendor ID (VID) 2878, Product ID (PID) 4169 and serial number "00000001." USB0::0x0B3E::0x1049::00000001::INSTR
LAN[*1]	VXI-11	TCPIP[*board*]::*hostname*[::inst*0*][::INSTR] Example :The measuring instrument whose IP address (hostname) is 169.254.7.8. TCPIP::169.254.7.8::INSTR You can also set the LAN device name using the host name.
	HiSLIP	TCPIP[*board*]::*hostname*::hislip0[::INSTR] Example :The measuring instrument whose IP address (hostname) is 169.254.7.8. TCPIP::169.254.7.8::hislip0::INSTR You can also set the LAN device name using the host name.
	SCPI-RAW	TCPIP[*board*]::*hostname*::*portno*::SOCKET Example :The measuring instrument whose IP address (hostname) is 169.254.7.8. (The "portno" setting of the PWR-01 is always 5025.) TCPIP::169.254.7.8::5025::SOCKET You can also set the LAN device name using the host name.

[*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.

For VISA, the alias can be used for the I/O resource.

When using the alias for the I/O resource, even if the alias name is hard-coded directly in the application, the alias name can be easily converted to the appropriate I/O resource name.

Example : When using the alias (MYDEV1) for the I/O resource.

```
Set msg = rm.Open("MYDEV1", NO_LOCK, 0, "")
```

When the alias is used, the actual I/O resource is specified by an external configuration table.

Please refer to the applicable VISA manual.

Controlling the devices

Next, use "Read" and "Write" commands to control devices. You must include line-feed codes in the command strings.

Example:

```
msg.WriteString ("VOLT 10" & vbLF)      'Set the voltage to 10 V
msg.WriteString ("OUTP 1" & vbLF)       'Turn the output on
```

Closing the VISA

Close the VISA at the end.

You only need to include one "open" VISA command and one "close" VISA command in the program.

```
msg.Close
```

■ Sample program

```
Imports Ivi.Visa.Interop


Public Class Form1


    Dim rm As ResourceManager
    Dim msg As IMessage


    Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        rm = CreateObject("VISA.GlobalRM")
        msg = rm.Open("USB0::0x0B3E::0x1049::00000001::INSTR", AccessMode.NO_LOCK, 0, "")


        'Example: Using an VISA alias
        'msg = rm.Open("MYDEV1", AccessMode.NO_LOCK, 0, "")
        'Example: LAN(SCPI-RAW)
        'msg = rm.Open("TCPIP::169.254.7.8::5025::SOCKET", AccessMode.NO_LOCK, 0, "")
```

```
        msg.TerminationCharacterEnabled = True
    End Sub

    'Query the instrument identity
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        msg.WriteString("SYST:COMM:RLST?" & vbLf)
        msg.WriteString("*IDN?" & vbLf)
        TextBox1.Text = msg.ReadString(256)
    End Sub

    'Set the voltage and current
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        msg.WriteString("OUTP 0" & vbLf)
        msg.WriteString("VOLT 8" & vbLf)
        msg.WriteString("CURR 5" & vbLf)
        msg.WriteString("OUTP 1" & vbLf)
    End Sub

    'Querys measurement voltage
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        msg.WriteString("MEAS:VOLT?" & vbLf)
        TextBox1.Text = msg.ReadString(256)
    End Sub

    Private Sub Form1_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Disposed
        msg.Close()
    End Sub

End Class
```

A 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 shall cause the Command Error (bit 5) in the event status register to be set.

Error code		Error message description
-100	Command error	This is the generic syntax error.
-101	Invalid character	A syntactic element contains a character that is invalid for that type.
-102	Syntax error	An unrecognized command or data type was encountered.
-103	Invalid separator	The parser was expecting a separator and encountered an illegal character.
-104	Data type error	The parser recognized a data element different than one allowed.
-105	GET not allowed	A Group Execute Trigger was received within a program message.
-108	Parameter not allowed	More parameters were received than expected for the header.
-109	Missing parameter	Fewer parameters were recieved than required for the header.
-110	Command header error	An error was detected in the header.
-112	Program mnemonic too long	The header contains more that twelve characters.
-113	Undefined header	The header is undefined for this device.
-114	Header suffix out of range	The value of a numeric suffix attached to a program mnemonic.
-115	Unexpected number of parameters	The number of parameters received does not correspond to the number of parameters expected.
-120	Numeric data error	This error is generated when parsing a data element that appears to be numeric, including the nondecimal numeric types.
-128	Numeric data not allowed	A legal numeric data element was received, but the device does not accept one in this position for the header.
-130	Suffix error	This error is generated when parsing a suffix.
-131	Invalid suffix	The suffix does not follow the syntax or the suffix is inappropriate for this device.
-134	Suffix too long	The suffix contained more than 12 characters.
-138	Suffix not allowed	A suffix was encountered after a numeric element which does not allow suffixes.

