

DC Electronic Load

PEL-3000(H)

PROGRAMMING MANUAL

VERSION: 1.11



ISO-9001 CERTIFIED MANUFACTURER

GW INSTEK

This manual contains proprietary information, which is protected by copyright. All rights are reserved. No part of this manual may be photocopied, reproduced or translated to another language without prior written consent of Good Will company.

The information in this manual was correct at the time of printing. However, Good Will continues to improve products and reserves the rights to change specification, equipment, and maintenance procedures at any time without notice.

Good Will Instrument Co., Ltd.
No. 7-1, Jhongsing Rd., Tucheng Dist., New Taipei City 236, Taiwan.

Table of Contents

| | |
|-----------------------------------|-----------|
| I INTERFACE OVERVIEW | 2 |
| Appearance | 3 |
| Interface Configuration | 9 |
| C COMMAND OVERVIEW | 18 |
| Command Syntax | 19 |
| Command List | 24 |
| Status Register Overview | 203 |
| Error Messages | 215 |

INTERFACE OVERVIEW

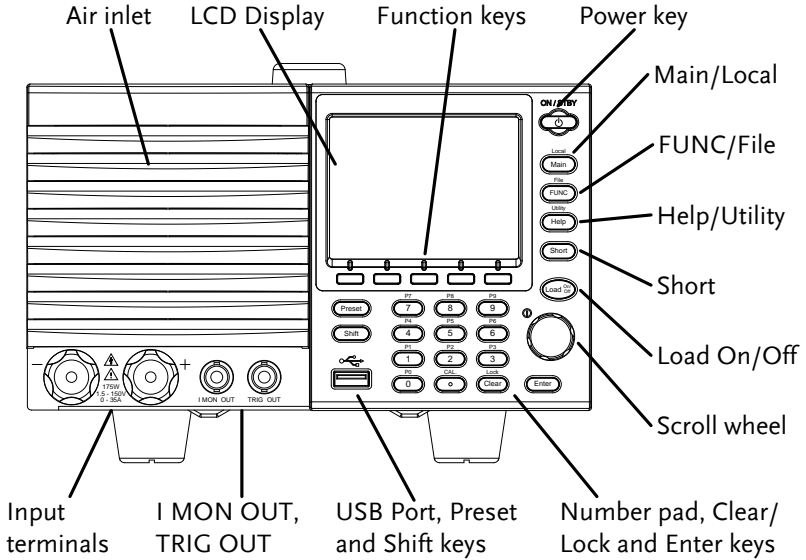
This chapter describes basic configuration of IEEE488.2 based remote control.

| | |
|---|----|
| Appearance | 3 |
| Front Panel | 3 |
| PEL-3021/ PEL-3041 | 3 |
| PEL-3021H/ PEL-3041H | 3 |
| PEL-3111 | 4 |
| PEL-3111H | 5 |
| PEL-3211(H) Booster Pack | 5 |
| Rear Panel | 6 |
| PEL-3021/ PEL-3041 | 6 |
| PEL-3021H/ PEL-3041H | 7 |
| PEL-3111 | 7 |
| PEL-3111H | 8 |
| PEL-3211 Booster Pack | 8 |
| PEL-3211H Booster Pack | 8 |
| Interface Configuration | 9 |
| Configure to USB Remote Interface | 9 |
| Configure GPIB Interface | 9 |
| Configure RS232C | 10 |
| RS232C/USB Remote Control Function Check | 11 |
| Using Realterm to Establish a Remote Connection | 12 |
| GPIB Function Check | 15 |

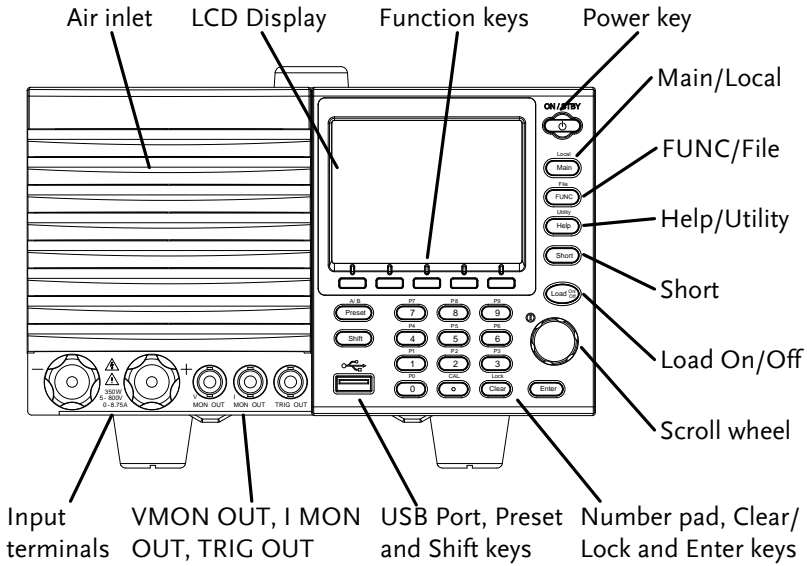
Appearance

Front Panel

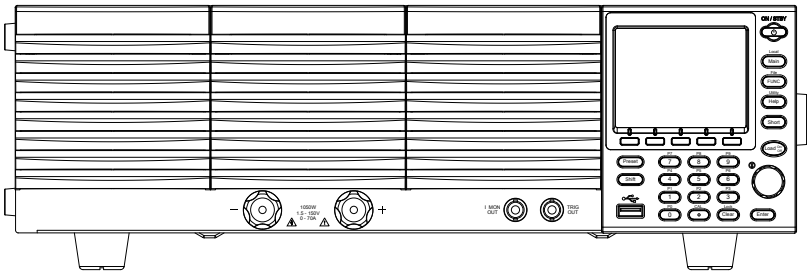
PEL-3021/ PEL-3041



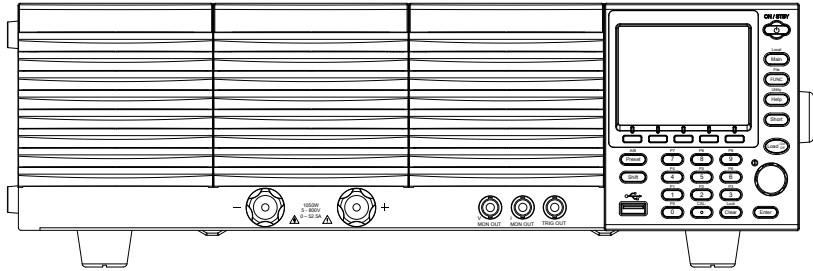
PEL-3021H/ PEL-3041H



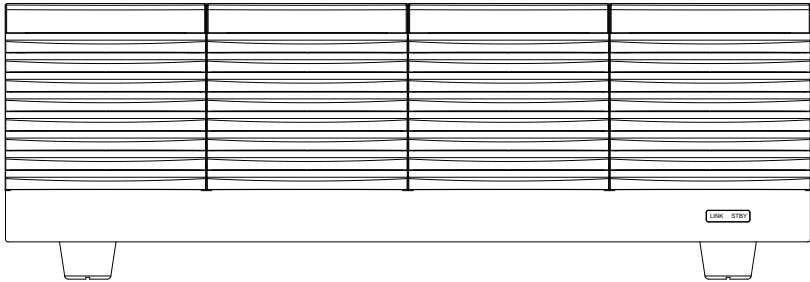
PEL-3111



PEL-3111H

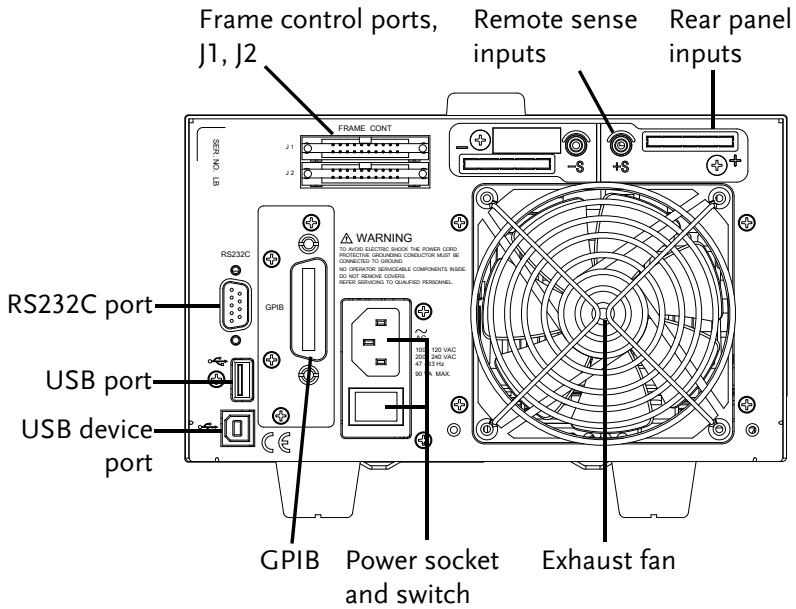


PEL-3211 (H) Booster Pack

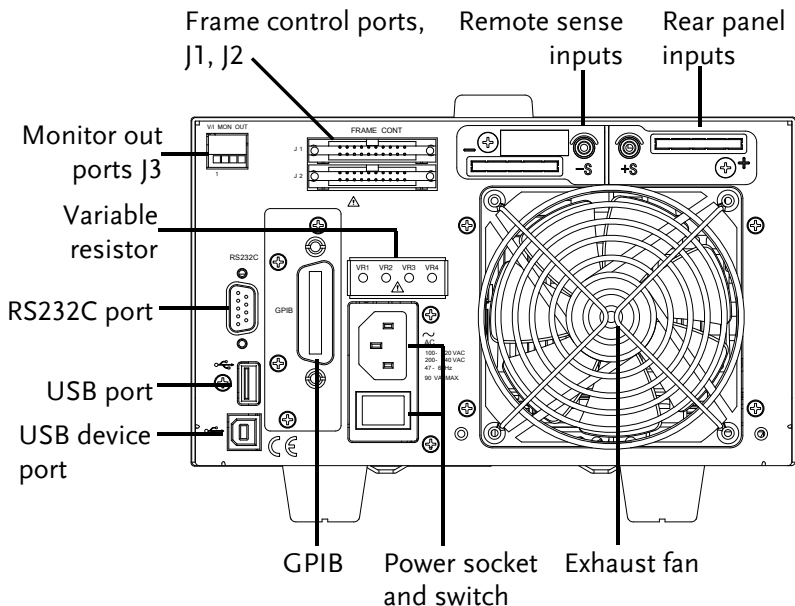


Rear Panel

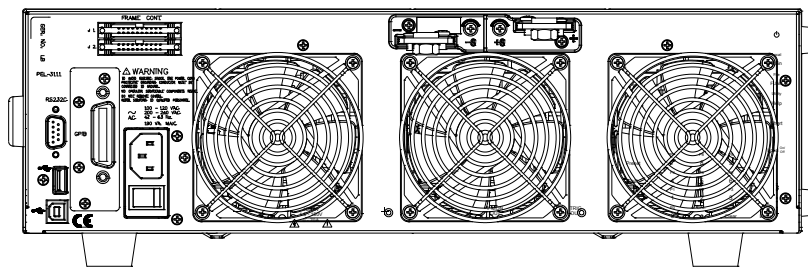
PEL-3021/ PEL-3041



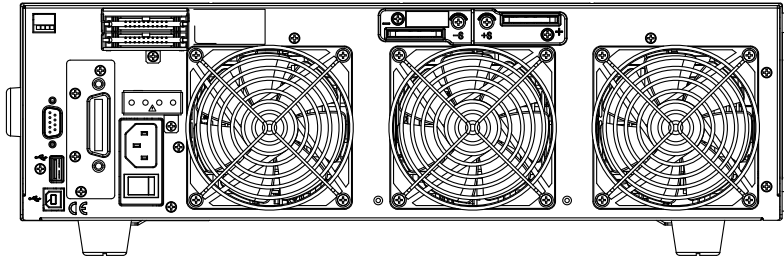
PEL-3021H/ PEL-3041H



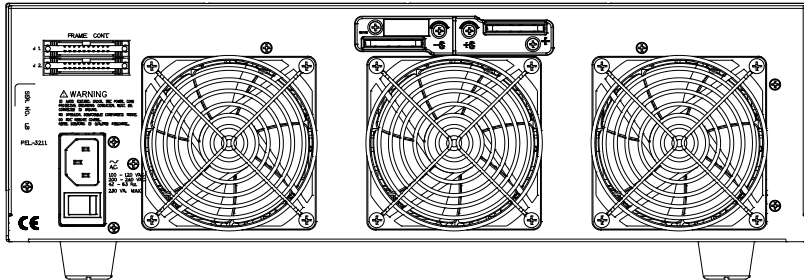
PEL-3111



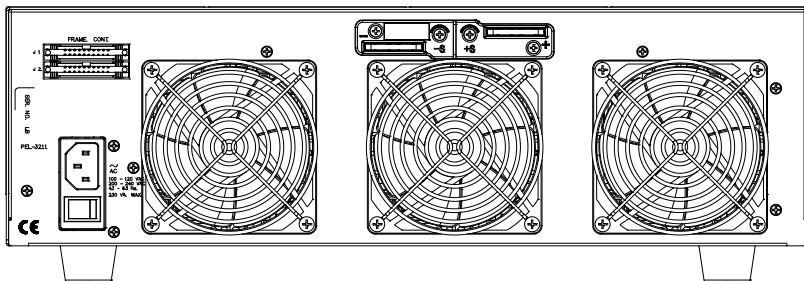
PEL-3111H



PEL-3211 Booster Pack




PEL-3211H Booster Pack





Interface Configuration

Configure to USB Remote Interface



| | | |
|-------------------|----------------------------|--------------------------|
| USB configuration | PC side connector | Type A, host |
| | PEL-3000(H) side connector | Rear panel Type B, slave |
| | Speed | 2.0 (full speed) |
| | USB Class | USB CDC ACM |

| | |
|--|--|
|  Note | Before USB can be used for remote control, it is necessary to install the PEL-3000(H) USB device driver, located on the accompanying User Manual CD. |
|--|--|

- | | |
|-----------|--|
| Operation | <ol style="list-style-type: none"> 1. Connect the USB cable to the rear panel USB B port. 2. Press  +  > <i>Interface[F3]</i> and set the <i>Interface</i> setting to <i>USB</i>. |
|-----------|--|

Configure GPIB Interface

To use GPIB, the optional GPIB port must be installed.

- | | |
|-----------|--|
| Operation | <ol style="list-style-type: none"> 1. Ensure the PEL-3000(H) is off before proceeding. 2. Connect a GPIB cable from a GPIB controller to the GPIB port on the PEL-3000(H). 3. Turn the PEL-3000(H) on. 4. Press  +  > <i>Interface[F3]</i> and set the <i>Interface</i> setting to <i>GPIB</i>. 5. Set the GPIB address. |
| | GPIB address 0-30 |

- GPIB constraints
- *Maximum 15 devices altogether, 20m cable length, 2m between each device*
 - *Unique address assigned to each device*
 - *At least 2/3 of the devices turned On*
 - *No loop or parallel connection*

Pin Assignment



| Pin | Signal | Pin | Signal |
|-----|---------------|-------|---------------|
| 1-4 | Data I/O 1-4 | 13-16 | Data I/O 5-8 |
| 5 | EOI | 17 | REN |
| 6 | DAV | 18 | Ground (DAV) |
| 7 | NRFD | 19 | Ground (NRFD) |
| 8 | NDAC | 20 | Ground (NDAC) |
| 9 | IFC | 21 | Ground (IFC) |
| 10 | SRQ | 22 | Ground (SRQ) |
| 11 | ATN | 23 | Ground (ATN) |
| 12 | SHIELD Ground | 24 | Single GND |

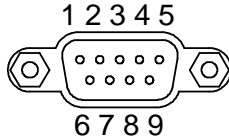
Configure RS232C

| | | |
|----------------------|-----------|--------------------------------|
| RS232C Configuration | Connector | DB-9, Male |
| | Baud Rate | 2400, 4800, 9600, 19200, 38400 |
| | Stop Bit | 1, 2 |
| | Parity | None, Odd, Even |

Operation

1. Connect an RS232C cable from the PC to the rear panel RS232 port.
2. Press **Shift** + **Utility Help** > *Interface*[F3] and set the *Interface* setting to RS232.
3. Set the *Baud Rate*, *Stop Bit* and *Parity* settings.

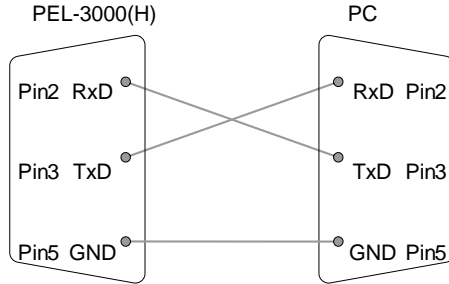
Pin Assignment



- 2: RxD (Receive data)
- 3: TxD (Transmit data)
- 5: GND
- 4, 6 - 9: No connection

PC Connection

Use a null modem connection as shown in the diagram below.



RS232C/USB Remote Control Function Check

Functionality check

Invoke a terminal application such as Realterm.
 For RS-232C, set the COM port, baud rate, stop bit, data bit and parity accordingly.
 To check the COM settings in Windows, see the Device Manager. For example, in WinXP go to the Control panel → System → Hardware tab.

 Note

If you are not familiar with using a terminal application to send/receive remote commands from the serial port or via a USB connection, please page 12 (Using Realterm to Establish a Remote Connection) for more information.

Run this query command via the terminal after the instrument has been configured for RS-232/USB remote control

*idn?

This should return the Manufacturer, Model number, Serial number, and Firmware version in

the following format.

- *GW-INSTEK, PEL-3000(H), XXXXXXXXXXXXX, V.X.X.X.X*

Manufacturer: GW-INSTEK

Model number : PEL-3000(H)

Serial number : XXXXXXXXXXXXX

Firmware version : V.X.X.X



Note

For further details, please see the programming manual, available on the GW Instek web site @ www.gwinstek.com.

Using Realterm to Establish a Remote Connection

Background

Realterm is a terminal program that can be used to communicate with a device attached to the serial port of a PC or via an emulated serial port via USB.

The following instructions apply to version 1.99.0.27. Even though Realterm is used as an example to establish a remote connection, any terminal program can be used that has similar functionality.



Note

Realterm can be downloaded on Sourceforge.net free of charge.

For more information please see <http://realterm.sourceforge.net/>

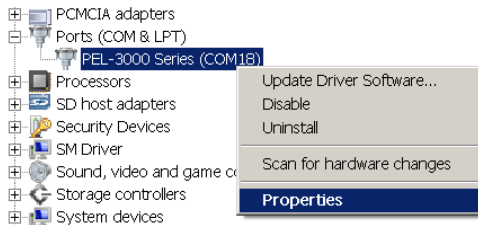
Operation

1. Download Realterm and install according to the instructions on the Realterm website.
2. Connect the PEL-3000(H) via USB (page 9) or via RS232 (page 10).
3. If using RS232, make note of the configured baud rate, stop bits and parity.
4. Go to the Windows device manager and find the COM port number for the connection. For example, go to the Start menu > Control

Panel > Device Manager

Double click the *Ports* icon to reveal the connected serial port devices and the COM port for the each connected device.

If using USB, the baud rate, stop bit and parity settings can be viewed by right-clicking connected device and selecting the *Properties* option.

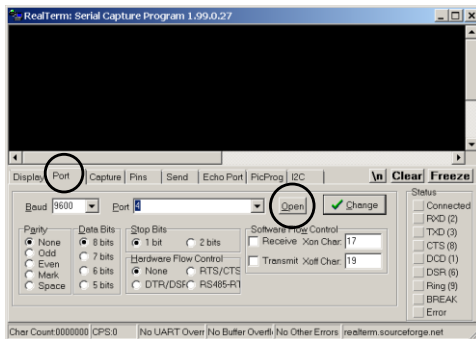


5. Start Realterm on the PC as an administrator.
Click:
Start menu>All Programs>RealTerm>realterm

Tip: to run as an administrator, you can right click the Realterm icon in the Windows Start menu and select the *Run as Administrator* option.
6. After Realterm has started, click on the *Port* tab.

Enter the *Baud, Parity, Data bits, Stop bits* and *Port* number configuration for the connection.

The *Hardware Flow Control, Software Flow Control* options can be left at the default settings.
7. Press *Open* to connect to the PEL-3000(H).



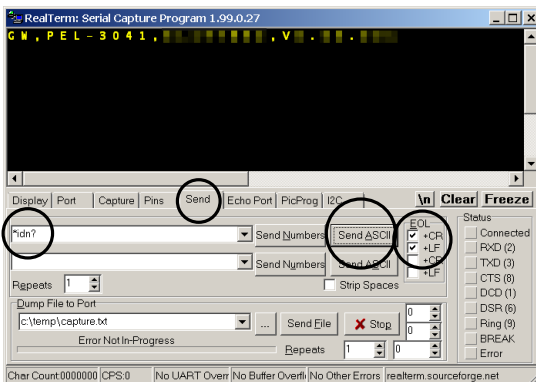
8. Click on the *Send* tab.

In the *EOL* configuration, check on the *+CR* and *+LF* check boxes.

Enter the query:

**idn?*

Click on *Send ASCII*.



9. The terminal display will return the following:

GW, PEL-3XXX(H),EXXXXXXX,VX.XX.XXX

(manufacturer, model, serial number, version)

10. If Realterm fails to connect to the PEL-3000(H), please check all the cables and settings and try again.

GPIO Function Check

Functionality check Please use the National Instruments Measurement & Automation Controller software to confirm GPIO functionality.

See the National Instrument website,
<http://www.ni.com> for details.



Note

For further details, please see the programming manual, available on the GW Instek web site @ www.gwinstek.com.

Operation

1. Start the NI Measurement and Automation Explorer (MAX) program. Using Windows, press:

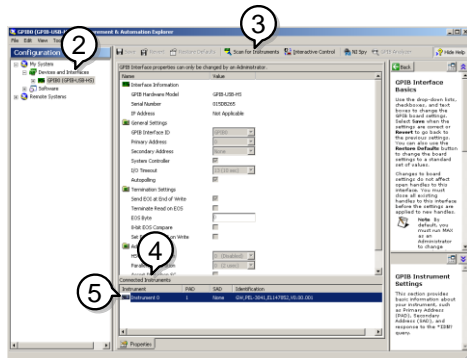


Start>All Programs>National Instruments>Measurement & Automation

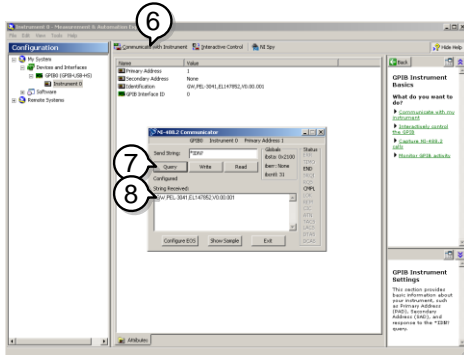


2. From the Configuration panel access;
My System>Devices and Interfaces>GPIO0
3. Press the Scan for Instruments button.

4. In the *Connected Instruments* panel the PEL-3000(H) should be detected as *Instrument 0* with the address the same as that configured on the PEL-3000(H).
5. Double click the *Instrument 0* icon.



6. Click on *Communicate with Instrument*.
7. In the *NI-488.2 Communicator* window, ensure **IDN?* is written in the *Send String:* text box.
Click on the *Query* button to send the **IDN?* query to the instrument.
8. The *String Received* text box will display the query return:
GW, PEL-3XXX(H),EXXXXXXX,VX.XX.XXX
(manufacturer, model, serial number, version)



9. The function check is complete.

C COMMAND OVERVIEW

The Command overview chapter lists all PEL-3000(H) commands in functional order as well as alphabetical order. The command syntax section shows you the basic syntax rules you have to apply when using commands.

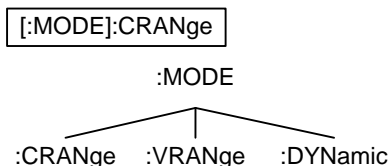
| | |
|--|-----|
| Command Syntax | 19 |
| Command List | 24 |
| Status Register Overview | 203 |
| Introduction to the Status Registers..... | 203 |
| Configuration in the Status register..... | 204 |
| Csummary Status Register Group | 205 |
| Operation Byte Register Group | 207 |
| Questionable Status Register Group..... | 208 |
| Standard Event Status Register Group | 210 |
| Status Byte Register Group..... | 212 |
| Error Messages | 215 |

Command Syntax

| | | |
|---------------------|------------|-----------------------|
| Compatible Standard | IEEE488.2 | Partial compatibility |
| | SCPI, 1999 | Partial compatibility |

Command Structure SCPI (Standard Commands for Programmable Instruments) commands follow a tree-like structure, organized into nodes. Each level of the command tree is a node. Each keyword in a SCPI command represents each node in the command tree. Each keyword (node) of a SCPI command is separated by a colon (:).

For example, the diagram below shows an SCPI sub-structure and a command example.



Command types There are a number of different instrument commands and queries. A command sends instructions or data to the unit and a query receives data or status information from the unit.

Command types

Simple A single command with/without a parameter

Example :CONFigure:SHORT HOLD

Query A query is a simple or compound command followed by a question mark (?). A parameter (data) is returned.

Example :CONFigure:SHORT?

Compound Two or more commands on the same command line. Compound commands are separated with either a semi-colon (;) or a semi-colon and a colon (;:).

A semi-colon is used to join two related commands, with the caveat that the last command must begin at the last node of the first command.

A semi-colon and colon are used to combine two commands from different nodes.

Example CONFigure:VON
MAX;:CONFigure:VDElay MIN

Command Forms Commands and queries have two different forms, long and short. The command syntax is written with the short form of the command in capitals and the remainder (long form) in lower case.

The commands can be written in capitals or lower-case, just so long as the short or long forms are complete. An incomplete command will not be recognized.

Below are examples of correctly written commands.

Long form :CURRent:LEVel?
 :CURRENT:LEVEL?
 :current:level?

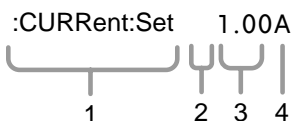
Short form :CURR:LEV?
 :curr:lev?

Square Brackets Commands that contain square brackets indicate that the contents are optional. The function of the command is the same with or without the square bracketed items, as shown below

For example for the query:
 “[:CONFIgure]:GNG [:PASS]?”

Both “:CONFIgure:GNG:PASS?” and “:GNG?” are both valid forms.

Command Format



- 1. Command header
- 2. Space
- 3. Parameter 1
- 4. Unit or suffix.

| Common Unit Parameters | Type | Description | Example |
|------------------------|-----------|------------------|-----------------|
| | <Boolean> | boolean logic | 0, 1 |
| | <NR1> | integers | 0, 1, 2, 3 |
| | <NR2> | decimal numbers | 0.1, 3.14, 8.5 |
| | <NR3> | floating point | 4.5e-1, 8.25e+1 |
| | <NRf> | any of NR1, 2, 3 | 1, 1.5, 4.5e-1 |

[MIN] (Optional parameter) For commands, this will set the setting to the lowest value. This parameter can be used in place of any numerical parameter where indicated.

For queries, it will return the lowest possible value allowed for the particular setting.

[MAX] (Optional parameter) For commands, this will set the setting to the highest value. This parameter can be used in place of any numerical parameter where indicated.

For queries, it will return the highest possible value allowed for the particular setting.

Unit Suffixes (Optional parameters) Unit suffixes can be optionally used with most NRf type input parameters.

| | | |
|---------|--------------------------|----------|
| [A] | Amps | 1.00A |
| [%] | Percentage | 10% |
| [V] | Volts | 5.00V |
| [W] | Watts | 3.00W |
| [ms] | milliseconds | 20ms |
| [mV] | Millivolts | 150mV |
| [s] | Seconds | 5s |
| [mS] | Reciprocal of 1000 ohms | 20mS |
| [OHM] | Ohm | 50OHM |
| [mA/us] | Millamps/ microsecond | 100mA/us |
| [Hz] | Hertz | 1000Hz |

Note: For [OHM] return values, an infinite resistance (open) will be returned as 9.9e³⁷.

| | | |
|--------------------|----|-----------------------|
| Message Terminator | LF | Line feed code (0x0A) |
|--------------------|----|-----------------------|

Command List

| | | |
|--------------------|------------|----|
| Common Commands | *CLS | 32 |
| | *ESE | 32 |
| | *ESR | 33 |
| | *IDN | 33 |
| | *OPC | 34 |
| | *RCL | 34 |
| | *RST | 35 |
| | *SAV | 35 |
| | *SRE | 35 |
| | *STB | 36 |
| | *TRG | 36 |
| | *TST | 37 |

| | | |
|---------------------|--------------------------------|----|
| Trigger Commands | :ABORt | 38 |
| | :INPut[:STATe]:TRIGgered | 39 |
| | :INITiate[:IMMediate] | 39 |
| | :INITiate:CONTInuous | 40 |
| | :TRIGger[:DELay]:TIME | 40 |
| | :TRIGger[:PULSe]:WIDTh | 41 |

| | | |
|----------------|----------------------|----|
| Input Commands | :INPut | 42 |
| | :INPut:MODE | 42 |
| | [:INPut]:SHORT | 43 |

| | | |
|-------------------------|------------------------|----|
| Measurement Commands | :MEASure:CURRent | 44 |
| | :MEASure:ETIMe | 44 |
| | :MEASure:POWer | 44 |
| | :MEASure:VOLTage | 45 |

| | | |
|-----------------------------|----------------------|----|
| Fetch Subsystem Commands | :FETCh:CURRent | 46 |
| | :FETCh:POWer | 46 |
| | :FETCh:VOLTage | 46 |

| | | |
|------------------------------|--------------------------------|----|
| Configure Subsystem Commands | [:CONFigure]:OCP..... | 48 |
| | [:CONFigure]:OPP..... | 49 |
| | [:CONFigure]:UVP..... | 50 |
| | [:CONFigure]:UVP:TIME..... | 50 |
| | [:CONFigure]:OVP..... | 51 |
| | [:CONFigure]:SStart..... | 52 |
| | [:CONFigure]:VON..... | 52 |
| | [:CONFigure]:VDElay..... | 53 |
| | :CONFigure:RESPonse..... | 54 |
| | [:CONFigure]:CNTime..... | 54 |
| | [:CONFigure]:COTime..... | 55 |
| | [:CONFigure]:CRUnit..... | 55 |
| | :CONFigure:DYNamic..... | 56 |
| | :CONFigure:MEMory..... | 56 |
| | :CONFigure:SHORT..... | 57 |
| | :CONFigure:SHORT:SAFety..... | 58 |
| | :CONFigure:SHORT:FUNctIon..... | 58 |
| | [:CONFigure]:GNG:MODE..... | 59 |
| | [:CONFigure]:GNG:H..... | 60 |
| | [:CONFigure]:GNG:L..... | 60 |
| | [:CONFigure]:GNG:C..... | 61 |
| | [:CONFigure]:GNG:DTIME..... | 62 |
| | [:CONFigure]:GNG:SPEctest..... | 62 |
| | [:CONFigure]:GNG:[PASS]..... | 63 |
| <hr/> | | |
| Parallel Command | [:CONFigure]:PARAllel..... | 64 |
| <hr/> | | |
| Step Resolution Commands | :CONFigure:STATus..... | 65 |
| | [:CONFigure]:STEP:CC..... | 66 |
| | [:CONFigure]:STEP:CCH..... | 66 |
| | [:CONFigure]:STEP:CCM..... | 67 |
| | [:CONFigure]:STEP:CCL..... | 67 |
| | [:CONFigure]:STEP:CR..... | 68 |
| | [:CONFigure]:STEP:CRH..... | 68 |
| | [:CONFigure]:STEP:CRM..... | 69 |
| | [:CONFigure]:STEP:CRL..... | 70 |
| | [:CONFigure]:STEP:CV..... | 70 |
| | [:CONFigure]:STEP:CVH..... | 71 |
| | [:CONFigure]:STEP:CVL..... | 71 |
| | [:CONFigure]:STEP:CP..... | 72 |

| | | |
|------------------|--------------------------------------|----|
| | [:CONFigure]:STEP:CPH | 72 |
| | [:CONFigure]:STEP:CPM..... | 73 |
| | [:CONFigure]:STEP:CPL | 74 |
| <hr/> | | |
| External Control | [:CONFigure]:EXTernal[:CONTRol]..... | 75 |
| Commands | [:CONFigure]:EXTernal:CV | 75 |
| | [:CONFigure]:EXTernal:LOADonin | 76 |
| <hr/> | | |
| Mode Subsystem | :MODE | 77 |
| Commands | [:MODE]:CRANge | 77 |
| | [:MODE]:VRANge..... | 78 |
| | [:MODE]:RESPonse | 78 |
| | [:MODE]:DYNamic | 79 |
| <hr/> | | |
| Current | :CURRent[:VA] | 80 |
| Subsystem | :CURRent[:VA]:TRIGgered | 81 |
| Commands | :CURRent:VB | 81 |
| | :CURRent:SRATe | 82 |
| | :CURRent:L1 | 82 |
| | :CURRent:L2..... | 83 |
| | :CURRent:SET | 84 |
| | :CURRent:LEVel..... | 84 |
| | :CURRent:RISE..... | 85 |
| | :CURRent:FALL | 86 |
| | :CURRent:T1 | 86 |
| | :CURRent:T2 | 87 |
| | :CURRent:FREQUency | 88 |
| | :CURRent:DUTY..... | 88 |
| | :CURRent:RECall..... | 89 |
| <hr/> | | |
| Resistance | :RESistance[:VA] | 90 |
| Subsystem | :RESistance[:VA]:TRIGgered | 91 |
| Commands | :RESistance:VB | 91 |
| | :RESistance:SRATe..... | 92 |
| | :RESistance:L1 | 93 |
| | :RESistance:L2..... | 93 |
| | :RESistance:SET..... | 94 |
| | :RESistance:LEVel | 95 |
| | :RESistance:RISE | 95 |
| | :RESistance:FALL..... | 96 |

| | | |
|--|----------------------------------|-----|
| | :RESistance:T1..... | 97 |
| | :RESistance:T2..... | 97 |
| | :RESistance:FREQuency..... | 98 |
| | :RESistance:DUTY..... | 99 |
| | :CONductance[:VA]..... | 99 |
| | :CONductance[:VA]:TRIGgered..... | 100 |
| | :CONductance:VB..... | 100 |
| | :CONductance:L1..... | 101 |
| | :CONductance:L2..... | 102 |
| | :CONductance:SET..... | 103 |
| | :CONductance:RECall..... | 103 |
| | :RESistance:RECall..... | 104 |

| | | |
|-----------|----------------------|-----|
| Voltage | :VOLTage[:VA]..... | 105 |
| Subsystem | :VOLTage:VB..... | 105 |
| Commands | :VOLTage:RECall..... | 106 |

| | | |
|-----------------|-----------------------|-----|
| Power Subsystem | :POWER[:VA]..... | 107 |
| Commands | :POWER:VB..... | 108 |
| | :POWER:L1..... | 108 |
| | :POWER:L2..... | 109 |
| | :POWER:SET..... | 110 |
| | :POWER:LEVel..... | 110 |
| | :POWER:T1..... | 111 |
| | :POWER:T2..... | 112 |
| | :POWER:FREQuency..... | 112 |
| | :POWER:DUTY..... | 113 |
| | :POWER:RECall..... | 114 |

| | | |
|----------|---------------------------------------|-----|
| Program | :FUNctIon[:COMplete][:RING]:TIME..... | 115 |
| Commands | :PROGram:STATe..... | 116 |
| | :PROGram..... | 117 |
| | :PROGram[:RECall]:DEFault..... | 118 |
| | :PROGram:STARt..... | 118 |
| | :PROGram:STEP..... | 119 |
| | :PROGram:MEMory..... | 119 |
| | :PROGram:RUN..... | 120 |
| | :PROGram:ONTime..... | 120 |
| | :PROGram:OFFTime..... | 121 |
| | :PROGram:PFTime..... | 121 |
| | :PROGram:STIME..... | 122 |

| | |
|--|-----|
| [:PROGram]:CHAI:n:STARt | 123 |
| [:PROGram]:CHAI:n | 123 |
| [:PROGram]:CHAI:n:P2P | 124 |
| [:PROGram]:CHAI:n[:RECall]:DEFault | 125 |
| :PROGram:SAVE | 125 |

| | | |
|-----------------|------------------------------|-----|
| Normal sequence | :NSEquence:STATe | 126 |
| Commands | :NSEquence | 127 |
| | :NSEquence:STARt | 129 |
| | :NSEquence:NUMBER | 130 |
| | :NSEquence:MEMO | 130 |
| | :NSEquence:MODE | 131 |
| | :NSEquence:RANGe | 131 |
| | :NSEquence:LOOP | 132 |
| | :NSEquence:LLOAD | 133 |
| | :NSEquence:LAST | 133 |
| | :NSEquence:CHAI:n | 134 |
| | :NSEquence:EDIT | 134 |
| | :NSEquence:EDIT:POINT | 136 |
| | :NSEquence:EDIT:END | 136 |
| | :NSEquence[:DELet]:ALL | 136 |
| | :NSEquence:SAVE | 137 |
| | :NSEquence:COTime | 137 |

| | | |
|---------------|------------------------------|-----|
| Fast sequence | :FSEquence:STATe | 138 |
| Commands | :FSEquence | 139 |
| | :FSEquence:MEMO | 140 |
| | :FSEquence:MODE | 141 |
| | :FSEquence:RANGe | 141 |
| | :FSEquence:LOOP | 142 |
| | :FSEquence:TBASE | 143 |
| | :FSEquence:LLOAD | 143 |
| | :FSEquence:LAST | 144 |
| | :FSEquence:RPTStep | 144 |
| | :FSEquence:EDIT | 145 |
| | :FSEquence:EDIT:POINT | 146 |
| | :FSEquence:EDIT:END | 146 |
| | :FSEquence[:DELet]:ALL | 146 |
| | :FSEquence[:EDIT]:FILL | 147 |
| | :FSEquence:SAVE | 147 |

| | | |
|----------|----------------------------------|-----|
| OCP test | :OCP:STATe..... | 149 |
| Commands | :OCP:EDIT[:CHANnel]..... | 150 |
| | :OCP[:CHANnel]:NUMBer..... | 151 |
| | :OCP:MEMO..... | 151 |
| | :OCP[:CHANnel]:RANGe..... | 152 |
| | :OCP[:CHANnel]:STARt..... | 152 |
| | :OCP[:CHANnel]:END..... | 153 |
| | :OCP[:CHANnel]:STEP:CURRent..... | 153 |
| | :OCP[:CHANnel]:STEP:TIME..... | 154 |
| | :OCP[:CHANnel]:DELay..... | 154 |
| | :OCP[:CHANnel]:TRIGger..... | 155 |
| | :OCP[:CHANnel]:LAST..... | 155 |
| | :OCP:CHANnel:STATus..... | 156 |
| | :OCP:RESult..... | 156 |
| | :OCP:RUN..... | 157 |

| | | |
|----------|-------------------------------|-----|
| OPP test | :OPP:STATe..... | 158 |
| Command | :OPP:EDIT[:CHANnel]..... | 159 |
| | :OPP[:CHANnel]:NUMBer..... | 160 |
| | :OPP:MEMO..... | 160 |
| | :OPP[:CHANnel]:RANGe..... | 161 |
| | :OPP[:CHANnel]:STARt..... | 161 |
| | :OPP[:CHANnel]:END..... | 162 |
| | :OPP[:CHANnel]:STEP:WATT..... | 163 |
| | :OPP[:CHANnel]:STEP:TIME..... | 163 |
| | :OPP[:CHANnel]:DELay..... | 164 |
| | :OPP[:CHANnel]:TRIGger..... | 164 |
| | :OPP[:CHANnel]:LAST..... | 165 |
| | :OPP:CHANnel:STATus..... | 166 |
| | :OPP:RESult..... | 166 |
| | :OPP:RUN..... | 166 |

| | | |
|-----------|---------------------------------|-----|
| BATT test | :BATTery:STATe..... | 167 |
| Command | :BATT:EDIT..... | 168 |
| | :BATTery [:CHANnel]:NUMBer..... | 169 |
| | :BATTery:MEMO..... | 170 |
| | :BATTery:MODE..... | 170 |
| | :BATTery:RANGe..... | 171 |
| | :BATTery:VALue..... | 171 |
| | :BATTery:RISE..... | 172 |
| | :BATTery:FALL..... | 172 |

| | |
|------------------------------|-----|
| :BATTery:STOP:VOLTage..... | 173 |
| :BATTery:STOP:TIME | 174 |
| :BATTery:STOP:AH | 174 |
| :BATTery:DATAlog:TIMer | 175 |
| :BATT:CHANnel:STATus | 175 |
| :BATT:RESult..... | 176 |
| :BATT:RUN..... | 176 |

| | | |
|------------------|---------------------------|-----|
| Utility Commands | :UTILity:SYSTem..... | 177 |
| | :UTILity:LOAD | 178 |
| | :UTILity:LOAD:MODE | 178 |
| | :UTILity:LOAD:RANGe | 179 |
| | :UTILity:TIME | 179 |
| | :UTILity:KNOB..... | 180 |
| | :UTILity:SPEAker..... | 181 |
| | :UTILity:ALARm..... | 181 |
| | :UTILity:UNReg..... | 182 |
| | :UTILity:GNG..... | 183 |
| | :UTILity:CONTrast..... | 183 |
| | :UTILity:BRIGhtness..... | 184 |
| | :UTILity:LANGUage | 184 |
| | :UTILity:REMote..... | 185 |
| | :UTILity:REMote:MODE..... | 185 |

| | | |
|-----------|-------------------------|-----|
| Interface | :UTILity:INTerface..... | 186 |
| Commands | :UTILity:BRATe..... | 186 |
| | :UTILity:SBIT | 187 |
| | :UTILity:PARity | 187 |

| | | |
|---------------|-----------------------------|-----|
| File Commands | :MEMory:SAVE | 189 |
| | :MEMory:RECall..... | 189 |
| | :PREset:SAVE | 190 |
| | :PREset:RECall..... | 190 |
| | :SETup:SAVE | 190 |
| | :SETup:RECall..... | 190 |
| | :FACTory[:RECall] | 191 |
| | :USER[:DEFault]:SAVE | 191 |
| | :USER[:DEFault]:RECall..... | 191 |

| | | |
|-------------|----------------------|-----|
| SCPI Status | :SYSTem:ERRor | 192 |
| Commands | :STATus:PRESet | 192 |

| | | |
|-----------------|------------------------------------|-----|
| Csummary Status | :STATus:CSUMmary:CONDition | 194 |
| Commands | :STATus:CSUMmary:ENABle | 194 |
| | :STATus:CSUMmary[:EVENT] | 195 |
| | :STATus:CSUMmary:NTRansition | 195 |
| | :STATus:CSUMmary:PTRansition | 195 |

| | | |
|------------------|--|-----|
| Operation Status | :STATus:QUESTionable:CONDition | 200 |
| Commands | :STATus:QUESTionable:ENABle | 200 |
| | :STATus:QUESTionable[:EVENT] | 201 |
| | :STATus:QUESTionable:NTRansition | 201 |
| | :STATus:QUESTionable:PTRansition | 201 |

| | | |
|---------------------|--|-----|
| Questionable Status | :STATus:QUESTionable:CONDition | 200 |
| Commands | :STATus:QUESTionable:ENABle | 200 |
| | :STATus:QUESTionable[:EVENT] | 201 |
| | :STATus:QUESTionable:NTRansition | 201 |
| | :STATus:QUESTionable:PTRansition | 201 |

Common Commands

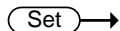
| | |
|------------|----|
| *CLS | 32 |
| *ESE | 32 |
| *ESR | 33 |
| *IDN | 33 |
| *OPC | 34 |
| *RCL | 34 |
| *RST | 35 |
| *SAV | 35 |
| *SRE | 35 |
| *STB | 36 |
| *TRG | 36 |
| *TST | 37 |

*CLS



| | |
|-------------|--|
| Description | Clears all Event registers and queues. |
| Syntax | *CLS |
| Example | *CLS Clears all Event registers and queues. |

*ESE



| | |
|--------------|--|
| Description | Queries or sets the Standard Event Status Enable register. The Standard Event Status Enable register determines which events can set the Event Summary bit (ESB) in the Status Byte Register. Any bits that are set to 1 enable the corresponding event. Each event is represented by a bit in the Standard Event Status Enable register. Refer to section “ Standard Event Status register group ” on page 210 for more information on bit. |
| Syntax | *ESE <NRF> |
| Query Syntax | *ESE? |

| | | |
|------------------|---|--|
| Parameter | <NR1> 1 | Sets the Standard Event Status Enable register. |
| Return parameter | Return in "<NR1>" the set value of the Standard Event Status Enable register. | |
| Example | *ESE 8 | Sets bit 3 of the ESE register. |
| Query example | *ESE? >12 | Bits 2 and 3 are set in the Standard Event Status Enable register. |

***ESR**

→ Query

| | |
|------------------|---|
| Description | Reads the Standard Event Status register. This command will also clear the Standard Event Status register. Refer to section " Standard Event Status register group " on page 210 for more information on bit. |
| Query Syntax | *ESR? |
| Return parameter | Return in "<NR1>" the set value of the Standard Event Status register. |
| Query example | *ESR? >48 Bits 5 and 6 are set in the Standard Event register. |

***IDN**

→ Query

| | |
|--------------|--|
| Description | Queries the manufacturer, model number, serial number, and firmware version of the instrument. |
| Query Syntax | *IDN? |

| | | |
|------------------|----------|---------------------------------|
| Return parameter | <string> | Returns the manufacturer name. |
| | <string> | Returns the model number. |
| | <NR1> | Returns the serial number. |
| | <string> | Returns the version of firmware |

Query example * IDN?
 > GWInstek, PEL-3111H,12345678,V1.01.001
 It is a response equipment manufacturer, model number, serial number, and firmware version.

Set →

***OPC**

→ Query

Description This command sets the OPC (Operation Command Bit) bit (bit 0) of the Standard Event Status Register after the instrument has completed all pending operations. The query will return the status of the OPC bit.

Syntax *OPC

Query Syntax *OPC?

Return parameter 1 Operation complete

Example *OPC

Query Example *OPC?
 >1
 Indicates that all pending operations are complete.