Error code		Error message description
-140	Character data error	This error is generated when parsing a character data element.
-141	Invalid character data	Either the character data element contains an invalid character or the particular element received is not valid for the header.
-144	Character data too Long	The character data element contains more than twelve characters.
-148	Character data not allowed	A legal character data element was encountered where prohibited by the device.
-150	String data error	This error is generated when parsing a string data element.
-151	Invalid string data	A string data element was expected, but was invalid for some reason.
-158	String data not allowed	A string data element was encountered but was not allowed by the device at this point in parsing.
-160	Block data error	This error is generated when parsing a block data element.
-170	Expression error	This error is generated when parsing an expression data element.
-180	Macro error	This error is generated when defining a macro or executing a macro.

■ 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 shall cause the Execution Error (bit 4) in the event status register to be set.

Error code		Error message description
-200	Execution error (generic)	This is the generic syntax error for devices that cannot detect more specific errors.
-203	Command protected	Indicates that a legal password-protected program command or query could not be executed because the command was disabled.
-210	Trigger error	Trigger error.
-211	Trigger ignored	Indicates that a GET, *TRG, or triggering signal was received and recognized by the device but was ignored because of device timing considerations.
-213	Init ignored	Indicates that a request for a measurement initiation was ignored as another measurement was already in progress.
-214	Trigger deadlock	Indicates that the trigger source for the initiation of a measurement is set to GET and a subsequent measurement query was received.
-220	Parameter error	Indicates that a program data element related error occurred.

Error code		Error message description
-221	Settings conflict	Indicates that a legal program data element was parsed but could not be executed due to the current device state.
-222	Data out of range	Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device.
-223	Too much data	Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.
-224	Illegal parameter value	Used where an exact value, from a list of possible values, was expected.
-230	Data corrupt or stale	Possibly invalid data; new reading started but not completed since last access.
-241	Hardware missing	Indicates that a legal program command or query could not be executed because of missing device hardware.

■ Device-specific errors

An error in the range [-399, -300] indicates that the instrument has detected an error which is not a command error, a query error, or an execution error. The occurrence of any error in this class should cause the device-specific error bit (bit 3) in the event status register to be set.

Error code		Error message description
-310	System error	Indicates that some error, termed “system error” by the device, has occurred.
-311	Memory error	Indicates some physical fault in the device’s memory, such as parity error.
-313	Calibration memory lost	Indicates that nonvolatile calibration data used by the *CAL? command has been lost.
-314	Save/recall memory lost	Indicates that the nonvolatile data saved by the *SAV? command has been lost.
-315	Configuration memory lost	Indicates that nonvolatile configuration data saved by the device has been lost.
-330	Self-test failed	Self-test failed
-350	Queue overflow	A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.
-360	Communication error	Communication error when the flow control is turned off. This error applies when the RS232C interface is used.
-362	Framing error in program message	Framing error. This error applies when the RS232C interface is used.

Error code		Error message description
-363	Input buffer overrun	Buffer overrun error. This error applies when the RS232C interface is used.
-365	Time out error	Time out error. This error applies when the RS232C interface is used.

■ 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 shall cause the Query Error (bit 2) in the event status register to be set.

Error code		Error message description
-400	Query error (generic)	This is the generic query error for devices that cannot detect more specific errors.
-410	Query INTERRUPTED	Received a new command before the response was read.
-420	Query UNTERMINATED	The controller attempted to read the response after the device received an unsupported query or has not received a query. The -100 "COMMAND ERROR" 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.
-440	Query UNTERMINATED after indefinite response	Received a separate query in semicolon-delimited format after a query that returns a response in an indefinite form. (Example: A command such as the following. *IDN?;SYST:ERR?)

■ Operation Complete Event

An error in the range [-899: -800] is used when the instrument wishes to report a 488.2 operation complete event to the error/event queue. This event occurs when instrument's synchronization protocol, having been enabled by an *OPC command, completes all selected pending operations. This protocol is described in IEEE 488.2, section 12.5.2. This event also sets the operation complete bit (bit 0) of the Standard Event Status Register.

Error code		Error message description
-800	Operation complete	The instrument has completed all selected pending operations in accordance with the IEEE 488.2, 12.5.2 synchronization protocol.