***RCL**

Set →

Description The Recall Instrument State command restores the instrument settings from a previously saved memory setting.

Syntax *RCL <NR1>

Parameter <NR1> Memory number 1 to 256

Example *RCL 20
 Recall setting memory 20.

Same function command :MEMory:RECall

***RST**

Set →

Description Resets the unit. This is command forces the ABORt, and *CLS

Syntax *RST

Example *RST
Resets the unit.

***SAV**

Set →

Description The Save Instrument State command saves the instrument settings to one of the memory setting slots.

Syntax *SAV <NR1>

Parameter <NR1> Memory number 1 to 256

Example *SAV 20
Saves the current setting to memory 20.

Same function command :MEMory:SAVe

***SRE**

Set →

→ Query

Description Queries or sets the Service Request Enable register. The Service Request Enable register determines which events in the Status Byte register can set the Master Summary bit (MSB) in the Status Byte Register. Any bits that are set to 1 will cause the MSS bit to be set.

Refer to section “**Status byte register group**” on page 212 for more information on bit.

Syntax *SRE <NRf>

| | | |
|------------------|--|--|
| Query Syntax | *SRE? | |
| Parameter | <NR1> | Sets the set value of the Service Request Enable register. |
| Return parameter | Return in "<NR1>" the set value of the Service Request Enable register. | |
| Example | *SRE 8 | Sets bit 3 of the Service Request Enable register. |
| Query example | *SRE? >12 Bits 2 and 3 are set in the Service Request Enable register. | |

***STB**

→ Query

| | | |
|------------------|--|--|
| Description | <p>Reads the Status Byte register. This command will not clear the Status Byte register.</p> <p>If the Master Summary Status bit (MSS) is set, it indicates that there is a reason for a service request.</p> <p>Refer to section “Status byte register group” on page 212 for more information on bit.</p> | |
| Query Syntax | *STB? | |
| Return parameter | Return in "<NR1>" the value of the Status Byte register. | |
| Query example | *STB? >36 Bits 2 and 5 are set in the Status Byte register. | |

***TRG**

Set →

| | | |
|-------------|---------------------------------|--|
| Description | This command triggers the unit. | |
| Syntax | *TRG | |

| | |
|------------------|---|
| Example | *TRG Issue the enforcement trigger. |
| Related Commands | :INITiate:CONTinuous, :INITiate[:IMMediate] |

***TST**

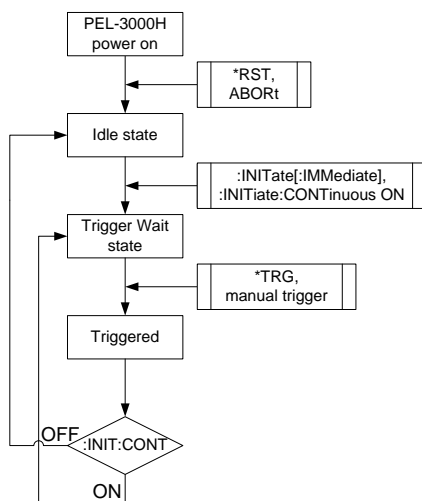
→ Query

| | |
|------------------|---|
| Description | This command is a standard SCPI self-test command. The PEL-3000(H) does not perform any self-tests so will always return 0 (pass) for this command. |
| Query Syntax | *TST? |
| Return parameter | <NR1> Pass |
| Query example | *TST? >0 |

Trigger Commands

| | |
|--------------------------------|----|
| :ABORt | 38 |
| :INPut[:STATe]:TRIGgered | 39 |
| :INITiate[:IMMediate] | 39 |
| :INITiate:CONTinuous | 40 |
| :TRIGger[:DELay]:TIME | 40 |
| :TRIGger[:PULSe]:WIDTh | 41 |

Trigger States



:ABORt

Set →

| | |
|--------------|---|
| Description | Clears the trigger wait status and returns to the idle state. |
| Query Syntax | :ABORt |
| Example | :ABOR Clears the trigger wait status. |

Set →
→ Query

:INPut[:STATe]:TRIGgered


| | | |
|------------------|---|---|
| Description | Sets whether to turn on the load input when the trigger is activated. Sets when PEL-3000H is on idle state. | |
| Syntax | :INPut[:STATe]:TRIGgered {<Boolean> OFF ON } | |
| Query Syntax | :INPut[:STATe]:TRIGgered? | |
| Parameter | OFF or 0 | Not change the load input in when the trigger active. |
| | ON or 1 | Turn on the load input in when the trigger active. |
| Example | :INP:TRIG ON Turn on the load input in when the trigger active. | |
| Related Commands | *TRG, :INITiate:CONTInuous, :INITiate[:IMMEDIATE] | |

Set →

:INITiate[:IMMEDIATE]

| | | |
|------------------|---|--|
| Description | Sets the trigger to the wait state. If the trigger is activated, the trigger will automatically go to the idle state. | |
| Query Syntax | :INITiate[:IMMEDIATE] | |
| Example | :INIT Sets the trigger to the wait state. | |
| Related Commands | *TRG, :INPut[:STATe]:TRIGgered, :CURRent[:VA]:TRIGgered, :RESistance[:VA]:TRIGgered | |

:INITiate:CONTInuous (Set) →
→ (Query)

| | | |
|--|--|--|
| Description | Sets or queries for state of the continuous waiting for the trigger. | |
|  Note | Release of the trigger wait state requires activation of the trigger. | |
| Syntax | :INITiate:CONTInuous {<Boolean> OFF ON } | |
| Query Syntax | :INITiate:CONTInuous? | |
| Parameter | OFF or 0 | Remove the continuous waiting for the trigger. |
| | ON or 1 | Sets to continuous waiting for the trigger. |
| Example | :INIT:CONT ON Sets to continuous waiting for the trigger. | |
| Query example | :INITiate:CONT? >1 Setting in a continuous wait for trigger, and wait for the trigger. | |
| Related Commands | *TRG, :INPut[:STATe]:TRIGgered, :CURRent[:VA]:TRIGgered, :RESistance[:VA]:TRIGgered | |

:TRIGger[:DELay]:TIME (Set) →
→ (Query)

| | | |
|--------------|--|-------------------------|
| Description | The command determines how long to delay any action after a trigger is received. Equivalent to using the “Trig In Delay” setting on the front panel. | |
| Syntax | :TRIGger[:DELay]:TIME <NR2> MINimum MAXimum | |
| Query Syntax | :TRIGger[:DELay]:TIME? [MINimum MAXimum] | |
| Parameter | <NR2> | 0 ~ 0.005s (0 ~ 5000us) |
| | MINimum | Minimum delay time |

| | | |
|---|---|-----------------------------------|
| | MAXimum | Maximum delay time |
| Return parameter | Returns the delay time | |
| Example | :TRIG:TIME 0.005 Sets the trigger in delay to 5ms. | |
| Query example | :TRIG:TIME? >0.0050000 Returns the delay time in seconds. | |
| <div style="display: inline-block; border: 1px solid black; border-radius: 10px; padding: 2px 5px;">Set</div> → → <div style="border: 1px solid black; border-radius: 10px; padding: 2px 5px;">Query</div> | | |
| :TRIGger[:PULSe]:WIDTh | | |
| Description | Sets and queries for the trigger output signal's pulse width. | |
| Syntax | :TRIGger[:PULSe]:WIDTh <NR2> MINimum MAXimum | |
| Query Syntax | :TRIGger[:PULSe]:WIDTh? [MINimum MAXimum] | |
| Parameter | <NR2> | 0.0000025~0.005s (2.5us ~ 5000us) |
| | MINimum | |
| | MAXimum | |
| Return parameter | Returns the pulse width | |
| Example | :TRIG:WIDTh 0.005 Sets the trigger pulse width to 5ms. | |
| Query example | TRIG:WIDTh? >0.0050000 Returns the pulse width of the trigger output. | |

Input Commands

| | |
|---------------------|----|
| :INPut..... | 42 |
| :INPut:MODE..... | 42 |
| [:INPut]:SHORT..... | 43 |

:INPut

Set →

→ Query

| | | |
|------------------|---|--|
| Description | Sets and queries for the status of the load. Setting stop and restart of program, sequence, OCP test. | |
| Syntax | :INPut {<Boolean> OFF ON } | |
| Query Syntax | :INPut? | |
| Parameter | OFF or 0 | Sets the off the load input setting. Sets stop of program, sequence, OCP test. |
| | ON or 1 | Sets the on the load input setting. Sets restart of program, sequence, OCP test. |
| Return parameter | Return in "<Boolean>" the set value of the load input. | |
| Example | :INP ON the on the load input setting. | |
| Query example | :INP? >1 Load input setting is on. | |

:INPut:MODE

Set →

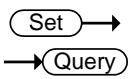
→ Query

| | | |
|--------------|--|------------------------------------|
| Description | Sets and queries for the operating function of the load. | |
| Syntax | :INPut:MODE{LOAD PROG NSEQ FSEQ} | |
| Query Syntax | :INPut:MODE | |
| Parameter | LOAD | Sets the manual operation. |
| | PROG | Sets the program function. |
| | NSEQ | Sets the normal sequence function. |

| | | |
|--|-------------|----------------------------------|
| | FSEQ | Sets the fast sequence function. |
|--|-------------|----------------------------------|

| | | |
|---------|------------------|-------------------------------|
| Example | :INPut:MODE LOAD | |
| | | Sets to the manual operation. |

| | | |
|---------------|--------------|--|
| Query example | :INPut:MODE? | |
| | >LOAD | |
| | | Mode of operation is the manual operation. |



[[:INPut]:SHORT

| | | |
|-------------|---|--|
| Description | Sets and queries for the input terminals state (open or short). | |
|-------------|---|--|

| | | |
|--------|--------------------------------------|--|
| Syntax | [:INPut]:SHORT {<Boolean> OFF ON } | |
|--------|--------------------------------------|--|

| | | |
|--------------|-----------------|--|
| Query Syntax | [:INPut]:SHORT? | |
|--------------|-----------------|--|

| | | |
|-----------|----------|-----------------|
| Parameter | OFF or 0 | Sets the open. |
| | ON or 1 | Sets the short. |

| | | |
|------------------|--|--|
| Return parameter | Return in "<Boolean>" the input terminals state. | |
|------------------|--|--|

| | | |
|---------|----------|-----------------|
| Example | :SHOR ON | |
| | | Sets the short. |

| | | |
|---------------|--------|-------------------------------------|
| Query example | :SHOR? | |
| | >1 | |
| | | The input terminals state is short. |

Measurement Commands

| | |
|------------------------|----|
| :MEASure:CURRent..... | 44 |
| :MEASure:ETIME..... | 44 |
| :MEASure:POWer | 44 |
| :MEASure:VOLTage | 45 |

:MEASure:CURRent → Query

| | |
|------------------|--|
| Description | Query of current measurement. |
| Query Syntax | :MEASure:CURRent? |
| Return parameter | Return the current measurement (<NR2>). The unit is the [A]. |
| Query example | :MEAS:CURR? >0.50000 Current measurement is 0.5A. |

:MEASure:ETIME → Query

| | |
|------------------|--|
| Description | Query of the elapsed time of the load-on. |
| Query Syntax | :MEASure:ETIME? |
| Return parameter | Return the elapsed time (<NR2>) of the load-on. The unit is the seconds. |
| Query example | :MEAS:ETIM? >10.0 The elapsed time of the load-on is 10 seconds. |

:MEASure:POWer → Query

| | |
|------------------|--|
| Description | Query of power measurement. |
| Query Syntax | :MEASure:POWer? |
| Return parameter | Return the power measurement (<NR2>). The unit is the [W]. |

Query example :MEAS:POWer?
 >15.00000
 Power measurement is 15W.

:MEASure:VOLTage → Query

Description Query of voltage measurement.

Query Syntax :MEASure:VOLTage?

Return parameter Return the voltage measurement (<NR2>). The unit is the [V].

Query example :MEAS:VOLT?
 >5.00000
 Voltage measurement is 5V.

Fetch Subsystem

| | |
|----------------------|----|
| :FETCh:CURRent..... | 46 |
| :FETCh:POWer..... | 46 |
| :FETCh:VOLTage | 46 |

:FETCh:CURRent

→ Query

Description This query returns the real-time current of the load input.

Query syntax :FETCh:CURRent?

Return parameter Returns the real-time current .The unit is [A]

Query example :FETC:CURR?
>0.5000

The load has a current of 0.5 amps at the input.

:FETCh:POWer

→ Query

Description This query returns the real-time power of the load input.

Query syntax :FETCh:POWer?

Return parameter Returns the real-time power. The unit is [W].

Query example :FETC:POW?
>15.00000

The load is at 15 watts.

:FETCh:VOLTage

→ Query

Description This query returns the real-time voltage of the load input.

Query syntax :FETCh:VOLTage?

Return parameter Returns the real-time voltage. The unit is [V].

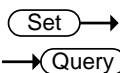
Query example :FETC:VOLT?
 >5.00000

The load has a voltage of 5 volts at the input.

Configure Subsystem Commands

| | |
|---------------------------------|----|
| [:CONFigure]:OCP | 48 |
| [:CONFigure]:OPP | 49 |
| [:CONFigure]:UVP | 50 |
| [:CONFigure]:UVP:TIME..... | 50 |
| [:CONFigure]:OVP | 51 |
| [:CONFigure]:SStart..... | 52 |
| [:CONFigure]:VON..... | 52 |
| [:CONFigure]:VDElay..... | 53 |
| :CONFigure:RESpense..... | 54 |
| [:CONFigure]:CNTime | 54 |
| [:CONFigure]:COTime | 55 |
| [:CONFigure]:CRUnit | 55 |
| :CONFigure:DYNamic | 56 |
| :CONFigure:MEMory | 56 |
| :CONFigure:SHORt..... | 57 |
| :CONFigure:SHORt:SAFety..... | 58 |
| :CONFigure:SHORt:FUNcTion | 58 |
| [:CONFigure]:GNG:MODE | 59 |
| [:CONFigure]:GNG:H..... | 60 |
| [:CONFigure]:GNG:L..... | 60 |
| [:CONFigure]:GNG:C | 61 |
| [:CONFigure]:GNG:DTIME..... | 62 |
| [:CONFigure]:GNG:SPECtest | 62 |
| [:CONFigure]:GNG[:PASS]..... | 63 |

[:CONFigure]:OCP



| | |
|--------------|--|
| Description | Sets and queries for the OCP trip settings. The OCP limit can be set to a specific value or the trip setting can be set to either limit the current or to turn the load off. |
| Syntax | [:CONFigure]:OCP {<NRf>[A] MINimum MAXimum LIMit LOFF} |
| Query Syntax | [:CONFigure]:OCP? |

| | |
|----------|------------------------------|
| <NRf>[A] | Current limit value. |
| MINimum | Minimum current limit value. |
| MAXimum | Maximum current limit value. |
| LIMit | Limit the current. |
| LOFF | Turn the load off. |

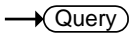
Return parameter Returns OCP setting followed by the OCP value, by the "{Load off | LIMIT},<NR2>" string.

Example1 :OCP LIM
Sets the OCP setting to limit.

Example2 :OCP 19.250
Sets the OCP value to 19.25A.

Query example :OCP?
>LIMIT, 19.250
The OCP setting is LIMIT and the OCP value is 19.25A.

[[:CONFigure]:OPP



Description Sets or queries for the OPP trip settings. The OPP limit can be set to a specific value or the trip setting can be set to either limit the power or to turn the load off.

Syntax [[:CONFigure]:OPP {<NRf> [W] | MINimum | MAXimum | LIMit | LOFF}

Query Syntax [[:CONFigure]:OPP?

| | | |
|-----------|----------|-----------------------------|
| Parameter | <NRf>[W] | Power limit value. |
| | MINimum | Minimum power limits value. |
| | MAXimum | Maximum power limits value. |
| | LIMit | Limit the power |
| | LOFF | Turn the load off |

Return parameter Returns a string with OPP setting followed by the OPP value, by the "{Load off | LIMIT},<NR2>" string.

| | |
|---------------|---|
| Example1 | :OPP LIM Sets the OCP setting to limit. |
| Example2 | :OPP 10.000 Sets the OPP value to 10W. |
| Query example | :OPP? >LIMIT, 10.000 The OPP setting is limited and the OPP value is 10.000W. |

Set →

→ Query

[:CONFigure]:UVP

| | | |
|------------------|--|--------------------------------------|
| Description | Sets or queries for the UVP trip settings. | |
| Syntax | [:CONFigure]:UVP {<NRf>[V] MINimum MAXimum } | |
| Query Syntax | [:CONFigure]:UVP? | |
| Parameter | <NRf>[V] | Under voltage limit value. ; 0 = OFF |
| | MINimum | Minimum value. (UVP setting is OFF.) |
| | MAXimum | Maximum value. |
| Return parameter | Return the set value of the UVP, by the "<NR2>" string | |

| | |
|---------------|--|
| Example1 | :UVP 10.0 Sets the UVP setting to 10V. |
| Query example | :UVP? > 10.0 The UVP setting is 10V. |

Set →

→ Query

[:CONFigure]:UVP:TIME

| | | |
|--------------|--|--|
| Description | Sets and queries for the UVP ring time setting. | |
| Syntax | [:CONFigure]:UVP:TIME {<NR1> MINimum MAXimum INFINITY} | |
| Query Syntax | [:CONFigure]:UVP:TIME? [MINimum MAXimum] | |

| | | |
|------------------|--|---|
| Parameter | <NR1>[s] MINimum MAXimum INfinity | The ring time in seconds(0~600); 0 = OFF Minimum ring time Maximum ring time Sets the ring time to infinity. |
| Return parameter | <NR1> Infinity OFF | The ring time in seconds Infinite Function complete ring time is off. |
| Example | :UVP:TIME 5 | Sets the UVP ring time to 5 seconds. |
| Query example | :UVP:TIME? >5 | The UVP ring time is 5 seconds. |

[[:CONFigure]:OVP




| | | |
|------------------|--|---|
| Description | Sets or queries for the OVP trip settings. | |
| Syntax | [:CONFigure]:OVP {<NRf>[V] MINimum MAXimum} | |
| Query Syntax | [:CONFigure]:OVP? | |
| Parameter | <NRf>[V] MINimum MAXimum | Over voltage limit value. Minimum value. Maximum value. (OVP setting is OFF.) |
| Return parameter | Return the set value of the OVP, by the "{<NR2> OFF}" string. "OFF" is a function off. | |
| Example1 | :OVP 10.00 | Sets the OVP setting to 10V. |
| Query example | :OVP? > 10.0000 | The OVP setting is 10.0000V. |

Set →

→ Query

[:CONFigure]:SStart

| | | |
|------------------|--|---------------------------------|
| Description | Sets and queries for the Soft Start time setting. | |
| Syntax | [:CONFigure]:SStart {<NRf>[s] MINimum MAXimum OFF} | |
| Query Syntax | [:CONFigure]:SStart? | |
| Parameter | <NRf>[s] | The soft start time in seconds. |
| | MINimum | Minimum time = 0 second |
| | MAXimum | Maximum time |
| | OFF | OFF = 0 second |
| Return parameter | Return the set value of the soft-start time, by the "{<NR2> OFF}" string. "OFF" is a function off. | |
| Example | :SSt OFF Turns the soft start function off. | |
| Query example | :SSt? >OFF The soft start function is off. | |

Set →

→ Query

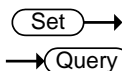
[:CONFigure]:VON

| | | |
|--------------|---|----------------------------|
| Description | Sets and queries for the Von voltage settings and latch. | |
| Syntax | [:CONFigure]:VON {<NRf>[V] MINimum MAXimum LON LOFF} | |
| Query Syntax | [:CONFigure]:VON? | |
| Parameter | {<NRf>[V] | The Von voltage value. |
| | MINimum | Minimum Von voltage value. |
| | MAXimum | Maximum Von voltage value. |
| | LON | Latch on |
| | LOFF | Latch off |

Return parameter Return the Von value and the mode of operation, by the " Latch:{OFF | ON},<NR2>" string.

Example :VON 10.0V LON
Sets the Von voltage value to 10.0 volts. And Von latch ON.

Query example :VON?
>Latch OFF, 0.00
The Von voltage value is 0V.



[[:CONFIgure]:VDElay

Description Sets and queries for the Von Delay settings in seconds.

Syntax [[:CONFIgure]:VDElay {<NRf>[s] | MINimum | MAXimum}

Query Syntax [[:CONFIgure]:VDElay?

| | | |
|------------------|----------|---------------------------|
| Parameter | <NRf>[s] | The delay time in seconds |
| | OFF | Disable the delay time |
| | MINimum | Minimum delay time |
| | MAXimum | Maximum delay time |

Return parameter Return the set value of the delay time, by the "{<NR2> | OFF}" string. "OFF" is a function off.

Example 1 :VDEL 2.5 ms
Sets the delay time to 2.5ms.

Example 2 :VDEL 0.0025s
Sets the delay time to 2.5ms.

Query example :VDEL?
>0.0025
The delay time is 2.5ms.

Set →
 → Query

:CONFigure:RESPonse

| | | |
|------------------|---|------------------------|
| Description | Sets and queries for the response speed of the CC, CR and CP mode. | |
| Syntax | :CONFigure:RESPonse{<NR2> MINimum MAXimum} | |
| Query Syntax | :CONFigure:RESPonse? | |
| Parameter | <NR2> | 0.1, 0.2, 0.5, 1.0 |
| | MINimum | Minimum response speed |
| | MAXimum | Maximum response speed |
| Return parameter | Return the set value of the response speed, by the "<NR2>" string. | |
| Example | :CONF:RESP MAX Sets the response to the maximum of the CC, CR and CP mode. | |
| Query example | :CONF:RESP? >1.0 Response speed of the CC, CR and CP mode is 1.0. | |

Set →
 → Query

[:CONFigure]:CNTTime

| | | |
|---------------|--|---------------------------------|
| Description | Sets and queries for the Count Timer function. | |
| Syntax | [:CONFigure]:CNTTime {<Boolean> OFF ON } | |
| Query Syntax | [:CONFigure]:CNTTime? | |
| Parameter | OFF | Turns the Count Time timer off. |
| | ON | Turns the Count Time timer on |
| Example | :CNT ON Turns the Count Time timer on. | |
| Query example | CNT? >ON Count Time timer is turned on. | |

Set →
 → Query

| | | |
|----------------------------|---|-------------------------------------|
| [:CONFigure]:COTime | | |
| Description | Sets and queries for the load cutoff time. A cutoff time of 0 seconds is the equivalent of disabling the cutoff time. | |
| Syntax | [:CONFigure]:COTime {<NRf>[s] OFF MINimum MAXimum} | |
| Query Syntax | [:CONFigure]:COTime? | |
| Parameter | <NRf>[s] | Cut off time in seconds (1~3599999) |
| | OFF | Turns the cutoff time off. |
| | MINimum | Sets the cutoff time to the maximum |
| | MAXimum | Sets the cutoff time to the minimum |
| Return parameter | Return the set value of the Cut-off time, by the "{<NR1> OFF}" string. "OFF" is a function off. | |
| Example | :COT MAX Sets the cutoff time to the maximum. | |
| Query example | :COT? >500 The cutoff time is set to 500 seconds. | |

Set →
 → Query

| | | |
|----------------------------|---|-------------------------------------|
| [:CONFigure]:CRUnit | | |
| Description | Sets and queries for the CR mode setting units. | |
| Syntax | [:CONFigure]:CRUNit {OHM MHO} | |
| Query Syntax | [:CONFigure]:CRUNit? | |
| Parameter | OHM | Set the units to "Ω". |
| | MHO | Set the units to "mS" (conductance) |
| Example | :CRU OHM Sets the CR mode units to ohms. | |

Query example :CRU?
 >OHM
 The CR mode units are ohms.

Set →

→ Query

:CONFigure:DYNamic

Description Sets and queries for the setting conditions of dynamic mode. Setting conditions can select the duty cycle or the timer, the percentage or the value.

Syntax :CONFigure:DYNamic { VALue | PERCent | TIME | FDUTy }

Query Syntax :CONFigure:DYNamic?

| | | |
|------------------|---------|----------------------------|
| Parameter | VALue | Set the units to Value. |
| | PERCent | Set the units to Percent. |
| | TIME | Use timers for timing. |
| | FDUTy | Use duty cycle for timing. |

Return parameter Return the unit and the timing mode, by the "{ Value | Percent},{T1/T2 | Fre./Duty}" string.

Example :CONF:DYN VAL
 Sets the dynamic mode units to value.

Query example :CONF:DYN?
 > Value,T1,T2
 The dynamic mode becomes a value setup and timer setup.

Set →

→ Query

:CONFigure:MEMory

Description This command configures the how the files are recalled Local operation mode. By default when you try to recall a file or setting from memory, a message will appear asking you to press the Enter key to confirm each time you wish to recall. This command enables (SAFety) or disables this feature

(DIRect).

| | | |
|------------------|--|----------------------------------|
| Syntax | :CONFigure:MEMory {SAFety DIRect } | |
| Query Syntax | :CONFigure:MEMory? | |
| Parameter | SAFety | Safety setting. |
| | DIRect | Directly recall the chosen file. |
| Return parameter | Return the presence or absence of confirmation of recall, by the "{ Safety Direct }" string. | |
| Example | :CONF:MEM SAF Enables the safety setting. | |
| Query example | :CONF:MEM? >Safety The safety setting is enabled. | |

Set →

→ Query

:CONFigure:SHORT

| | | |
|------------------|--|--|
| Description | Sets and queries for the short key behavior. | |
| Syntax | :CONFigure:SHORT { TOGGle HOLD } | |
| Query Syntax | :CONFigure:SHORT? | |
| Parameter | HOLD | Sets the short key configuration to hold |
| | TOGGle | Sets the short key configuration to toggle |
| Return parameter | Return the short key Action, by the "{Toggle Hold}" string. | |
| Example | :CONF:SHOR TOGG Sets the short key configuration to toggle. | |
| Query example | :CONF:SHOR? >Toggle The short key is configured to toggle. | |

Set →
 → Query

:CONFigure:SHORT:SAFety

| | | |
|------------------|---|-----|
| Description | Turns the Short Safety function on/off. The short safety function requires the load to already be turned on before the load can be shorted using the Short key or :INPut:SHORT command. | |
| Syntax | :CONFigure:SHORT:SAFety {<bool> OFF ON} | |
| Query Syntax | :CONFigure:SHORT:SAFety? | |
| Parameter | OFF 0 | OFF |
| | ON 1 | ON |
| Return parameter | Return the short safety function on/off. | |
| Example | :CONF:SHOR:SAF OFF Sets the Short Safety off. | |
| Query example | :CONF:SHOR:SAF? >OFF Short safety is turned off. | |

Set →
 → Query

:CONFigure:SHORT:FUNction

| | | |
|------------------|---|-----------------------------|
| Description | Enables or disables the short function by short key. The load cannot be shorted until the short function has been enabled with this function. | |
| Syntax | CONFigure:SHORT:FUNction {<Boolean> OFF ON} | |
| Query Syntax | :CONFigure:SHORT:FUNction? | |
| Parameter | OFF 0 | Disables the short function |
| | ON 1 | Enables the short function |
| Return parameter | Return the short key/short function. | |
| Example | :CONF:SHOR:FUNC ON Enables the short function. | |

Query example :CONF:SHOR:FUNC?
>ON

Indicates that the short function is enabled.

Set →

← Query

[[:CONFigure]:GNG:MODE

Description Sets and queries for the entry mode for the Go-NoGo settings. The entry mode determines whether the Go-NoGo limits are set as values or as a percentage value from a center reference value.

Syntax [[:CONFigure]:GNG:MODE {PERCent | VALue }

Query Syntax [[:CONFigure]:GNG:MODE?

| | | |
|------------------|---------|------------------------------|
| Parameter | PERCent | Sets the entry mode to %. |
| | VALue | Sets the entry mode to value |

Return parameter Returns the Setting value of Go-NoGo input mode, by the "{Percent | Value}" string.

Example :GNG:MODE PERC
Sets the entry mode to %.

Query example :GNG:MODE?
>Percent
The entry mode is %.

Related Commands [[:CONFigure]:GNG:H
[[:CONFigure]:GNG:L
[[:CONFigure]:GNG:C

Set →

→ Query

[:CONFigure]:GNG:H

| | | |
|------------------|---|--|
| Description | Sets and queries for the high voltage/current limit value. If the entry mode is set to value, the high voltage/current limit value units are in volts/amps. If the entry mode is set to percent, the high voltage/current limit value units are in percent. | |
| Syntax | [:CONFigure]:GNG:H <NRf> | |
| Query Syntax | [:CONFigure]:GNG:H? | |
| Parameter | <NRf> | Sets the high voltage/current limit value in volts/amps or in percent. |
| Return parameter | Returns the voltage/current upper limit value, by the "{<NR2>}" string. | |
| Example | :GNG:H 100.0 Sets the high voltage limit value to 100%. | |
| Query example | :GNG:H? >100.0 Returns the high voltage limit value as 100.0%. | |
| Related Commands | [:CONFigure]:GNG:Mode [:CONFigure]:GNG:L | |

Set →

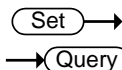
→ Query

[:CONFigure]:GNG:L

| | | |
|--------------|--|--|
| Description | Sets and queries for the low voltage/current limit value. If the entry mode is set to value, the low voltage/current limit value units are in volts/amps. If the entry mode is set to percent, the low voltage/current limit value units are in percent. | |
| Syntax | [:CONFigure]:GNG:L <NRf> | |
| Query Syntax | [:CONFigure]:GNG:L? | |

| | | |
|------------------|---|---|
| Parameter | <NRf> | Sets the low voltage/current limit value in volts/amps or in percent. |
| Return parameter | Returns the voltage/current lower limit value, by the "{<NR2>}" string. | |
| Example | :GNG:L 10.0 | Sets the low voltage limit value to 10%. |
| Query example | :GNG:L? >10.0 | Returns the low voltage limit value as 10.0%. |
| Related Commands | [:CONFigure]:GNG:Mode [:CONFigure]:GNG:H | |

[:CONFigure]:GNG:C



| | | |
|------------------|--|---|
| Description | Sets and queries for the center voltage/current limit value. The center voltage limit value is used as the center reference value when the entry mode is set to percent. | |
| Syntax | [:CONFigure]:GNG:C <NRf> | |
| Query Syntax | [:CONFigure]:GNG:C? | |
| Parameter | <NRf> | Sets the center voltage/current limit value in volts/amps. |
| Return parameter | Returns the Center value of voltage/current, by the "{<NR2>}" string. | |
| Example | :GNG:C 10.0 | Sets the center voltage/current limit value to 10V or 10A. |
| Query example | :GNG:C? >10.0 | Returns the center voltage/current limit value of 10V or 10A. |
| Related Commands | [:CONFigure]:GNG:Mode | |

Set →
 → Query

[:CONFigure]:GNG:DTIME

| | | |
|------------------|---|--|
| Description | Sets and queries for the Go-NoGo delay time. | |
| Syntax | [:CONFigure]:GNG:DTIME {<NRf>[s] MINimum MAXimum } | |
| Query Syntax | [:CONFigure]:GNG:DTIME? | |
| Parameter | <NRf>[s] | Sets the Go-NoGo delay time in seconds (0.0~1.0) with 0.1 second resolution. |
| | MINimum | Minimum delay time |
| | MAXimum | Maximum delay time |
| Return parameter | Returns the delay time in seconds, by the "<NR2>" string. | |
| Example | :GNG:DTIM 0.5 Sets the delay time to 0.5 seconds. | |
| Query example | :GNG:DTIM? >0.5 The delay time is 0.5 seconds. | |

Set →
 → Query

[:CONFigure]:GNG:SPECTest

| | | |
|---------------|--|-----------------|
| Description | Sets and queries for Go-NoGo testing | |
| Syntax | [:CONFigure]:GNG:SPECTest {<Boolean> OFF ON } | |
| Query Syntax | [:CONFigure]:GNG:SPECTest? | |
| Parameter | OFF | SPEC test = OFF |
| | ON | SPEC test = ON |
| Example | :GNG:SPECTest ON Turns Go-NoGo testing on. | |
| Query example | :GNG:SPECTest? >OFF Indicates that Go-NoGo testing is off. | |

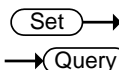
[:CONFigure]:GNG[:PASS]

→ Query

| | |
|------------------|---|
| Description | Queries the Go-NoGo test result(s). This command can be used for all test modes (CC, CV, CR, CP). |
| Query Syntax | [:CONFigure]:GNG[:PASS]? |
| Return parameter | NG No Go (fail) GO Go (Pass) |
| Query example | :GNG? >GO Returns the Go-NoGo test result. |

Parallel Command

[[:CONFigure]:PARAllel



| | | |
|------------------|--|------------------------------|
| Description | Configures the unit for parallel operation, or queries its state. This command configures the unit as a Master or Slave, and configures how many slave units are connected if the unit is configured as a master. | |
| Syntax | [:CONFigure]:PARAllel { MASTer SLAVe OFF P2 P3 P4 P5 B1 B2 B3 B4} | |
| Query Syntax | [:CONFigure]:PARAllel? | |
| Parameter | P2,P3,P4 or P5 | Number of connected slaves |
| | B1,B2,B3 or B4 | Number of connected Booster. |
| | OFF | Turn parallel mode off |
| | MASTer | Sets the unit to Master |
| | SLAVe | Sets the unit to Slave |
| Return parameter | Return the connections number and mode(master / slave). Response of master mode is the "Mode:Master,{Number:OFF Parallel Number:{2 3 4 5} Booster Number: {1 2 3 4} }". Response of Slave mode is the ""Mode:Slave". | |
| Example 1 | :PAR MAST Sets the unit to Master. | |
| Example 2 | :PAR B2 Configures the unit for use with 2 booster units. | |
| Query example | :PAR? >Mode:Master, Number:OFF The unit is set to Master and there are no connected slaves. | |

Step Resolution Commands

| | |
|------------------------------|----|
| :CONFigure:STATus | 65 |
| [[:CONFigure]:STEP:CC | 66 |
| [[:CONFigure]:STEP:CCH | 66 |
| [[:CONFigure]:STEP:CCM | 67 |
| [[:CONFigure]:STEP:CCL | 67 |
| [[:CONFigure]:STEP:CR | 68 |
| [[:CONFigure]:STEP:CRH | 68 |
| [[:CONFigure]:STEP:CRM | 69 |
| [[:CONFigure]:STEP:CRL | 70 |
| [[:CONFigure]:STEP:CV | 70 |
| [[:CONFigure]:STEP:CVH | 71 |
| [[:CONFigure]:STEP:CVL | 71 |
| [[:CONFigure]:STEP:CP | 72 |
| [[:CONFigure]:STEP:CPH | 72 |
| [[:CONFigure]:STEP:CPM | 73 |
| [[:CONFigure]:STEP:CPL | 74 |

Set →

→ Query

:CONFigure:STATus


| | | |
|------------------|---|--|
| Description | Sets the mode used for the set resolution when using the scroll wheel to edit parameters. | |
| Syntax | :CONFigure:STATus { STEP CURSOR } | |
| Query Syntax | :CONFigure:STATus? | |
| Parameter | STEP | Knob Status = Step (coarse/fine) |
| | CURSOR | Knob Status = Cursor |
| Return parameter | <ASCII string> | Returns the Knob Status configuration as a string. |
| Example | :CONF:STAT STEP Sets the mode to STEP. | |
| Query example | :CONF:STAT? >Step Returns the mode. | |

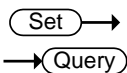
[:CONFigure]:STEP:CC → Query

| | |
|------------------|---|
| Description | Queries the step resolution for each CC Mode range. |
| Query Syntax | <code>[:CONFigure]:STEP:CC?</code> |
| Return parameter | Returns the step resolution for each CC Mode range, by the "CCH:<NR2>, CCM:<NR2>, CCL:<NR2>" string. Unit is [A] |
| Query example | <code>:STEP:CC?</code> <code>>CCH:0.0300, CCM:0.00300, CCL:0.000300</code> Returns the CC mode step resolution for each range. |

Set →

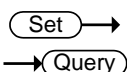
[:CONFigure]:STEP:CCH → Query

| | | |
|--|---|------------------------------|
| Description | Sets and queries for the step resolution for CC High Range. | |
|  Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. | |
| Syntax | <code>[:CONFigure]:STEP:CCH { <NRf>[A] MINimum MAXimum }</code> | |
| Query Syntax | <code>[:CONFigure]:STEP:CCH?</code> | |
| Parameter | <code><NRf>[A]</code> | Step resolution. Unit is [A] |
| | <code>MINimum</code> | Minimum step resolution |
| | <code>MAXimum</code> | Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CCH:<NR2>" string. | |
| Example | <code>:STEP:CCH 0.03A</code> Sets the step resolution to 0.03A. | |
| Query example | <code>:STEP:CCH?</code> <code>>CCH:0.0300</code> Returns the step resolution (0.03A). | |



[[:CONFigure]:STEP:CCM

| | | |
|------------------|---|------------------------------|
| Description | Sets and queries for the step resolution for CC medium Range. | |
| Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. | |
| Syntax | [:CONFigure]:STEP:CCM {<NRf>[A] MINimum MAXimum } | |
| Query Syntax | [:CONFigure]:STEP:CCM? | |
| Parameter | <NRf>[A] | Step resolution. Unit is [A] |
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CCM:<NR2>" string. | |
| Example | :STEP:CCM 0.003A Sets the step resolution to 0.003A. | |
| Query example | :STEP:CCM? >CCM:0.00300 Returns the step resolution (0.003A). | |



[[:CONFigure]:STEP:CCL

| | | |
|--------------|---|------------------------------|
| Description | Sets and queries for the step resolution for CC low Range. | |
| Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. | |
| Syntax | [:CONFigure]:STEP:CCL {<NRf>[A] MINimum MAXimum } | |
| Query Syntax | [:CONFigure]:STEP:CCL? | |
| Parameter | <NRf>[A] | Step resolution. Unit is [A] |


| | | |
|------------------|--|-------------------------|
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CCL:<NR2>" string. | |
| Example | :STEP:CCL 0.0003A Sets the step resolution to .0003A. | |
| Query example | :STEP:CCL? > CCL:0.000300 Returns the step resolution (0.0003A). | |

[[:CONFIgure]:STEP:CR → Query

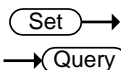
| | | |
|------------------|--|--|
| Description | Queries the step resolution for each CR Mode range. | |
| Query Syntax | [:CONFIgure]:STEP:CR? | |
| Return parameter | Returns the step resolution for each CR Mode range, by the " CRH:<NR2>, CRM:<NR2>, CRL:<NR2>" string. Unit is [mS] | |
| Query example | :STEP:CR? >CRH:3.00, CRM:0.300, CRL:0.0300 Returns the CR mode step resolution for each range. | |

Set →


[[:CONFIgure]:STEP:CRH → Query

| | | |
|--|---|-------------------------------|
| Description | Sets and queries for the step resolution for CR High Range. | |
|  Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. | |
| Syntax | [:CONFIgure]:STEP:CRH {<NRF>[mS] MINimum MAXimum } | |
| Query Syntax | [:CONFIgure]:STEP:CRH? | |
| Parameter | <NRF>[mS] | Step resolution. Unit is [mS] |
| | MINimum | Minimum step resolution |

| | |
|------------------|---|
| | MAXimum Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CRH:<NR2>" string. |
| Example | :STEP:CRH 3 Sets the step resolution to 3mS. |
| Query example | :STEP:CRH? >CRH:3.00 Returns the step resolution (3mS). |




[[:CONFigure]:STEP:CRM

| | |
|--|---|
| Description | Sets and queries for the step resolution for CR Medium Range. |
|  Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. |
| Syntax | [[:CONFigure]:STEP:CRM {<Nrf>[mS] MINimum MAXimum } |
| Query Syntax | [[:CONFigure]:STEP:CRM? |
| Parameter | <Nrf>[mS] Step resolution. Unit is [mS] MINimum Minimum step resolution MAXimum Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CRM:<NR2>" string. |
| Example | :STEP:CRM 0.3 Sets the step resolution to 0.3mS. |
| Query example | :STEP:CRM? >CRM:0.300 Returns the step resolution (0.3mS). |

Set →
 → Query

[[:CONFigure]:STEP:CRL

Description Sets and queries for the step resolution for CR Low Range.

 **Note** The step resolution setting will be automatically rounded to the closest multiple of the base resolution.

Syntax [[:CONFigure]:STEP:CRL {<NRf>[mS] | MINimum | MAXimum }

Query Syntax [[:CONFigure]:STEP:CRL?

| | | |
|------------------|------------------------|-------------------------------|
| Parameter | <NRf>[mS] | Step resolution. Unit is [mS] |
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |

Return parameter Returns the range and the step resolution, by the "CRL:<NR2>" string.

Example :STEP:CRL 0.03
 Sets the step resolution to 0.03mS.

Query example :STEP:CRL?
 >CRL:0.0300
 Returns the step resolution (0.03mS).

[[:CONFigure]:STEP:CV

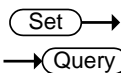
→ Query

Description Queries the step resolution for each CV Mode range.

Query Syntax [[:CONFigure]:STEP:CV?


Return parameter Returns the step resolution for each CV Mode range, by the "CVH:<NR2>, CVL:<NR2>" string. Unit is [V]

Query example :STEP:CV?
 >CVH:2.00, CVL:0.200
 Returns the CV mode step resolution for each range.



[[:CONFigure]:STEP:CVH

Description Sets and queries for the step resolution for CV High Range.

 **Note** The step resolution setting will be automatically rounded to the closest multiple of the base resolution.

Syntax [[:CONFigure]:STEP:CVH {<NRf>[V] | MINimum | MAXimum }

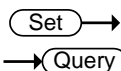
Query Syntax [[:CONFigure]:STEP:CVH?

| | | |
|------------------|----------|------------------------------|
| Parameter | <NRf>[V] | Step resolution. Unit is [V] |
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |

Return parameter Returns the range and the step resolution, by the "CVH:<NR2>" string.


Example :STEP:CVH 2V
Sets the step resolution to 2V.

Query example :STEP:CVH?
> CVH:2.00
Returns the step resolution (2V).



[[:CONFigure]:STEP:CVL

Description Sets and queries for the step resolution for CV Low Range.

 **Note** The step resolution setting will be automatically rounded to the closest multiple of the base resolution.

Syntax [[:CONFigure]:STEP:CVL {<NRf>[V] | MINimum | MAXimum }

Query Syntax [[:CONFigure]:STEP:CVL?

| | | |
|------------------|----------|------------------------------|
| Parameter | <NRf>[V] | Step resolution. Unit is [V] |
|------------------|----------|------------------------------|


| | | |
|------------------|---|-------------------------|
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CVL:<NR2>" string. | |
| Example | :STEP:CVL 0.2V Sets the step resolution to 0.2V. | |
| Query example | :STEP:CVL? > CVL:0.200 Returns the step resolution (0.2V). | |

[[:CONFIgure]:STEP:CP → **Query**

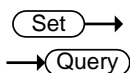
| | | |
|------------------|--|--|
| Description | Queries the step resolution for each CP Mode range. | |
| Query Syntax | [:CONFIgure]:STEP:CP? | |
| Return parameter | Returns the step resolution for each CP Mode range, by the "CPH:<NR2>, CPM:<NR2>, CPL:<NR2>" string. Unit is [W] | |
| Query example | :STEP:CP? > CPH:1.00, CPM:0.100, CPL:0.0100 Returns the CP mode step resolution for each range. | |

→ **Set** →


[[:CONFIgure]:STEP:CPH → **Query**

| | | |
|--|---|------------------------------|
| Description | Sets and queries for the step resolution for CP High Range. | |
|  Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. | |
| Syntax | [:CONFIgure]:STEP:CPH {<NRF>[W] MINimum MAXimum } | |
| Query Syntax | [:CONFIgure]:STEP:CPH? | |
| Parameter | <NRF>[W] | Step resolution. Unit is [W] |
| | MINimum | Minimum step resolution |

| | |
|------------------|---|
| | MAXimum Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CPH:<NR2>" string. |
| Example | :STEP:CPH 1 Sets the step resolution to 1W. |
| Query example | :STEP:CPH? >CPH:1.00 Returns the step resolution (1W). |



[[:CONFIgure]:STEP:CPM


| | |
|--|---|
| Description | Sets and queries for the step resolution for CP Medium Range. |
|  Note | The step resolution setting will be automatically rounded to the closest multiple of the base resolution. |
| Syntax | [[:CONFIgure]:STEP:CPM {<NRf>[W] MINimum MAXimum } |
| Query Syntax | [[:CONFIgure]:STEP:CPM? |
| Parameter | <NRf>[W] Step resolution. Unit is [W] MINimum Minimum step resolution MAXimum Maximum step resolution |
| Return parameter | Returns the range and the step resolution, by the "CPM:<NR2>" string. |
| Example | :STEP:CPM 0.1 Sets the step resolution to 0.1W. |
| Query example | :STEP:CPM? >CPM:0.100 Returns the step resolution (0.1W). |

Set →

→ Query

[[:CONFigure]:STEP:CPL

Description Sets and queries for the step resolution for CP Low Range.

 **Note** The step resolution setting will be automatically rounded to the closest multiple of the base resolution.

Syntax [[:CONFigure]:STEP:CPL {<NRf>[W] | MINimum | MAXimum }

Query Syntax [[:CONFigure]:STEP:CPL?

| | | |
|------------------|-----------------------|------------------------------|
| Parameter | <NRf>[W] | Step resolution. Unit is [W] |
| | MINimum | Minimum step resolution |
| | MAXimum | Maximum step resolution |

Return parameter Returns the range and the step resolution, by the "CPL:<NR2>" string.

Example :STEP:CPL 0.01
Sets the step resolution to 0.01W.

Query example :STEP:CPL?
>CPM:0.0100
Returns the step resolution (0.01W).