■ Device-dependent errors

An error in the range [+1, +32767] indicates that the instrument has detected an error which is not a command error, a query error, or an execution error. The occurrence of any error in this class should cause the device-specific 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
+103	Conflicts with SLAVe operation	Configuration is not possible because the unit is running in slave mode.
+141	CURR setting conflicts with CURR:PROT setting	Configuration is not possible because CURR is disabled by the CURR:PROT setting.
+142	CURR:PROT setting conflicts with CURR setting	Configuration is not possible because CURR:PROT is disabled by the CURR setting.
+151	VOLT setting conflicts with VOLT:PROT setting	Configuration is not possible because VOLT is disabled by the VOLT:PROT setting.
+152	VOLT:PROT setting conflicts with VOLT setting	Configuration is not possible because VOLT:PROT is disabled by the VOLT setting.
+153	VOLT setting conflicts with VOLT:LIM LOW setting	Configuration is not possible because VOLT is disabled by the VOLT:LIM:LOW setting.
+154	VOlt:LIM:LOW setting conflicts with VOLT setting	Configuration is not possible because VOLT:LIM:LOW is disabled by the VOLT setting.
+155	Conflicts with PROTection state	Configuration is not possible because a protection function is activated.
+170	MEMory contents coflict with CURR:PROT setting	Memory content cannot be recalled because of the CURR:PROT setting.
+171	MEMory contents coflict with VOLT:PROT setting	Memory content cannot be recalled because of the VOLT:PROT setting.
+172	MEMory contents coflict with VOLT:LIM:LOW setting	Memory content cannot be recalled because of the VOLT:LIM:LOW setting.

Conflict errors during trigger function execution

These errors occur settings are in conflict with the existing settings.

Error code		Error message description
+211	Conflicts with TRANsient in progress	Configuration is not possible because a TRANsient subsystem is running.
+212	Conflicts with PROGram in progress	Configuration is not possible because a PROGram subsystem is running.
+213	Conflicts with OUTPut DELay in progress	Configuration is not possible because an output delay is in effect.
+214	Conflicts with Soft Start or Soft Stop in progress	Configuration is not possible because a soft start or soft stop is in effect.

Sequence execution errors

These errors occur when invalid or incorrect settings are specified.

Error code		Error message description
+301	Coflicts with OUPut OFF state	Configuration is not possible because the output is off.
+302	Coflicts with OUPut ON state	Configuration is not possible because the output is on.
+303	Coflicts with OUP:EXT is active	Configuration is not possible because output on/off control using an external contact is enabled.
+304	Coflicts with CURR:EXT:SOUR is active	Configuration is not possible because output current control using an external voltage or external resistance is enabled.
+305	Coflicts with VOLT:EXT:SOUR is active	Configuration is not possible because output voltage control using an external voltage or external resistance is enabled.
+306	PROG:STEP contents conflict with CURR:PROT settings	PROG:STEP cannot be set because of the CURR:PROT setting.
+307	PROG:STEP contents conflict with VOLT:PROT settings	PROG:STEP cannot be set because of the VOLT:PROT setting.
+308	PROG:STEP contents conflict with VOLT:LIM:LOW settings	PROG:STEP cannot be set because of the VOLT:LIM:LOW setting.

Errors during sequence editing

Error code		Error message description
+401	Invalid STEP index	The step index is invalid.
+402	Invalid STEP loop begin index	The start index of the step loop is invalid.
+403	Invalid STEP loop end index	The end index of the step loop is invalid.
+450	Program runtime error	Program runtime error

Errors related to the self-test function

These errors occur as results of self-tests executed with *TST? queries.

Error code		Error message description
+901	EEPROM MODEL info lost	Model information was lost.
+902	EEPROM CAL info lost	Calibration information was lost.
+903	Wrong Model ID setup	An incorrect model ID is specified.

Processing Time of Commands

The command processing time is the time until the next command is accepted.

The processing times indicated here are typical values. They are not guaranteed.

The processing times vary depending on the settings and the measurement conditions.

It does not include the response time of the hardware.

Command	Processing Time (ms)						Description
	USB	RS232C		LAN[*1]			
		Data rate setting: 19200 bps	Data rate setting: 1152000 bps	VXI-11	HiSLIP	SCPI-RAW	
VOLT CURR	0.72	4	0.7	2.4	0.7	0.02	Sets the voltage or the current
MEAS:VOLT? MEAS:CURR?	0.72	15	2.8	4.75	2.4	2.8	Queries the measured value
OUTP	0.25	4	0.7	2.4	0.7	0.02	Output on/off
*IDN	0.34	25	4.5	4.75	2.2	2	Queries the model name
*RST	190	3	0.4	2.4	0.7	0.02	Performs a device reset

[*1]: 100BASE-TX Ethernet

KIKUSUI ELECTRONICS CORP.

1-1-3 Higashiyamata, Tsuzuki-ku, Yokohama,
224-0023, Japan

Tel: +81-45-482-6353
Fax: +81-45-482-6261

www.kikusui.co.jp/en