External Control Commands

| | |
|--------------------------------------|----|
| [:CONFigure]:EXTernal[:CONTRol]..... | 75 |
| [:CONFigure]:EXTernal:CV | 75 |
| [:CONFigure]:EXTernal:LOADonin..... | 76 |

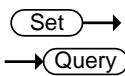
[:CONFigure]:EXTernal[:CONTRol]

| | | |
|------------------|--|---|
| Description | Sets and queries for the external control of CC, CR, CV, CP mode. | |
| Syntax | [:CONFi gure]:EXTernal[:CONTRol]{ OFF VOLTage RESistance RINV } | |
| Query Syntax | [:CONFi gure]:EXTernal[:CONTRol]? | |
| Parameter | OFF | Disables external control |
| | VOLTage | Sets the unit to external voltage control |
| | RESistance | Sets the unit to external resistance control |
| | RINV | Sets the unit to external resistance (inverted) control |
| Return parameter | Returns mode of the external control, by the "Control:{OFF Volt Res Rinverse}" string. | |
| Example | :EXT OFF Turns external control off. | |
| Query example | :EXT? >Control:OFF External control is setting is off. | |

[:CONFigure]:EXTernal:CV

| | | |
|--|---|--|
| Description | Sets and queries for the external control of +CV mode. | |
|  Note | Please set the unit to external control of CC, CR, CV, CP mode. | |

| | | |
|------------------|--|---|
| Syntax | [:CONFigure]:EXTernal:CV { OFF ON } | |
| Query Syntax | [:CONFigure]:EXTernal:CV? | |
| Parameter | OFF | Disables external control of +CV mode |
| | ON | Sets the unit to external control of +CV mode |
| Return parameter | Returns the external control of +CV mode configuration, by the "CV:{OFF ON}" string. | |
| Example | :EXT:CV ON Turns external control of +CV mode on. | |
| Query example | :EXT:CV? >CV:ON Uses external control of +CV mode. | |



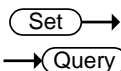
[:CONFigure]:EXTernal:LOADonin

| | | |
|------------------|---|--------------------|
| Description | Sets and queries for turned on and off with the external switch. External switch (LoadOn IN) is whether or not turned on the load setting of at the time of the open(HIGH) or closed(LOW). | |
| Syntax | [:CONFigure]:EXTernal:LOADonin {OFF High Low} | |
| Query Syntax | [:CONFigure]:EXTernal:LOADonin? | |
| Parameter | OFF | LoadOn IN = off |
| | HIGH | LoadOn IN = open |
| | LOW | LoadOn IN = closed |
| Return Parameter | Returns the setting value of external switch, by the "LoadOn In:{OFF High Low}" string. | |
| Example | :EXT:LOAD OFF Turns The LoadOn IN off. | |
| Query example | : EXT:LOAD? >LoadOn In:OFF The LoadOn In setting is off. | |

Mode Subsystem Commands

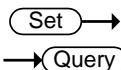
| | |
|------------------------|----|
| :MODE | 77 |
| [:MODE]:CRANge | 77 |
| [:MODE]:VRANge | 78 |
| [:MODE]:RESPonse | 78 |
| [:MODE]:DYNamic | 79 |

:MODE



| | | |
|---------------|--|--------------|
| Description | Sets and queries for the operating modes. | |
| Syntax | :MODE {CC CR CV CP CCCV CRCV CPCV} | |
| Query Syntax | :MODE? | |
| Parameter | CC | CC mode |
| | CR | CR mode |
| | CV | CV mode |
| | CP | CP mode |
| | CCCV | CC + CV mode |
| | CRCV | CR + CV mode |
| | CPCV | CP + CV mode |
| Example | :MODE CC Sets the mode to CC mode. | |
| Query example | :MODE? >CC Returns the operating mode (CC mode). | |

[:MODE]:CRANge



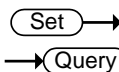
| | | |
|--------------|--|------------|
| Description | Sets and queries for the current range of all the operating modes. | |
| Syntax | [:MODE]:CRANge {HIGH MIDDLE LOW} | |
| Query Syntax | [:MODE]:CRANge? | |
| Parameter | HIGH | High range |

| | |
|--------|--------------|
| MIDDLE | Middle range |
| LOW | Low range |

Return parameter Returns the setting of Setting of the current range, by the "{High | Mid | Low}" string.

Example :CRAN LOW
Sets the current range to Low.

Query example :CRAN?
>Low
The current range is set to Low.



[[:MODE]:VRANge

Description Sets and queries for the voltage range of all operating modes.

Syntax [[:MODE]:VRANge { HIGH | LOW }

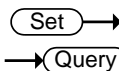
Query Syntax [[:MODE]:VRANge?

| | | |
|-----------|------|------------|
| Parameter | HIGH | High range |
| | LOW | Low range |

Return parameter Returns the setting of Setting of the voltage range, by the "{High | Low}" string.

Example :VRAN LOW
Sets the voltage range to Low.

Query example :VRAN?
>Low
The voltage range is set to Low.



[[:MODE]:RESPonse

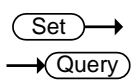
Description Sets and queries for The response speed of the CV and +CV mode. The default is fast response.

Syntax [[:MODE]:RESPonse { FAST | RESP{ 6 | 5 | 4 | 3 | 2 | 1 } | SLOW }

Query Syntax [[:MODE]:RESPonse?

| | | |
|------------------|--|---|
| Parameter / | FAST RESP6 | Response speed |
| Return parameter | RESP5 RESP4 RESP3 RESP2 RESP1 SLOW | |
| Example | :RESP FAST | Sets the CV and +CV response to fast. |
| Query example | :RESP? >FAST | The speed response of CV and +CV mode is set to fast. |

[[:MODE]:DYNamic

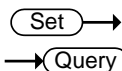


| | | |
|------------------|--|--|
| Description | Sets and queries for the Switching function. | |
| Syntax | [:MODE]:DYNamic { DYNamic STATic } | |
| Query Syntax | [:MODE]:DYNamic? | |
| Parameter | DYNamic | Set to Dynamic mode |
| | STATic | Set to Static mode |
| Return parameter | Returns the Setting of Switching function, by the "{Dynamic Static}" string. | |
| Example | :DYN DYN | Set the switching function to dynamic |
| Query example | :DYN? >Dynamic | The switching function is set to dynamic mode. |

Current Subsystem Commands

| | |
|-------------------------|----|
| :CURRent[:VA] | 80 |
| :CURRent[:VA]:TRIGgered | 81 |
| :CURRent:VB | 81 |
| :CURRent:SRATe | 82 |
| :CURRent:L1 | 82 |
| :CURRent:L2 | 83 |
| :CURRent:SET | 84 |
| :CURRent:LEVel | 84 |
| :CURRent:RISE | 85 |
| :CURRent:FALL | 86 |
| :CURRent:T1 | 86 |
| :CURRent:T2 | 87 |
| :CURRent:FREQuency | 88 |
| :CURRent:DUTY | 88 |
| :CURRent:RECall | 89 |

:CURRent[:VA]



Description Sets and queries for the “A Value” current of the CC static mode.



Note

A different current value can be set for each current range (High/Mid/Low).

The [:VA] node can only be omitted when in static mode.

Syntax :CURRent[:VA] {<NRf>[A] | MINimum | MAXimum }

Query Syntax :CURRent[:VA]?

| | | |
|------------------|----------|-------------------------|
| Parameter | <NRf>[A] | “A Value” current value |
| | MINimum | Minimum current level |
| | MAXimum | Maximum current level |

Return parameter Return the current value of “A Value”, by the “<NR2>” string.

Example :CURR MIN
Sets the current value to the minimum.

Query example :CURR?
>1.0000
Current setting of "A Value" is set to 1A.

:CURRent[:VA]:TRIGgered (Set) →

Description Set the current value when the trigger is activated.

Syntax :CURRent[:VA]:TRIGgered {<NR2>[A] | MINimum | MAXimum }


| | | |
|-----------|----------|-------------------------|
| Parameter | <NRf>[A] | "A Value" current value |
| | MINimum | Minimum current level |
| | MAXimum | Maximum current level |

Example :CURR:TRIG MIN
Set the minimum current value when the trigger is activated.

Related Commands *TRG, :INITiate:CONTInuous, :INITiate[:IMMEDIATE]

:CURRent:VB (Set) →
→ (Query)

Description Sets and queries for the "B Value" current of the CC static mode.

 Note A different current value can be set for each current range (High/Mid/Low).

Syntax :CURRent:VB {<NRf>[A] | MINimum | MAXimum }

Query Syntax :CURRent:VB?

| | | |
|-----------|----------|-------------------------|
| Parameter | <NRf>[A] | "B Value" current value |
| | MINimum | Minimum current level |
| | MAXimum | Maximum current level |

Return parameter Return the Current value of "B Value", by the "<NR2>" string.

Example :CURR:VB MIN
Sets the current value to the minimum.

Query example :CURR:VB?
 >1.0000
 Current setting of "B Value" is set to 1A.

Set →

:CURRent:SRATe

→ Query

Description Sets and queries for the current slew rate of CC static mode.

Syntax :CURRent:SRATe {<NRf> | MINimum | MAXimum }

Query Syntax :CURRent:SRATe?

| | | |
|-----------|---------|------------------------------|
| Parameter | <NRf> | Sets the slew rate in mA/us |
| | MINimum | Set to the lowest slew rate |
| | MAXimum | Set to the highest slew rate |

Return parameter Return the slew rate, by the "<NR2>" string.

Example :CURR:SRAT MIN
 Sets the slew rate to the minimum.


Query example :CURR:SRAT?
 >5.0000
 The slew rate is set to 5mA/us.

Set →

:CURRent:L1

→ Query


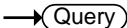
Description Sets and queries for the "Level1" current of the CC dynamic mode.
 This command applies in "Dyna.Level" is "Value".


 Note A different current value can be set for each range (High/Mid/Low).

Syntax :CURRent:L1 {<NRf>[A] | MINimum | MAXimum }

Query Syntax :CURRent:L1?

| | | |
|-----------|----------|------------------------|
| Parameter | <NRf>[A] | "Level1" current value |
| | MINimum | Minimum current level |

| | |
|--|--|
| | MAXimum Maximum current level |
| Return parameter | Return the current value of "Level1", by the "<NR2>" string. |
| Example | :CURR:L1 MIN Sets the current value to the minimum. |
| Query example | :CURR:L1? >1.0000 Current setting of " Level1" is set to 1A. |
|   | |
| :CURRent:L2 | |

| | |
|--|---|
| Description | Sets and queries for the "Level2" current of the CC dynamic mode. This command applies in "Dyna.Level" is "Value". |
|  Note | A different current value can be set for each range (High/Mid/Low). |
| Syntax | :CURRent:L2 {<NRf>[A] MINimum MAXimum } |
| Query Syntax | :CURRent:L2? |
| Parameter | <NRf>[A] "Level2" current value MINimum Minimum current level MAXimum Maximum current level |
| Return parameter | Return the current value of "Level2", by the "<NR2>" string. |
| Example | :CURR:L2 MIN Sets the current value to the minimum. |
| Query example | :CURR:L2? >1.0000 Current setting of " Level2" is set to 1A. |

:CURRent:SET




Description Sets and queries for the "Set" current of the CC dynamic mode.
 This command applies in "Dyna.Level" is "Percent".

Syntax :CURRent:SET{<NRf>[A] | MINimum | MAXimum }

Query Syntax :CURRent:SET?

| | | |
|------------------|----------|---|
| Parameter | <NRf>[A] | The current value at the time of "Level=100%" |
| | MINimum | Minimum current value |
| | MAXimum | Maximum current value |

Return parameter Return the current value of "Level = 100%", by the "<NR2>" string.

Example :CURR:SET MIN
 Sets the minimum current value of "Level = 100%".

Query example :CURR:SET?
 >1.0
 Current value of "Level = 100%" is set to 1A.

Related Commands :CURRent:LEVel

:CURRent:LEVel




Description Sets and queries for the "Level" % of the CC dynamic mode.
 This command applies in "Dyna.Level" is "Percent".

Syntax :CURRent:LEVel {<NRf> | MINimum | MAXimum }

Query Syntax :CURRent:LEVel?

| | | |
|------------------|---------|--|
| Parameter | <NRf> | % of "SET" current level (unit is [%]) |
| | MINimum | Minimum % of "SET" current level |

| | |
|------------------|---|
| | MAXimum Maximum % of "SET" current level |
| Return parameter | Return the current of "% Level", by the "<NR2>" string. |
| Example | :CURR:LEV MIN Sets the % level current value to the minimum. |
| Query example | :CURR:LEV? >50 Percentage of the set current value is set to 50%. |
| Related Commands | :CURRent:SET |



| | | | | | | | |
|------------------|--|-------|---|---------|-------------------|---------|-------------------|
| Description | Sets and queries for the rising current slew rate of the CC dynamic mode. | | | | | | |
| Syntax | :CURRent:RISE {<NRf> MINimum MAXimum } | | | | | | |
| Query Syntax | :CURRent:RISE? | | | | | | |
| Parameter | <table border="1"> <tr> <td><NRf></td> <td>Rising current slew rate(unit is [mA/us])</td> </tr> <tr> <td>MINimum</td> <td>Minimum slew rate</td> </tr> <tr> <td>MAXimum</td> <td>Maximum slew rate</td> </tr> </table> | <NRf> | Rising current slew rate(unit is [mA/us]) | MINimum | Minimum slew rate | MAXimum | Maximum slew rate |
| <NRf> | Rising current slew rate(unit is [mA/us]) | | | | | | |
| MINimum | Minimum slew rate | | | | | | |
| MAXimum | Maximum slew rate | | | | | | |
| Return parameter | Return the rise of the current slew rate, by the "<NR2>" string. | | | | | | |
| Example | :CURR:RISE MIN Sets the rising slew rate to the minimum. | | | | | | |
| Query example | :CURR:RISE? >5000 Returns the rising slew rate as 5000mA/us. | | | | | | |
| Related Commands | :CURRent:FALL | | | | | | |

Set →

→ Query

:CURRent:FALL

Description Sets and queries for the falling of the current slew rate of the CC dynamic mode.

Syntax :CURRent:FALL {<NRf> | MINimum | MAXimum }

Query Syntax :CURRent:FALL?

| | | |
|------------------|--------------------|---|
| Parameter | <NRf> | Falling current slew rate (unit is [mA/us]) |
| | MINimum | Minimum slew rate |
| | MAXimum | Maximum slew rate |

Return parameter Return the falling of the current slew rate, by the "<NR2>" string.

Example :CURR:FALL MIN
Sets the falling slew rate to the minimum.

Query example :CURR:FALL?
>5000
Returns the falling slew rate as 5000mA/us.

Related Commands :CURRent: RISE

Set →

→ Query

:CURRent:T1

Description Sets and queries for the "Timer1" time of CC dynamic mode.
This command applies in "Dyna.Time" is "T1/T2".

Syntax :CURRent:T1 {<NRf>[s] | MINimum | MAXimum }

Query Syntax :CURRent:T1?

| | | |
|------------------|-----------------------|-------------------------------------|
| Parameter | <NRf>[s] | T1 timer setting. (unit is seconds) |
| | MINimum | Minimum time |
| | MAXimum | Maximum time |

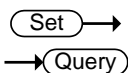
Return parameter Return the setting of the timer T1, by the "<NR2>" string.

Example :CURR:T1 0.2
Sets the setting of the timer T1.

Query example :CURR:T1?
>0.2
Return the setting of the timer T1.

Related Commands :CURRent:T2

:CURRent:T2



Description Sets and queries for the "Timer2" time of CC dynamic mode.
This command applies in "Dyna.Time" is "T1/T2".

Syntax :CURRent:T2 {<NRf>[s] | MINimum | MAXimum }

Query Syntax :CURRent:T2?

| | | |
|-----------|----------|-------------------------------------|
| Parameter | <NRf>[s] | T2 timer setting. (unit is seconds) |
| | MINimum | Minimum time |
| | MAXimum | Maximum time |

Return parameter Return the setting of the timer T2, by the "<NR2>" string.

Example :CURR:T2 0.2
Sets the setting of the timer T2.

Query example :CURR:T2?
>0.2
Returns the setting of the timer T2.

Related Commands :CURRent:T1

Set →
 → Query

:CURRent:FREQuency

Description Sets and queries for “Frequency” value of the CC dynamic mode.
 This command applies in “Dyna.Time” is “Freq/Duty”.

Syntax :CURRent:FREQuency {<NRf> | MINimum | MAXimum }

Query Syntax :CURRent:FREQuency?

| | | |
|------------------|--------------------|---|
| Parameter | <NRf> | Sets the switching frequency.(unit is Hz) |
| | MINimum | Minimum frequency |
| | MAXimum | Maximum frequency |

Return parameter Return the switching frequency, by the "<NR2>" string.

Example :CURR:FREQ 60
 Sets frequency to 60Hz.

Query example :CURR:FREQ?
 >60
 Returns the switching frequency as 60Hz.

Related Commands :CURRent:DUTY

Set →
 → Query

:CURRent:DUTY

Description Sets and queries for “Duty” % of the CC dynamic mode.
 This command applies in “Dyna.Time” is “Freq/Duty”.

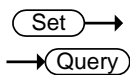
Syntax :CURRent:DUTY {<NRf> | MINimum | MAXimum }

Query Syntax :CURRent:DUTY?

| | | |
|------------------|--------------------|--------------------------------------|
| Parameter | <NRf> | Sets the duty cycle as a percentage. |
|------------------|--------------------|--------------------------------------|

| | | |
|------------------|---|--------------------|
| | MINimum | Minimum duty cycle |
| | MAXimum | Maximum duty cycle |
| Return parameter | Return the duty cycle of positive, by the "<NR2>" string. | |
| Example | :CURR:DUTY 50 Sets the duty cycle 50%. | |
| Query example | :CURR:DUTY? >50 Returns the duty cycle as 50%. | |
| Related Commands | :CURRent:FREQuency | |

:CURRent:RECall

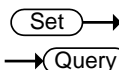


| | | |
|------------------|---|-----------------------------|
| Description | Sets or queries whether A Value or B Value is the currently active value in CC static mode. | |
| Syntax | :CURRent:RECall {A 0 B 1} | |
| Query Syntax | :CURRent:RECall? | |
| Parameter | A or 0 | CC active setting = A value |
| | B or 1 | CC active setting = B value |
| Return parameter | 0 | CC active setting = A value |
| | 1 | CC active setting = B value |
| Example | :CURR:REC A Sets A value of CC setting mode to active. | |
| Query example | :CURR:REC? >0 Return the CC active setting. | |

Resistance Subsystem Commands

| | |
|-----------------------------|-----|
| :RESistance[:VA] | 90 |
| :RESistance[:VA]:TRIGgered | 91 |
| :RESistance:VB | 91 |
| :RESistance:SRATe | 92 |
| :RESistance:L1 | 93 |
| :RESistance:L2 | 93 |
| :RESistance:SET | 94 |
| :RESistance:LEVel | 95 |
| :RESistance:RISE | 95 |
| :RESistance:FALL | 96 |
| :RESistance:T1 | 97 |
| :RESistance:T2 | 97 |
| :RESistance:FREQuency | 98 |
| :RESistance:DUTY | 99 |
| :CONDuctance[:VA] | 99 |
| :CONDuctance[:VA]:TRIGgered | 100 |
| :CONDuctance:VB | 100 |
| :CONDuctance:L1 | 101 |
| :CONDuctance:L2 | 102 |
| :CONDuctance:SET | 103 |
| :CONDuctance:RECall | 103 |
| :RESistance:RECall | 104 |

:RESistance[:VA]



Description Sets and queries for the “A Value” resistance of the CR static mode.



Note

A different value can be set for each current range (High/Mid/Low).

The optional command node [:VA] can only be omitted when in static mode.

Syntax :RESistance[:VA] {<NRF>[OHM] | MINimum | MAXimum }

Query Syntax :RESistance[:VA]?


Parameter <NRF>[OHM] “A Value” resistance value


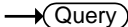
| | | |
|------------------|---|--------------------------|
| | MINimum | Minimum resistance level |
| | MAXimum | Maximum resistance level |
| Return parameter | Return the resistance value of "A Value", by the "<NR2>" string. | |
| Example | :RES:VA MIN Sets the resistance value to the minimum. | |
| Query example | :RES:VA? >9.840 Return the resistance value(Ω) of "A Value". | |

:RESistance[:VA]:TRIGgered (Set) →

| | | |
|------------------|--|----------------------------|
| Description | Set the resistance value when the trigger is activated. | |
| Syntax | :RESistance[:VA]:TRIGgered {<NRf>[OHM] MINimum MAXimum } | |
| Parameter | <NRf>[OHM] | "A Value" resistance value |
| | MINimum | Minimum resistance level |
| | MAXimum | Maximum resistance level |
| Example | :RES:TRIG MIN Set the minimum resistance value when the trigger is activated. | |
| Related Commands | *TRG, :INITiate:CONTinuous, :INITiate[:IMMediate] | |

:RESistance:VB (Set) →
→ (Query)

| | | |
|--|--|--|
| Description | Sets and queries for the "B Value" resistance of the CR static mode. | |
|  Note | A different value can be set for each current range (High/Mid/Low). | |
| Syntax | :RESistance:VB {<NRf>[OHM] MINimum MAXimum } | |


| | |
|---|---|
| Query Syntax | :RESistance:VB? |
| Parameter | <NRf>[OHM] "B Value" resistance value MINimum Minimum resistance level MAXimum Maximum resistance level |
| Return parameter | Return the resistance value of "B Value", by the "<NR2>" string. |
| Example | :RES:VB MIN Sets the resistance value to the minimum. |
| Query example | :RES:VB? >9.840 Return the resistance value(Ω) of "B Value". |
|  | |
|  | |

:RESistance:SRATe

| | |
|------------------|--|
| Description | Sets and queries for the current slew rate of CR static mode. |
| Syntax | :RESistance:SRATe {<NRf> MINimum MAXimum } |
| Query Syntax | :RESistance:SRATe? |
| Parameter | <NRf> Sets the slew rate in mA/us MINimum Set to the lowest slew rate MAXimum Set to the highest slew rate |
| Return parameter | Return the slew rate, by the "<NR2>" string. |
| Example | :RES:SRAT MIN Sets the slew rate to the minimum. |
| Query example | :RES:SRAT? >5.0000 Return the slew rate. |


:RESistance:L1



| | | | | | | | |
|--|--|------------|---------------------------|---------|--------------------------|---------|--------------------------|
| Description | <p>Sets and queries for the "Level1" resistance of the CR dynamic mode.</p> <p>This command applies in "Dyna.Level" is "Value".</p> | | | | | | |
|  Note | A different value can be set for each current range (High/Mid/Low). | | | | | | |
| Syntax | :RESistance:L1 {<NRf>[OHM] MINimum MAXimum } | | | | | | |
| Query Syntax | :RESistance:L1? | | | | | | |
| Parameter | <table border="0"> <tr> <td style="background-color: #e0e0e0;"><NRf>[OHM]</td> <td>"level1" resistance value</td> </tr> <tr> <td style="background-color: #e0e0e0;">MINimum</td> <td>Minimum resistance level</td> </tr> <tr> <td style="background-color: #e0e0e0;">MAXimum</td> <td>Maximum resistance level</td> </tr> </table> | <NRf>[OHM] | "level1" resistance value | MINimum | Minimum resistance level | MAXimum | Maximum resistance level |
| <NRf>[OHM] | "level1" resistance value | | | | | | |
| MINimum | Minimum resistance level | | | | | | |
| MAXimum | Maximum resistance level | | | | | | |
| Return parameter | Return the resistance value of "Level1", by the "<NR2>" string. | | | | | | |
| Example | <p>:RES:L1 MIN</p> <p>Sets the resistance value to the minimum.</p> | | | | | | |
| Query example | <p>:RES:L1?</p> <p>>9.840</p> <p>Return the resistance value(Ω) of "Level1".</p> | | | | | | |

:RESistance:L2



| | |
|--|---|
| Description | <p>Sets and queries for the "Level2" resistance of the CR dynamic mode.</p> <p>This command applies in "Dyna.Level" is "Value".</p> |
|  Note | A different value can be set for each current range (High/Mid/Low). |
| Syntax | :RESistance:L2 {<NRf>[OHM] MINimum MAXimum } |

| | | |
|------------------|--|---------------------------|
| Query Syntax | :RESistance:L2? | |
| Parameter | <NRf>[OHM] | “level2” resistance value |
| | MINimum | Minimum resistance level |
| | MAXimum | Maximum resistance level |
| Return parameter | Return the resistance value of "Level2", by the "<NR2>" string. | |
| Example | :RES:L2 MIN Sets the resistance value to the minimum. | |
| Query example | :RES:L2? >9.840 Return the resistance value (Ω) of "Level2". | |

Set →
 → Query

:RESistance:SET

| | | |
|------------------|---|---|
| Description | Sets and queries for the “Set” resistance of the CR dynamic mode. This command applies in “Dyna.Level” is “Percent”. | |
| Syntax | :RESistance:SET {<NRf>[OHM] MINimum MAXimum } | |
| Query Syntax | :RESistance:SET? | |
| Parameter | <NRf>[OHM] | “The resistance value at the time of "Level=100%" |
| | MINimum | Minimum resistance value |
| | MAXimum | Maximum resistance value |
| Return parameter | Return the resistance value of "Level = 100%", by the "<NR2>" string. | |
| Example | :RES:SET MIN Sets the minimum resistance value of "Level = 100%". | |

Query example :RES:SET?
>9.840

Return the resistance value (Ω) of "Level = 100%".

Set →

:RESistance:LEVel

→ Query

Description Sets and queries for the "Level" % (percentage of the Set conductance value) of the CR dynamic mode.

This command applies in "Dyna.Level" is "Percent".

Syntax :RESistance:LEVel {<NRf> | MINimum | MAXimum }

Query Syntax :RESistance:LEVel?

| | | |
|------------------|---------|--------------------------------------|
| Parameter | <NRf> | % of "Set" value level (unit is %) |
| | MINimum | Minimum % of "Set" conductance level |
| | MAXimum | Maximum % of "Set" conductance level |

Return parameter Return the Millisiemens of "% Level", by the "<NR2>" string.

Example :RES:LEV MIN
Sets the % level Millisiemens value to the minimum.

Query example :RES:LEV?
>50
Return the Millisiemens of "50% Level".

Related Commands :RESistance:SET

Set →

:RESistance:RISE

→ Query

Description Sets and queries for the rising current slew rate of the CR dynamic mode.

Syntax :RESistance:RISE {<NRf> | MINimum | MAXimum }

Query Syntax :RESistance:RISE?

| | | |
|------------------|-------|---|
| Parameter | <NRf> | Rising current slew rate(unit is [mA/us]) |
|------------------|-------|---|

| | |
|---------|-------------------|
| MINimum | Minimum slew rate |
| MAXimum | Maximum slew rate |

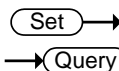
Return parameter Return the rise of the slew rate, by the "<NR2>" string.

Example :RES:RISE MIN
Sets the rising slew rate to the minimum.

Query example :RES:RISE?
>50.000
Return the rise of the slew rate.

Related :RESistance:FALL
Commands

:RESistance:FALL



Description Sets and queries for the falling current slew rate of the CR dynamic mode.

Syntax :RESistance:FALL {<NRf> | MINimum | MAXimum }

Query Syntax :RESistance:FALL?

| | | |
|-----------|---------|-------------------------------------|
| Parameter | <NRf> | Falling slew rate (unit is [mA/us]) |
| | MINimum | Minimum slew rate |
| | MAXimum | Maximum slew rate |

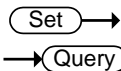
Return parameter Return the falling of the conductance slew rate, by the "<NR2>" string.

Example :RES:FALL MIN
Sets the falling slew rate to the minimum.

Query example :RES:FALL?
>50.000
Return the falling of the slew rate.

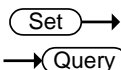
Related :RESistance:RISE
Commands

:RESistance:T1



| | | | | | | | |
|------------------|---|----------|-------------------------------------|---------|--------------|---------|--------------|
| Description | Sets and queries for the “Timer1” time of CR dynamic mode. This command applies in “Dyna.Time” is “T1/T2”. | | | | | | |
| Syntax | :RESistance:T1 {<NRf>[s] MINimum MAXimum } | | | | | | |
| Query Syntax | :RESistance:T1? | | | | | | |
| Parameter | <table border="1"> <tr> <td><NRf>[s]</td> <td>T1 timer setting. (unit is seconds)</td> </tr> <tr> <td>MINimum</td> <td>Minimum time</td> </tr> <tr> <td>MAXimum</td> <td>Maximum time</td> </tr> </table> | <NRf>[s] | T1 timer setting. (unit is seconds) | MINimum | Minimum time | MAXimum | Maximum time |
| <NRf>[s] | T1 timer setting. (unit is seconds) | | | | | | |
| MINimum | Minimum time | | | | | | |
| MAXimum | Maximum time | | | | | | |
| Return parameter | Return the setting of the timer T1, by the "<NR2>" string. | | | | | | |
| Example | :RES:T1 0.2 Sets the setting of the timer T1. | | | | | | |
| Query example | : RES:T1? >0.2 Return the setting of the timer T1. | | | | | | |
| Related Commands | :RESistance:T2 | | | | | | |

:RESistance:T2



| | | | | | |
|--------------|---|----------|-------------------------------------|---------|--------------|
| Description | Sets and queries for the “Timer2” time of CR dynamic mode. This command applies in “Dyna.Time” is “T1/T2”. | | | | |
| Syntax | :RESistance:T2 {<NRf>[s] MINimum MAXimum } | | | | |
| Query Syntax | :RESistance:T2? | | | | |
| Parameter | <table border="1"> <tr> <td><NRf>[s]</td> <td>T2 timer setting. (unit is seconds)</td> </tr> <tr> <td>MINimum</td> <td>Minimum time</td> </tr> </table> | <NRf>[s] | T2 timer setting. (unit is seconds) | MINimum | Minimum time |
| <NRf>[s] | T2 timer setting. (unit is seconds) | | | | |
| MINimum | Minimum time | | | | |

| | |
|------------------|--|
| | MAXimum Maximum time |
| Return parameter | Return the setting of the timer T2, by the "<NR2>" string. |
| Example | :RES:T2 0.2 Sets the setting of the timer T2. |
| Query example | :RES:T2? >0.2 Return the setting of the timer T2. |
| Related Commands | :RESistance:T1 |

:RESistance:FREQUENCY



| | | |
|------------------|---|---|
| Description | Sets and queries for "Frequency" value of the CR dynamic mode. This command applies in "Dyna.Time" is "Freq/Duty". | |
| Syntax | :RESistance:FREQUENCY {<NRf> MINimum MAXimum } | |
| Query Syntax | :RESistance:FREQUENCY? | |
| Parameter | <NRf> | Sets the switching frequency. (unit is hertz) |
| | MINimum | Minimum frequency |
| | MAXimum | Maximum frequency |
| Return parameter | Return the switching frequency, by the "<NR2>" string. | |
| Example | :RES:FREQ 60 Sets frequency to 60Hz. | |
| Query example | :RES:FREQ? >60 Returns the switching frequency as 60Hz. | |
| Related Commands | :RESistance:DUTY | |

Set →

→ Query


:RESistance:DUTY

| | | |
|------------------|--|--------------------------------|
| Description | Sets and queries for “Duty” % of the CR dynamic mode. This command applies in “Dyna.Time” is “Freq/Duty”. | |
| Syntax | :RESistance:DUTY {<NRf> MINimum MAXimum } | |
| Query Syntax | :RESistance:DUTY? | |
| Parameter | <NRf> | Sets the duty as a percentage. |
| | MINimum | Minimum duty |
| | MAXimum | Maximum duty |
| Return parameter | Return the duty cycle of positive, by the "<NR2>" string. | |
| Example | :RES:DUTY 50 Sets the duty cycle 50%. | |
| Query example | :RES:DUTY? >50 Returns the duty cycle as 50%. | |
| Related Commands | :RESistance:FREQuency | |

Set →

→ Query

:CONDuctance[:VA]

| | | |
|--|---|--|
| Description | Sets and queries for the “A Value” conductance of the CR static mode. | |
|  Note | A different value can be set for each current range (High/Mid/Low). The optional command node [:VA] can only be omitted when in static mode. | |
| Syntax | :CONDuctance [:VA] {<NRf>[mS] MINimum MAXimum } | |
| Query Syntax | :RESistance[:VA]? | |

| | | |
|------------------|--|---|
| Parameter | <NRf>[mS] | "A Value" conductance value. Unit is [mS] |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |
| Return parameter | Return the conductance value of "A Value", by the "<NR2>" string. | |
| Example | :COND:VA MIN Sets the conductance value to the minimum. | |
| Query example | :COND:VA? >9.840 Return the conductance value (mS) of "A Value". | |

:CONDuctance[:VA]:TRIGgered (Set) →

| | | |
|------------------|--|---|
| Description | Set the conductance value when the trigger is activated. | |
| Syntax | :CONDuctance[:VA]:TRIGgered {<NRf>[mS] MINimum MAXimum } | |
| Parameter | <NRf>[mS] | "A Value" conductance value. Unit is mS |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |
| Example | :COND:TRIG MIN Set the minimum conductance value when the trigger is activated. | |
| Related Commands | *TRG, :INITiate:CONTinuous, :INITiate[:IMMediate] | |

:CONDuctance:VB (Set) →
→ (Query)

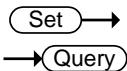
| | |
|-------------|---|
| Description | Sets and queries for the "B Value" conductance of the CR static mode. |
|-------------|---|




Note

A different value can be set for each current range (High/Mid/Low).

| | | |
|------------------|--|---|
| Syntax | :CONDuctance:VB {<NRf>[mS] MINimum MAXimum } | |
| Query Syntax | :CONDuctance:VB? | |
| Parameter | <NRf>[mS] | "B Value" conductance value. Unit is [mS] |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |
| Return parameter | Return the conductance value of "B Value", by the "<NR2>" string. | |
| Example | :COND:VB MIN Sets the conductance value to the minimum. | |
| Query example | :COND:VB? >9.840 Return the conductance value ([mS]) of "B Value". | |



:CONDuctance:L1

| | | |
|---|---|--|
| Description | Sets and queries for the "Level1" conductance of the CR dynamic mode. This command applies in "Dyna.Level" is "Value". | |
|  Note | A different value can be set for each current range (High/Mid/Low). | |
| Syntax | :CONDuctance:L1 {<NRf>[mS] MINimum MAXimum } | |
| Query Syntax | :CONDuctance:L1? | |
| Parameter | <NRf>[mS] | "level1" conductance value. Unit is [mS] |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |
| Return parameter | Return the conductance value of "Level1", by the "<NR2>" string. | |


Example :COND:L1 MIN
Sets the conductance value to the minimum.

Query example :COND:L1?
>9.840
Return the conductance value ([mS]) of "Level1".

Set →

:CONDuctance:L2 → Query

Description Sets and queries for the "Level2" conductance of the CR dynamic mode.
This command applies in "Dyna.Level" is "Value".

 Note A different value can be set for each current range (High/Mid/Low).

Syntax :CONDuctance:L2 {<NRf>[mS] | MINimum | MAXimum }

Query Syntax :CONDuctance:L2?

| | | |
|-----------|-----------|--|
| Parameter | <NRf>[mS] | "level2" conductance value. Unit is [mS] |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |

Return parameter Return the Millisiemens value of "Level2", by the "<NR2>" string.

Example :COND:L2 MIN
Sets the conductance value to the minimum.

Query example :COND:L2?
>9.840
Return the conductance value ([mS]) of "Level2".

:CONDuctance:SET




| | | |
|------------------|--|--|
| Description | Sets and queries for the “Set” conductance of the CR dynamic mode. This command applies in “Dyna.Level” is “Percent”. | |
| Syntax | :CONDuctance:SET {<NRf>[mS] MINimum MAXimum } | |
| Query Syntax | :CONDuctance:SET? | |
| Parameter | <NRf>[mS] | “The conductance value at the time of "Level=100%". Unit is [mS] |
| | MINimum | Minimum conductance level |
| | MAXimum | Maximum conductance level |
| Return parameter | Return the conductance value of "Level = 100%", by the "<NR2>" string. | |
| Example | :COND:SET MIN Sets the minimum conductance value of "Level = 100%". | |
| Query example | :COND:SET? >9.840 Return the conductance value ([mS]) of "Level = 100%". | |

:CONDuctance:RECall




| | | |
|--------------|---|--------------------|
| Description | Sets or queries whether A Value or B Value is the currently active value in CR static mode when the units are set to Siemens. | |
| Syntax | :CONDuctance:RECall {A 0 B 1} | |
| Query Syntax | :CONDuctance:RECall? | |
| Parameter | A or 0 | CR Value = A Value |
| | B or 1 | CR Value = B Value |

| | | |
|------------------|---|--------------------|
| Return parameter | 0 | CR Value = A Value |
| | 1 | CR Value = B Value |

Example :COND:REC A
 Sets A value of CR conductance setting mode to active.

Query example :RES:REC?
 >0
 Return the CR conductance mode active setting.

:RESistance:RECall
 Set →
 → Query

Description Sets or queries whether A Value or B Value is the currently active value in CR static mode when the units are set to ohm.

Syntax :RESistance:RECall {A | 0 | B | 1}

Query Syntax :RESistance:RECall?

| | | |
|-----------|--------|--------------------|
| Parameter | A or 0 | CR Value = A Value |
| | B or 1 | CR Value = B Value |

| | | |
|------------------|---|--------------------|
| Return parameter | 0 | CR Value = A Value |
| | 1 | CR Value = B Value |

Example :RES:REC A
 Sets A value of CR resistance setting mode to active.

Query example :RES:REC?
 >0
 Return the CR resistance mode active setting.


Voltage Subsystem Commands

| | |
|----------------------|-----|
| :VOLTage[:VA] | 105 |
| :VOLTage:VB..... | 105 |
| :VOLTage:RECall..... | 106 |

:VOLTage[:VA]



Description Sets and queries for the CV mode “A Value” voltage or the +CV voltage value.

 **Note** The same value applies for each current range (High/Mid/Low).
The optional command node [:VA] can only be omitted when in static mode.

Syntax :VOLTage[:VA] {<NRf>[V] | MINimum | MAXimum }

Query Syntax :VOLTage[:VA]?

| | | |
|------------------|----------|-------------------------|
| Parameter | <NRf>[V] | “A Value” voltage value |
| | MINimum | Minimum voltage level |
| | MAXimum | Maximum voltage level |

Return parameter Return the voltage value of "A Value", by the "<NR2>" string.


Example :VOLT:VA MIN
Sets the voltage value to the minimum.

Query example :VOLT:VA?
>1.00
Voltage setting of " A Value " is set to 1V.

:VOLTage:VB



Description Sets and queries for the CV mode “B Value”.

 **Note** The same value applies for each current range (High/Mid/Low).

Syntax :VOLTage:VB {<NRf>[V] | MINimum | MAXimum }

Query Syntax :VOLTage:VB?

| | | |
|-----------|----------|-------------------------|
| Parameter | <NRf>[V] | "B Value" voltage value |
| | MINimum | Minimum voltage level |
| | MAXimum | Maximum voltage level |

Return parameter Return the voltage value of "B Value", by the "<NR2>" string.

Example :VOLT:VB MIN
Sets the voltage value to the minimum.

Query example :VOLT:VB?
>1.00
Voltage setting of " B Value " is set to 1V.

Set →

→ Query

:VOLTage:RECall

Description Sets or queries whether A Value or B Value is the currently active value in CV mode.

Syntax :VOLTage:RECall {A | 0 | B | 1}

Query Syntax :VOLTage:RECall?

| | | |
|-----------|--------|-----------------------------|
| Parameter | A or 0 | CV active setting = A Value |
| | B or 1 | CV active setting = B Value |

| | | |
|------------------|---|-----------------------------|
| Return parameter | 0 | CV active setting = A Value |
| | 1 | CV active setting = B Value |

Example :VOLT:REC A
Sets A value of CV setting mode to active.

Query example :VOLT:REC?
>0
Return the CV active setting.


Power Subsystem Commands

| | |
|------------------|-----|
| :POWer[:VA] | 107 |
| :POWer:VB | 108 |
| :POWer:L1 | 108 |
| :POWer:L2 | 109 |
| :POWer:SET | 110 |
| :POWer:LEVel | 110 |
| :POWer:T1 | 111 |
| :POWer:T2 | 112 |
| :POWer:FREQuency | 112 |
| :POWer:DUTY | 113 |
| :POWer:RECall | 114 |

:POWer[:VA]




Description Sets and queries for the “A Value” power of the CP static mode.

 **Note** A different current value can be set for each current range (High/Mid/Low).
The [:VA] node can only be omitted when in static mode.

Syntax :POWer[:VA] {<NRf>[W] | MINimum | MAXimum }

Query Syntax :POWer[:VA]?

| | | |
|------------------|----------|-----------------------|
| Parameter | <NRf>[W] | “A Value” power value |
| | MINimum | Minimum power level |
| | MAXimum | Maximum power level |

Return parameter Return the power value of "A Value", by the "<NR2>" string.

Example :POW:VA MIN
Sets the power value to the minimum.


Query example :POW:VA?
>10
Power setting of "A Value" is set to 10W.

Set →

→ Query

:POWer:VB

Description Sets and queries for the "B Value" power of the CP static mode.

 **Note** A different current value can be set for each current range (High/Mid/Low).

Syntax :POWer:VB {<NRf>[W] | MINimum | MAXimum }

Query Syntax :POWer:VB?

| | | |
|------------------|----------|-----------------------|
| Parameter | <NRf>[W] | "B Value" power value |
| | MINimum | Minimum power level |
| | MAXimum | Maximum power level |

Return parameter Return the power value of "B Value", by the "<NR2>" string.

Example :POW:VB MIN
Sets the power value to the minimum.

Query example :POW:VB?
>10
Power setting of "B Value" is set to 10W.


Set →

→ Query

:POWer:L1

Description Sets and queries for the "Level1" power of the CP dynamic mode.

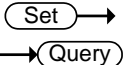
This command applies in "Dyna.Level" is "Value".

 **Note** A different "Level1" value can be set for different current ranges.


Syntax :POWer:L1 {<NRf>[W] | MINimum | MAXimum }

Query Syntax :POWer:L1?

| | | |
|------------------|----------|----------------------|
| Parameter | <NRf>[W] | "Level1" power value |
| | MINimum | Minimum power level |

| | |
|------------------|---|
| | MAXimum Maximum power level |
| Return parameter | Return the power value of "Level1", by the "<NR2>" string. |
| Example | :POW:L1 MIN Sets the power value to the minimum. |
| Query example | :POW:L1? >10 Power setting of " Level1" is set to 10W. |
| :POWer:L2 | <div style="text-align: right;">  </div> |

Description Sets and queries for the "Level2" power of the CP dynamic mode.
This command applies in "Dyna.Level" is "Value".

 **Note** A different "Level2" value can be set to different current ranges.

Syntax :POWer:L2 {<NRf>[W] | MINimum | MAXimum }

Query Syntax :POWer:L2?

| | | |
|------------------|----------------|----------------------|
| Parameter | <NRf>[W] | "Level2" power value |
| | MINimum | Minimum power level |
| | MAXimum | Maximum power level |

Return parameter Return the power value of "Level2", by the "<NR2>" string.

Example :POW:L2 MIN
Sets the power value to the minimum.

Query example :POW:L2?
>10
Power setting of " Level2" is set to 10W.

Set →
 → Query

:POWER:SET

| | | |
|------------------|--|---|
| Description | Sets and queries for the "Set" power of the CP dynamic mode. This command applies in "Dyna.Level" is "Percent". | |
| Syntax | :POWER:SET {<NRf>[W] MINimum MAXimum } | |
| Query Syntax | :POWER:SET? | |
| Parameter | <NRf>[W] | The power value at the time of "Level=100%" |
| | MINimum | Minimum power value |
| | MAXimum | Maximum power value |
| Return parameter | Return the power value of "Level = 100%", by the "<NR2>" string. | |
| Example | :POW:SET MIN Sets the minimum power value of "Level = 100%". | |
| Query example | :POW:SET? >10 power value of "Level = 100%" is set to 10W. | |
| Related Commands | :POWER:LEVel | |

Set →
 → Query

:POWER:LEVel

| | | |
|--------------|--|------------------------------------|
| Description | Sets and queries for the "Set" power of the CP dynamic mode. This command applies in "Dyna.Level" is "Percent". | |
| Syntax | :POWER:LEVel {<NRf> MINimum MAXimum } | |
| Query Syntax | :POWER:LEVel? | |
| Parameter | <NRf> | % of "Set" power level (unit is %) |
| | MINimum | Minimum % power level |

| | |
|------------------|--|
| | MAXimum Maximum % power level |
| Return parameter | Return the power of "% Level", by the "<NR2>" string. |
| Example | :POW:LEV MIN Sets the % level power value to the minimum. |
| Query example | :POW:LEV? >50 Percentage of the set power value is set to 50%. |
| Related Commands | :POWer:Set |



| | |
|------------------|---|
| Description | Sets and queries for the "Timer1" time of CP dynamic mode. This command applies in "Dyna.Time" is "T1/T2". |
| Syntax | :POWer:T1 {<NRf>[s] MINimum MAXimum } |
| Query Syntax | :POWer:T1? |
| Parameter | <NRf>[s] T1 timer setting. (unit is seconds) MINimum Minimum time MAXimum Maximum time |
| Return parameter | Return the setting of the timer T1, by the "<NR2>" string. |
| Example | :POW:T1 0.2 Sets the setting of the timer T1. |
| Query example | :POW:T1? >0.2 Return the setting of the timer T1. |
| Related Commands | :POWer:T2 |

:POWer:T2
 →
 →

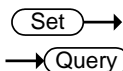
| | | |
|------------------|---|-------------------------------------|
| Description | Sets and queries for the "Timer2" time of CP dynamic mode. This command applies in "Dyna.Time" is "T1/T2". | |
| Syntax | :POWer:T2 {<NRf>[S] MINimum MAXimum } | |
| Query Syntax | :POWer:T2? | |
| Parameter | <NRf>[S] | T2 timer setting. (unit is seconds) |
| | MINimum | Minimum time |
| | MAXimum | Maximum time |
| Return parameter | Return the setting of the timer T2, by the "<NR2>" string. | |
| Example | :POW:T2 0.2 Sets the setting of the timer T2. | |
| Query example | :POW:T2? >0.2 Returns the setting of the timer T2. | |
| Related Commands | :POWer:T1 | |

:POWer:FREQuency
 →
 →

| | | |
|--------------|---|--|
| Description | Sets and queries for "Frequency" value of the CP dynamic mode. This command applies in "Dyna.Time" is "Freq/Duty". | |
| Syntax | :POWer:FREQuency {<NRf> MINimum MAXimum } | |
| Query Syntax | :POWer: FREQuency? | |
| Parameter | <NRf> | Sets the switching frequency. (unit is hertz). |

| | | |
|------------------|--|-------------------|
| | MINimum | Minimum frequency |
| | MAXimum | Maximum frequency |
| Return parameter | Return the switching frequency, by the "<NR2>" string. | |
| Example | :POW: FREQ 60 Sets the frequency to 60Hz. | |
| Query example | :POW: FREQ? >60 Returns the switching frequency as 60Hz. | |
| Related Commands | :POWer: DUTY | |

:POWer:DUTY



| | | |
|------------------|--|--------------------------------------|
| Description | Sets and queries for "Duty" % of the CP dynamic mode. This command applies in "Dyna.Time" is "Freq/Duty". | |
| Syntax | :POWer:DUTY {<NRf> MINimum MAXimum } | |
| Query Syntax | :POWer:DUTY? | |
| Parameter | <NRf> | Sets the duty cycle as a percentage. |
| | MINimum | Minimum duty |
| | MAXimum | Maximum duty |
| Return parameter | Return the duty cycle of positive, by the "<NR2>" string. | |
| Example | :POW:DUTY 50 Sets the duty cycle 50%. | |
| Query example | :POW:DUTY? >50 Returns the duty cycle as 50%. | |
| Related Commands | :POWer:FREQuency | |

Set →

→ Query

:POWer:RECall

Description Sets or queries whether A Value or B Value is the currently active value in CP mode.

Syntax :POWer:RECall {A | 0 | B | 1}

Query Syntax :POWer:RECall?

| | | |
|------------------|--------|-----------------------------|
| Parameter | A or 0 | CP active setting = A Value |
| | B or 1 | CP active setting = B Value |

| | | |
|-------------------------|---|-----------------------------|
| Return parameter | 0 | CP active setting = A Value |
| | 1 | CP active setting = B Value |

Example :POW:REC A
Sets A value of CP setting mode to active.

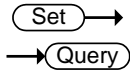
Query example :POW:REC?
>0
Return the CP active setting.

Program Commands

To execute / stop the program function, use: INPUT command.

| | |
|--|-----|
| :FUNction[:COMPLete][:RING]:TIME | 115 |
| :PROGram:STATe | 116 |
| :PROGram | 117 |
| :PROGram[:RECall]:DEFault | 118 |
| :PROGram:STARt..... | 118 |
| :PROGram:STEP..... | 119 |
| :PROGram:MEMory..... | 119 |
| :PROGram:RUN..... | 120 |
| :PROGram:ONTime..... | 120 |
| :PROGram:OFFTime | 121 |
| :PROGram:PFTime..... | 121 |
| :PROGram:STIME..... | 122 |
| [:PROGram]:CHAin:STARt | 123 |
| [:PROGram]:CHAin..... | 123 |
| [:PROGram]:CHAin:P2P | 124 |
| [:PROGram]:CHAin[:RECall]:DEFault..... | 125 |
| :PROGram:SAVE..... | 125 |

:FUNction[:COMPLete][:RING]:TIME

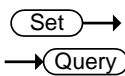


| | | |
|------------------|--|--|
| Description | Sets and queries for how long the alarm will buzz for after a program, NSEQ, FSEQ or OCP test function has finished. | |
| Syntax | :FUNction[:COMPLete][:RING]:TIME {<NR1> MINimum MAXimum INFINITY} | |
| Query Syntax | :FUNction[:COMPLete][:RING]:TIME? [MINimum MAXimum] | |
| Parameter | <NR1> | Alarm time in seconds (0~600); 0 = OFF |
| | MINimum | Minimum alarm time |
| | MAXimum | Maximum alarm time |
| | INFINITY | Sets the alarm time to infinity. |
| Return parameter | <NR1> | Alarm time in seconds |

| | |
|----------|-------------------------------------|
| Infinity | Infinite |
| OFF | Function complete ring time is off. |

Example :FUNC:TIME 5
Sets the function complete ring time to 5s.

Query example :FUNC:TIME?
>5
The function complete ring time is set as 5s.



:PROG:STATe

Description Sets and queries for the state of the program function.

Syntax :PROG:STATe {ON | OFF | PAUSE | CONTINUE | NEXT}

Query Syntax :PROG:STATe?

| | | |
|-----------|----------|-----------------------------------|
| Parameter | ON | Turn program function on |
| | OFF | Program function off |
| | PAUSE | Program function pause |
| | RUN | Program function running |
| | CONTINUE | Program function continue |
| | NEXT | Next step in the program function |

Return parameter Return the state of the program function.
Return the on status of program function, by the "ON,{STOP | RUN | PAUSE}" string. ("STOP": stopped, "RUN": running, "PAUSE": paused)
Return the off status of program function, by the "OFF" string: Program function is off.

Example :PROG:STAT ON
Turn program function on.

Query example :PROG:STAT?
 >ON,STOP
 Return the state of the program function.

Set →

→ Query

:PROGram

Description Sets and queries for all parameters to specified step of the program function.

Parameters of “Timing Edit for Program”

Syntax :PROGram (1),(2),(3),(4),(5),(6),(7),(8)

Query Syntax :PROGram?

| | | |
|------------------|--------------------|---|
| Parameter | (1) <NR1> | Program number (1~16) |
| | (2) <NR1> | Step number (1~16) |
| | (3) <NR1> | Internal memory number (1~256) |
| | (4) <ASCII string> | Processing settings AUTO: Run processing MANUAL: Wait for the start-up process. SKIP: Proceed to the next step |
| | (5) <NRf> | On-time (0.1~60)s |
| | (6) <NRf> | Off-time (Off,0.1~60)s |
| | (7) <NRf> | Delay time of the P/F (Off,0.0 ~119.9)s |
| | (8) <NRf> | Short time (Off,0.1 ~ On-Time)s |
| Return parameter | (1) <ASCII string> | Return the program function.{ON OFF} |
| | (2) <ASCII string> | Program number |
| | (3) <ASCII string> | Step number |
| | (4) <ASCII string> | Internal memory number |
| | (5) <ASCII string> | Processing is returned by Auto/Manual/Skip. |

- (6) <ASCII string> On-time (unit is seconds)
- (7) <ASCII string> Off-time (unit is seconds)
- (8) <ASCII string> Delay time of the P/F (unit is seconds)
- (9) <ASCII string> Short time (unit is seconds)

Example :PROG 2,3,1,AUTO,40.1,0,0,0
 Set all the parameters to specified program step.

Query example :PROG?
 >Program:OFF; Start:1, Step:1, Memory:1, Run:Skip, On-Time:0.1, Off-Time:0.0, P/F-Time:0.0, Short-Time:0.0
 Return the parameters of the program step that is selected after the state of the program function.

:PROG[am[:RECall]:DEFault (Set) →

Description All steps of a selected program are set by default value.

Syntax :PROG[am[:RECall]:DEFault

Example :PROG:DEF
 All steps of a selected program are set by default value.

(Set) →

:PROG[am:START → (Query)

Description Sets and queries for select program number. (Contents of PROG:)

Syntax :PROG[am:START <NR1>

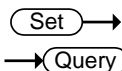
Query Syntax :PROG[am:START?

Parameter <NR1> Program number to select. (1~16)

Return parameter Return the Program number in selected, by the "Start:<NR1>" string.

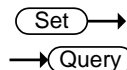
| | |
|---------------|--|
| Example | :ROG:STAR 1 Sets to 1 of program number. |
| Query example | :PROG:STAR? >Start:1 Return The selected program number. |

:PROG:STEP



| | |
|------------------|--|
| Description | Sets and queries for the step number of the program to select. (Contents of STEP:) |
| Syntax | :PROG:STEP <NR1> |
| Query Syntax | :PROG:STEP? |
| Parameter | <NR1> Step number to select. (1~16) |
| Return parameter | Return the step number of the program, by the "Step:<NR1>" string. |
| Example | :PROG:STEP 1 Sets to 1 of step number. |
| Query example | :PROG:STEP? >Step:1 Return the step number of the program. |

:PROG:MEMory



| | |
|------------------|---|
| Description | Sets and queries for memory number of selected program steps. (Contents of Memory:) |
| Syntax | :PROG:MEMory <NR1> |
| Query Syntax | :PROG:MEMory? |
| Parameter | <NR1> Sets the memory number. (1~256) |
| Return parameter | Return the memory number of selected, by the "Memory:M<NR1>" string. |
| Example | :PROG:MEM 1 Sets to 1 of memory number. |

Query example :PROG:MEM?
 >Memory:M 1
 Return the memory number of selected.

Set →

:PROG:RUN

→ Query

Description Sets and queries for execution process of selected program steps. (Contents of Run:)

Syntax :PROG:RUN { AUTO | MANual | SKIP}

Query Syntax :PROG:RUN?

| | | |
|-----------|--------|------------------------------------|
| Parameter | AUTO | Sets to auto-run processing. |
| | MANual | Sets to wait run. |
| | SKIP | Sets to the next processing steps. |

Return parameter Return the execution process, by the "Run:{Auto | Manual | Skip}" string.

Example :PROG:RUN AUTO
 Sets to auto-run processing.

Query example :PROG:RUN?
 >Run:Auto
 Return the execution process.

Set →

:PROG:ONTime

→ Query

Description Sets and queries for On time of selected program steps. (Contents of On-Time:)

Syntax :PROG:ONTime <NRf>

Query Syntax :PROG:ONTime?

| | | |
|-----------|-------|--------------------------|
| Parameter | <NRf> | On time (0.1~60 seconds) |
|-----------|-------|--------------------------|

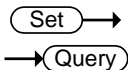
Return parameter Return the On time of selected program steps, by the "On-Time:<NR2>" string.

Example :PROG:ONT 1
 Sets to 1 second On time.

Query example :PROG:ONT?
>On-Time:0.1
Return in seconds On time.

Related :PROG:PFTime
Commands :PROG:STIME

:PROG:OFFTime



Description Sets and queries for Off time of selected program steps. (Contents of Off-Time:)

Syntax :PROG:OFFTime {<NRf> | OFF}

Query Syntax :PROG:OFFTime?

| | | |
|-----------|-------|--|
| Parameter | <NRf> | Off time (Off,0.1~60 seconds) Setting of 0 is off Setting function. |
| | OFF | Setting off function |

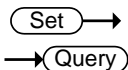
Return parameter Return the Off time of selected program steps, by the "Off-Time:<NR2>" string.
"Off-Time:0.0" is a function off.

Example :PROG:PFT 1
Sets to 1 second Off time.

Query example :PROG:OFFT?
>Off-Time:1.0
Return the Off time.

Related :PROG:PFTime
Commands

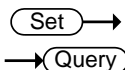
:PROG:PFTime



Description The pass / fail judgment (Go-NoGo Test) of the program selected step.
Sets and queries for the judgment delay time. (Contents of P/F-Time:)

Syntax :PROG:PFTime {<NRf> | OFF}

| | | |
|------------------|--|--|
| Query Syntax | :PROG:PFTime? | |
| Parameter | <NRf> | Set in seconds the judgment delay time. (Off,0.0~119.9)s 0 setting is off a judgment function. |
| | OFF | Sets off Judgment function |
| Return parameter | Return the judgment delay time, by the "P/F-Time:<NR2>" string. "P/F-Time:0.0" is a function off. | |
| Example | :PROG:PFT 1 Delay time of the judgment sets 1 second. | |
| Query example | :PROG:PFT >P/F-Time:1.0 Return the judgment delay time. | |
| Related Commands | :PROG:OFFTime :PROG:ONTime | |



:PROG:STIME

| | | |
|------------------|---|---|
| Description | Sets and queries for load short time of selected program step. (Contents of Short-Time:) | |
| Syntax | :PROG:STIME <NR1> | |
| Query Syntax | :PROG:STIME? | |
| Parameter | <NRf> | Sets in time seconds of load short time. (Off,0.1 ~ On-Time)s 0 setting is the off feature of load short. |
| | OFF | Sets the off feature of load short. |
| Return parameter | Return the setting time of load short, by the "Short-Time:<NR2>" string. | |
| Example | :PROG:STIM 1 Sets to 1 second of load short. | |

Query example :PROG:STIM?
>Short-Time:0.0
Return the setting time of load short.

Related :PROG:ONTime
Commands

[[:PROG:]:CHAI:STARt

Set →
→ Query

Description Sets and queries for the start program number of the program chain. (Contents of Start)
Parameters of "Program Chain Set"

Syntax [[:PROG:]:CHAI:STARt <NR1>

Query Syntax [[:PROG:]:CHAI:STARt?

Parameter <NR1> Start program number.

Return parameter Return the start program number, by the "P<NR1>" string.

Example :CHA:STAR 1
Sets to number 1 start program.

Query example :CHA:STAR?
>P1
Return the start program number.

[[:PROG:]:CHAI

Set →
→ Query

Description Sets and queries for all chain of the program chain. (Contents of PXX)

Syntax [[:PROG:]:CHAI
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16)

Query Syntax [[:PROG:]:CHAI?

Parameter (1)~(16) <NR1> Is a program number to be chain.
"OFF" sets off the chain.

| | | |
|------------------|---|---|
| Return parameter | <ASCII string> | Return the setting value of all program chain. "Off" is the end of the chain. |
| Example | <p>:CHA 4,OFF,OFF,5,6,OFF,OFF,OFF,OFF,OFF,OFF,OFF,OFF,OFF,OFF</p> <p>Configures the program chain as follows:</p> <p>P01→P04→P05→P06→OFF</p> | |
| Query example | <p>:CHA?</p> <p>>P1->P4;P2->Off;P3->Off;P4->P5;P5->P6;P6->Off;P7->Off;P8->Off;P9->Off;P10->Off;P11->Off;P12->Off;P13->Off;P14->Off;P15->Off;P16->Off</p> <p>Return the setting value of all program chain.</p> | |





[[:PROG]am]:CHAin:P2P

| | | |
|------------------|--|---|
| Description | Sets and queries for a single link in the program chain. | |
| Syntax | [:PROG]am]:CHAin:P2P (1),(2) | |
| Query Syntax | [:PROG]am]:CHAin:P2P? PX | |
| Parameter | (1) <NR1> | (1~16) Program number of single link. |
| | (2) {<NR1> OFF} | (1~16) Program number that follows. |
| Query parameter | PX | P1~P16 The program number of the first link of the chain that you are inquiring about. |
| Return parameter | PX→PX OFF | Returns the program number that follows or OFF. |
| Example | <p>:CHA:P2P 4,3</p> <p>Program 4 will chain set in the program 3.P4→P3</p> | |

Query example :CHA:P2P? P4
>P4->3

Indicates that program 3 follows from program 4 in the chain. P4→P3

[[:PROGram]:CHAI[:RECall]:DEFault

Set →

Description Turn off all the program chains.

Syntax [[:PROGram]:CHAI[:RECall]:DEFault

Example :CHA:DEF

Turn off all the program chain.

:PROGram:SAVE

Set →

Description Save program.

Syntax :PROG:SAVE

Example :PROG:SAVE

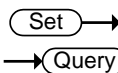
Save program.

Normal sequence Commands

To execute / stop the test, use: INPUT command.

| | |
|------------------------------|-----|
| :NSEquence:STATe | 126 |
| :NSEquence | 127 |
| :NSEquence:STARt | 129 |
| :NSEquence:NUMBer | 130 |
| :NSEquence:MEMO | 130 |
| :NSEquence:MODE | 131 |
| :NSEquence:RANGe | 131 |
| :NSEquence:LOOP | 132 |
| :NSEquence:LLOAD | 133 |
| :NSEquence:LAST | 133 |
| :NSEquence:CHAin | 134 |
| :NSEquence:EDIT | 134 |
| :NSEquence:EDIT:POINT | 136 |
| :NSEquence:EDIT:END | 136 |
| :NSEquence[:DELet]:ALL | 136 |
| :NSEquence:SAVE | 137 |
| :NSEquence:COTime | 137 |

:NSEquence:STATe



| | | |
|--------------|---|------------------------------|
| Description | Sets and queries for the state of the Normal Sequence function. | |
| Syntax | :NSEquence:STATe {OFF ON PAUSE CONTINUE EXT} | |
| Query Syntax | :NSEquence:STATe? | |
| Parameter | OFF | Normal sequence function Off |
| | ON | Normal sequence function On |
| | PAUSE | Pause sequence function |
| | CONTINUE | Continue sequence function |
| | NEXT | Go to next sequence function |

Return parameter Return the state of the normal sequence function.

Return the on status of the normal sequence function, by the “ON, {STOP | RUN | PAUSE}” string. (“STOP”: stopped, “RUN”: running, “PAUSE”: Suspended)

Return the off status of the normal sequence function, by the “OFF” string. Normal sequence function is off

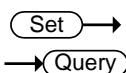
Example :PROG:STAT ON

Turn on the normal sequence function.

Query example :PROG:STAT?

>ON,STOP

Return the state of the normal sequence function.



:NSEquence

Description Sets and queries for parameters of the Normal Sequence function.

Parameters of “Timing Edit for Normal Sequence”



Note

Cannot change the range and mode of the load when there is a step data.

Syntax :NSEquence (1),(2),(3),(4),(5),(6),(7),(8),(9)

Query Syntax :NSEquence?

| | | |
|------------------|--------------------|---|
| Parameter | (1) <NR1> | Sets the start sequence number.(1~10) |
| | (2) <NR1> | Sets the sequence number.(1~10) |
| | (3) <ASCII string> | Sets the note of up to 12 characters. Enclose the string in double coat. |
| | (4) <ASCII string> | Sets the load mode. {CC CR CV CP} |

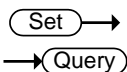
| | | |
|------------------|--------------------|---|
| | (5) <ASCII string> | Sets the load range. {IHVH IMVH ILVH IHVL IMVL ILVL} |
| | (6) <ASCII string> | Sets the loop count of the sequence. <NR1> 0 is infinite times. 1~9999. INFINITY |
| | (7) <ASCII string> | Sets the Last Load after the end. {ON OFF} |
| | (8) <NRf> | Sets the Last Value after the end. (The Last Load ON is enabled by ON.) |
| | (9) <ASCII string> | Sets the chain of the next sequence. <NR1> : Sequence number OFF : No chain |
| Return parameter | (1) <ASCII string> | Return whether it is in normal sequence function state. {ON OFF} |
| | (2) <ASCII string> | Return the start sequence number setting. |
| | (3) <ASCII string> | Return the sequence number setting. |
| | (4) <ASCII string> | Return the contents of the memo. |
| | (5) <ASCII string> | Return the load mode setting. {CC CR CV CP} |
| | (6) <ASCII string> | Return the load range setting. {IHVH IMVH ILVH IHVL IMVL ILVL} |
| | (7) <ASCII string> | Return the loop count of the sequence. (Infinite times is the string "Infinity".) |

| | |
|--------------------|--|
| (8) <ASCII string> | Return the Last Load after the end. {ON OFF} |
| (9) <ASCII string> | Return the Last Value after the end. |
| (10)<ASCII string> | Return the chain setting of the next sequence. (No-chain is a string "OFF".) |

Example :NSEQ 1,1,"ABC",CC,ILVL,5,ON,1.5000,OFF
 Sets the parameters of "Timing Edit for Normal Sequence".

Query example :NSEQ?
 >NSeq:ON; Start:1, Seq No:1, Memo:ABC, Mode:CC, Range:ILVL, Loop:5, Last Load:ON, Last:1.5000, Chain:Off
 Return the parameters of "Timing Edit for Normal Sequence".

:NSEquence:START



Description Sets and queries for the start sequence number of the normal sequence. (Contents of Start:)

Syntax :NSEquence:START <NR1>

Query Syntax :NSEquence:START?

Parameter <NR1> Sets the start sequence number. (1~10)

Return parameter Return the start sequence number of the normal sequence, by the "Start:<NR1>" string.

Example :NSEQ:STAR 1
 Sets the start sequence number of the normal sequence.

Query example :NSEQ:STAR?
 >Start:1
 Return the start sequence number of normal sequence.

Set →
 → Query

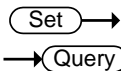
:NSEquence:NUMBER

| | |
|------------------|---|
| Description | Sets and queries for the sequence number of the normal sequence. (Contents of Seq.No:) |
| Syntax | :NSEquence:NUMBER <NR1> |
| Query Syntax | :NSEquence:NUMBER? |
| Parameter | <NR1> Sets the sequence number. (1~10) |
| Return parameter | Return the sequence number of the normal sequence by the "Seq No:<NR1>" string. |
| Example | :NSEQ:NUMB 1 Sets to 1 of normal sequence number. |
| Query example | :NSEQ:NUMB? >Seq No:1 Return a sequence number that is selected. |

Set →
 → Query

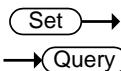
:NSEquence:MEMO

| | |
|------------------|--|
| Description | Sets and queries for the memo string of normal sequence. (Contents of Memo:) |
| Syntax | :NSEquence:MEMO <ASCII string> |
| Query Syntax | :NSEquence:MEMO? |
| Parameter | <ASCII string> Set the characters up to 12 characters. Enclose the string in double coat. |
| Return parameter | Return the memo of normal sequence that is selected, by the "Memo:<ASCII string>" string. |
| Example | :NSEQ:MEMO "ABCD" Sets the memo of step. |
| Query example | :NSEQ:MEMO? >Memo:ABCD Return the memo of step. |



:NSEquence:MODE

| | | |
|------------------|---|-----------------------------------|
| Description | Sets and queries for the operating mode of the selected normal sequence. (Contents of Mode:) | |
| Syntax | :NSEquence:MODE {CC CR CV CP} | |
| Query Syntax | :NSEquence:MODE? | |
| Parameter | CC | Sets to constant current mode. |
| | CR | Sets to constant resistance mode. |
| | CV | Sets to constant voltage mode. |
| | CP | Sets to constant power mode. |
| Return parameter | Return the operating mode of the normal sequence, by the “Mode:{CC CR CV CP}” string. | |
| Example | :NSEQ:MODE CC Sets the operating mode of the normal sequence. | |
| Query example | :NSEQ:MODE? >Mode:CC Return the operating mode of the normal sequence. | |



:NSEquence:RANGe

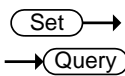
| | | |
|--------------|---|---|
| Description | Sets and queries for the operating range of the selected normal sequence. (Contents of Range:) | |
| Syntax | :NSEquence:RANGe { IHVH IMVH ILVH IHVL IMVL ILVL} | |
| Query Syntax | :NSEquence:RANGe? | |
| Parameter | IHVH | Sets High current range and High voltage range. |
| | IMVH | Sets Middle current range and High voltage range. |
| | ILVH | Sets Low current range and High voltage range. |

| | |
|------|--|
| IHVL | Sets High current range and Low voltage range. |
| IMVL | Sets Middle current range and Low voltage range. |
| ILVL | Sets Low current range Low voltage range. |

Return parameter Return the operating range of the normal sequence, by the "Range:{IHVH | IMVH | ILVH | IHVL | IMVL | ILVL}" string.

Example :NSEQ:RANG IHVL
 Sets operating of High current range and Low voltage range.

Query example :NSEQ:RANG?
 > Range:IHVL
 Return the operating range of settings.



:NSEQ:LOOP

Description Sets and queries for number of loops of normal sequence. (Contents of Loop:)

Syntax :NSEQ:LOOP {<NR1> | INFinity }

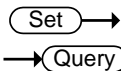
Query Syntax :NSEQ:LOOP?

| | | |
|------------------|----------|------------------------------|
| Parameter | <NR1> | Loop count setting. (1~9999) |
| | INFinity | Infinite number of times. |

Return parameter Return the number of loops of normal sequence, by the "Loop:{<NR1> | Infinity}" string.

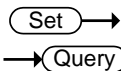
Example :NSEQ:LOOP 1
 Sets 1 to loop count of normal sequence.

Query example :NSEQ:LOOP?
 >Loop:Infinity
 Return the loop number of normal sequence.



:NSequence:LLOAD

| | | |
|------------------|--|----------------------|
| Description | Sets and queries for the Last Load state after the end of the normal sequence. (Contents of Last Load:) | |
| Syntax | :NSequence:LLOAD {ON OFF} | |
| Query Syntax | :NSequence:LLOAD? | |
| Parameter | ON | Turns Last Load on. |
| | OFF | Turns Last Load off. |
| Return parameter | Return the Last Load state of the load after the end of the step, by the “Last Load:{ON OFF}” string. | |
| Example | :NSEQ:LLOAD ON Turns Last Load on. | |
| Query example | :NSEQ:LLOAD? >Last Load:OFF Return the Last Load state after the end of the step. | |
| Related Commands | :NSequence:LAST | |



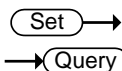
:NSequence:LAST

| | | |
|------------------|--|--|
| Description | Sets and queries for load value after the end of the normal sequence. (Contents of Last:) | |
| Syntax | :NSequence:LAST <NR2> | |
| Query Syntax | :NSequence:LAST? | |
| Parameter | <NR2> | Sets load value after the end of step. |
| Return parameter | Return load value of after the step the end, by the “Last:<NR2>” string. | |
| Example | :NSEQ:LAST 1 Sets load value of after the step the end. | |

Query example :NSEQ:LAST?
 >Last:1.00
 Return load value of after the step the end.

Related Commands :NSEquence:LLOAD

:NSEquence:CHAIIn



Description Query and settings for the next sequence in the chain. (Contents of Chain:)

Syntax :NSEquence:CHAIIn {<NR1> | OFF}

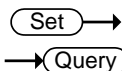
Query Syntax :NSEquence:CHAIIn?

Parameter <NR1> The sequence number of the next sequence in the chain.
 OFF chain is end.

Return parameter Return the chain of Normal sequence number, by the "Chain: {<NR1>|Off}" string. "Off" is the end of the chain.

Example :NSEQ:CHA 1
 Sets the next sequence in the chain as S01.

Query example :NSEQ:CHA?
 >Chain:1
 Returns the next sequence in the chain (S01).



:NSEquence:EDIT

Description Sets and queries for the data edit of normal sequence.
 Parameters of "Data Edit for Normal Sequence"

Syntax :NSEquence:EDIT
 (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11)

Query Syntax :NSEquence:EDIT?

Parameter (1) <NR1> The edit step number.

| | | |
|------------------|--|---|
| | (2) <NR1> | The total number of steps. |
| | (3) <NRf> | Sets a load value of operation mode. |
| | (4) <NR1> | Sets hours. |
| | (5) <NR1> | Sets minute. |
| | (6) <NR1> | Sets seconds. |
| | (7) <NR1> | Sets milliseconds. |
| | (8) <ASCII string> | Load setting. {ON OFF} |
| | (9) <ASCII string> | Lamp operation setting. {ON OFF} |
| | (10)<ASCII string> | Trigger Output setting. {ON OFF} |
| | (11)<ASCII string> | PAUSE Setting. {ON OFF} |
| Return parameter | (1) <ASCII string> | Return the edit step number/total step number. |
| | (2) <ASCII string> | Return the setting value of the load of the operating mode. |
| | (3) <ASCII string> | Return the set time. |
| | (4) <ASCII string> | Return the setting of the load state. |
| | (5) <ASCII string> | Return the setting of the lamp operation. |
| | (6) <ASCII string> | Return the setting of TRIG OUT. |
| | (7) <ASCII string> | Return the setting of pause. |
| Example | :NSEQ:EDIT 1,2,1,1,2,3,4,OFF,OFF,OFF,OFF Sets the data of normal step sequence program. | |
| Query example | :NSEQ:EDIT? > Step:1/2, Value:1.000, Time:1H:2M:3S:4mS, LOAD:OFF, TRIG OUT:OFF, RAMP:OFF, PAUSE:OFF Return the selected step data of normal sequence program. | |

:NSEquence:EDIT:POINT Set →
→ Query

| | | |
|--------------------------------|---|--|
| Description | Sets and queries for the edit step number of the normal sequence. | |
| Syntax | :NSEquence:EDIT:POINT {<NR1>} | |
| Query Syntax | :NSEquence:EDIT:POINT? | |
| Parameter/ Return parameter | <NR1> | 1~1000 The edit step number of the normal sequence. |
| Example | :NSEquence:EDIT:POINT 10 Sets the edit step number 10. | |
| Query example | :NSEquence:EDIT:POINT? >10 Returns the edit step number. | |
| Related Commands | :NSEquence:EDIT:END? | |

:NSEquence:EDIT:END → Query

| | | |
|------------------|--|--------|
| Description | Returns the end of edit step number in the normal sequence. | |
| Query Syntax | :NSEquence:EDIT:END? | |
| Return parameter | <NR1> | 1~1000 |
| Query Example | :NSEquence:EDIT:END? > 20 Returns the end of edit step number. | |

:NSEquence[:DELet]:ALL Set →

| | | |
|-------------|---|--|
| Description | Delete all the steps of the selected normal sequence. | |
| Syntax | :NSEquence[:DELet]:ALL | |

Example :NSEQ:ALL
Delete all the steps of the selected normal sequence.

:NSEquence:SAVE (Set) →

Description Save program of normal sequence.

Syntax :NSEquence:SAVE

Example :NSEQ:SAVE
Save program of normal sequence.

(Set) →

:NSEquence:COTime → (Query)

Description Sets and queries for the display timer of the normal sequence.

Syntax :NSEquence:COTime {UP|DOWN}

Query Syntax :NSEquence:COTime?

Parameter UP Elapsed time (Count up)

DOWN Remaining (Count down)

Return parameter :NSEQ:COT UP
Sets the display timer to UP (elapsed time).

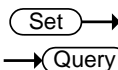
Example :NSEQ:COT?
>UP
Returns the display timer as UP (elapsed time).

Fast sequence Commands

To execute / stop the test, use: INPUT command.

| | |
|------------------------------|-----|
| :FSEquence:STATe | 138 |
| :FSEquence | 139 |
| :FSEquence:MEMO | 140 |
| :FSEquence:MODE | 141 |
| :FSEquence:RANGe | 141 |
| :FSEquence:LOOP | 142 |
| :FSEquence:TBASe | 143 |
| :FSEquence:LLOAD | 143 |
| :FSEquence:LAST | 144 |
| :FSEquence:RPTStep | 144 |
| :FSEquence:EDIT | 145 |
| :FSEquence:EDIT:POINT | 146 |
| :FSEquence:EDIT:END | 146 |
| :FSEquence[:DELet]:ALL | 146 |
| :FSEquence[:EDIT]:FILL | 147 |
| :FSEquence:SAVE | 147 |

:FSEquence:STATe



Description Sets and queries for the state of the fast sequence function.

Syntax :FSEquence:STATe {OFF | ON}

Query Syntax :FSEquence:STATe?

| | | |
|------------------|-----|---------------------------------|
| Parameter | OFF | Turn fast sequence function off |
| | ON | Turn fast sequence function on |

Return parameter Return the state of the fast sequence function.
 “ON, {STOP | RUN }” string. (“STOP”: stopped, “RUN”: running)
 “OFF” string means that fast sequence is off.

Example :FSEQ:STAT ON
 Turn on the fast sequence.

Query example :FSEQ:STAT?

>ON,STOP

The fast sequence function is on, and stopped

→ Set →

→ Query

:FSEquence

Description Sets and queries for parameters of fast sequence. parameters of "Timing Edit Fast Sequence"

Syntax :FSEquence (1),(2),(3),(4),(5),(6),(7)

Query Syntax :FSEquence?

| | | |
|------------------|--------------------|---|
| Parameter | (1) <ASCII string> | Sets the note of up to 12 characters. Enclose the string in double coat. |
| | (2) <ASCII string> | Sets the load mode. {CC CR} |
| | (3) <ASCII string> | Sets the load range. {IHVH IMVH ILVH IHVL IMVL ILVL} |
| | (4) <ASCII string> | Return the loop count of the sequence. (An infinite time is the string "Infinity".) |
| | (5) <NRf> | Sets the time base. (unit is seconds) |
| | (6) <ASCII string> | Sets the Last Load after the end. {ON OFF} |
| | (7) <NRf> | Sets the Last Value after the end. (It is enabled by the Last Load ON.) |
| | (8) <NR1> | Sets the step number setting of the loop last. |
| Return parameter | (1) <ASCII string> | Return whether it is in fast sequence function state. {ON OFF} |
| | (2) <ASCII string> | Return the contents of the memo. |

| | |
|--------------------|---|
| (3) <ASCII string> | Return the load mode setting. {CC CR} |
| (4) <ASCII string> | Return the load range setting. {IHVH IMVH ILVH IHVL IMVL ILVL} |
| (5) <ASCII string> | Return the loop count of the sequence. (An infinite time is the string "Infinity".) |
| (6) <ASCII string> | Return the time base. (unit is m seconds) |
| (7) <ASCII string> | Return the Last Load after the end. {ON OFF} |
| (8) <ASCII string> | Return the Last Value after the end. |
| (9) <ASCII string> | Return the step number of the last loop. |

Example :FSEQ "ABC",CC,IHVL,1,0.025,OFF,1.0,1
Sets the parameters of "Timing Edit Fast Sequence".

Query example :FSEQ?
> FSeq:OFF; Memo:ABC, Mode:CC, Range:IHVL, Loop:1, Time Base:25.000, Last Load:OFF, Last:1.00, RPTSTEP:1
Return the parameters of "Timing Edit Fast Sequence".

:FSEquence:MEMO Set →
→ Query

Description Sets and queries for the memo of fast sequence. (Contents of Memo:)

Syntax :FSEquence:MEMO <ASCII string>

Query Syntax :FSEquence:MEMO?

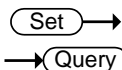
Parameter <ASCII string> Set the characters up to 12 characters.
Enclose the string in double coat.

Return parameter Return the memo of fast sequence, by the "Memo:<ASCII string>" string.

Example :FSEQ:MEMO "ABC"
Sets the memo of fast sequence.

Query example :FSEQ:MEMO?
>Memo: ABCD
Return the memo of fast sequence.

:FSEquence:MODE



Description Sets and queries for the operating mode of fast sequence. (Contents of Mode:)

Syntax :FSEquence:MODE {CC | CR}

Query Syntax :FSEquence:MODE?

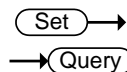
Parameter CC Sets to constant current mode.
CR Sets to constant resistance mode.

Return parameter Return the operating mode of fast sequence, by the "Mode:{CC | CR}" string.

Example :FSEQ:MODE CC
Sets the operating mode of the fast sequence.

Query example :FSEQ:MODE?
>Mode:CC
Return the operating mode of the fast sequence.

:FSEquence:RANGe



Description Sets and queries for the operating range of the fast sequence. (Contents of Range:)

Syntax :FSEquence:RANGe { IHVH | IMVH | ILVH | IHVL | IMVL | ILVL }

Query Syntax :FSEquence:RANGe?

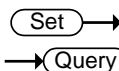
Parameter IHVH Sets High current range and High voltage range.

| | |
|------|---|
| IMVH | Sets Middle current range and High voltage range. |
| ILVH | Sets Low current range and High voltage range. |
| IHVL | Sets High current range and Low voltage range. |
| IMVL | Sets Middle current range and Low voltage range. |
| ILVL | Sets Low current range Low voltage range. |

Return parameter Return the operating range of the fast sequence, by the "Range:{IHVH | IMVH | ILVH | IHVL | IMVL | ILVL}" string.

Example :FSEQ:RANG IHVL
Sets operating of High current range and Low voltage range.

Query example :FSEQ:RANG?
>Range:IHVH
Return the operating range of settings.



:FSEQuence:LOOP

Description Sets and queries for number of loops of fast sequence. (Contents of Loop:)

Syntax :FSEQuence:LOOP {<NR1> | INFIInity}

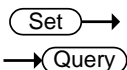
Query Syntax :FSEQuence:LOOP?

Parameter <NR1> Loop count setting. (1~9999)
INFIInity Infinite number of times.

Return parameter Return the number of loops of fast sequence, by the "Loop:{<NR1> | InFIInity}" string.

Example :FSEQ:LOOP 1
Sets 1 to loop count of fast sequence.

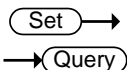
Query example :FSEQ:LOOP?
 >Loop:Infinity
 Return the loop number of fast sequence.



:FSEquence:TBASe

| | |
|------------------|---|
| Description | Sets and queries for the time-based of fast sequence. (Contents of Time Base:) |
| Syntax | :FSEquence:TBASe <NRf> |
| Query Syntax | :FSEquence:TBASe? |
| Parameter | <NRf> Sets the time-base. (Unit is second.) |
| Return parameter | Return the time-based of fast sequence, by the "Time Base:<NR2>" string. |
| Example | :FSEQ:TBAS 0.6 Sets the 0.6 seconds to the time-base of fast sequence. |

Query example :FSEQ:TBAS?
 >Time Base:0.60000
 Return the time-base.



:FSEquence:LLOAD

| | |
|------------------|--|
| Description | Sets and queries for the Last Load state of Fast sequence. (Contents of Last Load:) |
| Syntax | :FSEquence:LLOAD {ON OFF} |
| Query Syntax | :FSEquence:LLOAD? |
| Parameter | ON Turns Last Load on. OFF Turns Last Load off. |
| Return parameter | Return the Last Load state, by the "Last Load:{ON OFF}" string. |
| Example | :FSEQ:LLOAD ON Turns Last Load on. |

Query example :FSEQ:LLOAD?
 >Last Load:OFF
 Return the Last Load state of Fast sequence.

Related Commands :FSEquence:LAST

:FSEquence:LAST (Set) →
 → (Query)

Description Sets and queries for the Load Value after the end of Fast sequence. (Contents of Last:)

Syntax :FSEquence:LAST <NRf>

Query Syntax :FSEquence:LAST?

Parameter <NRf> Sets the load setting value of after the end of step. (It is enabled by load conditions ON.)

Return parameter Return the load setting value of after the end of step, by the "Last:<NR2>" string.

Example :FSEQ:LAST1
 Sets the load setting value of after the end of step.

Query example :FSEQ:LAST?
 >Last:0.070000
 Return the load setting value of after the end of step.

Related Commands :FSEquence:LLOAD

:FSEquence:RPTStep (Set) →
 → (Query)

Description Sets and queries for the last step number per loop of the fast sequence. (Contents of RPTSTEP:)

Syntax :FSEquence:RPTStep <NR1>

Query Syntax :FSEquence:RPTStep?

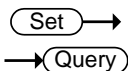
Parameter <NR1> Sets the step number of the last loop.

Return parameter Return the step number of the loop end of the fast sequence, by the "RPTSTEP:<NR1>" string.

Example :FSEQ:RPTS 1
Sets the last step number per loop.

Query example :FSEQ:RPTS?
>RPTSTEP:1
Return the last step number per loop.

:FSEquence:EDIT



Description Sets and queries for data of fast sequence.
Parameters of "Data Edit for Fast Sequence"

Syntax :FSEquence:EDIT (1),(2),(3),(4)

Query Syntax :FSEquence:EDIT?

| | | |
|-----------|--------------------|--|
| Parameter | (1) <NR1> | Editing step. |
| | (2) <NR1> | Step total number. |
| | (3) <NRF> | Sets the load value of the operation mode. |
| | (4) <ASCII string> | Sets the TRIG OUT. {ON OFF} |

| | | |
|------------------|--------------------|--|
| Return parameter | (1) <ASCII string> | Return the total of an edit step and a step. |
| | (2) <ASCII string> | Return the load setting value of the operation mode. |
| | (3) <ASCII string> | Return the TRIG OUT. |

Example :FSEQ:EDIT 2,6,1,ON
Sets the data of the fast sequence.

Query example :FSEQ:EDIT?
> Step:0001/0003; Value:0.00, TRIG OUT:OFF
Return the data of the fast sequence.

:FSEquence:EDIT:POINT Set →
→ Query

| | | |
|---------------------------------|---|--|
| Description | Sets and queries for the edit step number of the fast sequence. | |
| Syntax | :FSEquence:EDIT:POINT {<NR1>} | |
| Query Syntax | :FSEquence:EDIT:POINT? | |
| Parameter / Return parameter | <NR1> | 1~1000 The edit step number of the fast sequence. |
| Example | :FSEquence:EDIT:POINT 10 Sets the edit step number 10. | |
| Query example | :FSEquence:EDIT:POINT? >10 Returns the edit step number. | |
| Related Commands | :FSEquence:EDIT:END? | |

:FSEquence:EDIT:END → Query

| | | |
|------------------|--|--------|
| Description | Returns the end of edit step number in the fast sequence. | |
| Query Syntax | :FSEquence:EDIT:END? | |
| Return parameter | <NR1> | 1~1000 |
| Query Example | :FSEquence:EDIT:END? > 20 Returns the end of edit step number. | |

:FSEquence[:DELet]:ALL Set →

| | | |
|-------------|---|--|
| Description | Delete all programs of the fast sequence. | |
| Syntax | :FSEquence[:DELet]:ALL | |

Example :FSEQ:ALL
Delete all programs of the fast sequence.

:FSEquence[:EDIT]:FILL

Set →
→ Query

Description Query and setting for FILL of fast sequence.
Parameters of "Fill Edit for Fast Sequence"

Syntax :FSEquence[:EDIT]:FILL (1),(2),(3),(4)

Query Syntax :FSEquence[:EDIT]:FILL?

| | | |
|-----------|-----------|--|
| Parameter | (1) <NRf> | Sets the start load value of the operating mode. |
| | (2) <NRf> | Sets the end load value of the operating mode. |
| | (3) <NR1> | Sets the step number of start. |
| | (4) <NR1> | Sets the step number of end. |

| | | |
|------------------|-------------------|--|
| Return parameter | (1)<ASCII string> | Return the start load value of the operating mode. |
| | (2)<ASCII string> | Return the end load value of the operating mode. |
| | (3)<ASCII string> | Return the step number of start. |
| | (4)<ASCII string> | Return the step number of end. |

Example :FSEQ:FILL 0,5,1,6
Sets the terms of the FILL to Fast sequence program.

Query example :FSEQ:FILL?
>Start Value:0.00, End Value:5.00, Start Step:1, End Step:6
Return the setting value of FILL of fast sequence.

:FSEquence:SAVE

Set →

Description Save program of fast sequence.

Syntax :FSEquence:SAVE

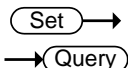
Example : FSEQ:SAVE
Save program of fast sequence.

OCP Commands

To execute / stop the test, use: INPUT command.

| | |
|-----------------------------------|-----|
| :OCP:STATe | 149 |
| :OCP:EDIT[:CHANnel] | 150 |
| :OCP[:CHANnel]:NUMBer | 151 |
| :OCP:MEMO | 151 |
| :OCP[:CHANnel]:RANGe..... | 152 |
| :OCP[:CHANnel]:STARt | 152 |
| :OCP[:CHANnel]:END | 153 |
| :OCP[:CHANnel]:STEP:CURRent | 153 |
| :OCP[:CHANnel]:STEP:TIME..... | 154 |
| :OCP[:CHANnel]:DELay..... | 154 |
| :OCP[:CHANnel]:TRIGger | 155 |
| :OCP[:CHANnel]:LAST | 155 |
| :OCP:CHANnel:STATus | 156 |
| :OCP:RESult | 156 |
| :OCP:RUN | 157 |

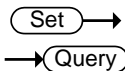
:OCP:STATe



| | | |
|------------------|---|---|
| Description | Sets and queries for the state of the OCP function. | |
| Syntax | :OCP:STATe {<Boolean> OFF ON} | |
| Query Syntax | :OCP:STATe? | |
| Parameter | ON / 1 | Turns the OCP function on |
| | OFF / 0 | Turns the OCP function off |
| Return Parameter | ON, STOP | OCP function is on. OCP test stop. |
| | ON, RUN | OCP function is on. OCP test is running. |
| | ON, END | OCP function is on. OCP test is finished. |
| | OFF | OCP function is off. |
| Return | Return the OCP function state. | |

example :OCP:STATe ON
Turns the OCP function on.

Query example :OCP:STATe?
OFF
Indicates that the OCP function is turned off



:OCP:EDIT[:CHANnel]

Description Sets and queries for the settings of the selected OCP memory number.
Parameters of "OCP Function"

Syntax :OCP:EDIT[:CHANnel]{<NR1>,LOW | MIDDle | HIGH,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>}

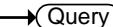
Query Syntax :OCP:EDIT?

| | | |
|-----------|---------------------|--|
| Parameter | <NR1> | Memory Number (OCP.No:1~12) |
| | LOW MIDDle HIGH | Current range (Range:) |
| | <NR2> | Start current value (Start C: unit A) |
| | <NR2> | End current value (End C: unit A) |
| | <NR2> | Current step value (Step C: unit A) |
| | <NR2> | Time step value (Step T:unit s) |
| | <NR2> | Delay time value (Delay:unit s) |
| | <NR2> | Trigger voltage (Trig V:unit V) |
| | <NR2> | Steady state current after test has finished (last C:unit A) |

Return Parameter Return all the settings for the selected OCP memory number.
Temp:<Seq_1~Seq_12>,Range:<High|Middle|Low>,
Start C:<NR2>, End C:<NR2>, Step C:<NR2>,Step
T:<NR2>, Delay: <NR2>, TrigV :<NR2>, Last C:<NR2>

example :OCP:EDIT 1, LOW, 0.5, 5.0, 0.1, 0.25, 0, 5.0, 1.0

Query example :OCP:EDIT?
 Temp:Seq_1, Range:High, Start C:0.36749, EndC:
 0.36750, Step C:0.00001, Step T:5.00, Delay:0.00, Trig
 V:0.01, Last C:0.00000

:OCP[:CHANnel]:NUMBer  

Description Sets and queries for the OCP memory number.
 (Contents of OCP.No:)

Syntax :OCP[:CHANnel]:NUMBer <NR1>

Query Syntax :OCP[:CHANnel]:NUMBer?

Parameter <NR1> Sets the OCP memory number

Return Returns the OCP memory number

example :OCP:NUMB 1
 Sets the OCP number is 1.

Query example :OCP:NUMB?
 1
 The OCP number is 1.

:OCP:MEMO  

Description Sets and queries for user-created note of the
 currently selected OCP function.

Setting syntax :OCP:MEMO <string>

Setting parameter <string> Set the OCP note.

Setting example :OCP:MEMO abc
 Set the OCP note is “abc”.

Query syntax :OCP:MEMO?

Return parameter <string> Return the OCP note.

Query example :OCP:MEMO?
 abc
 The OCP note is “abc”.

Set →
 → Query

:OCP[:CHANnel]:RANGe

| | | |
|---------------|---|-----------------------|
| Description | Sets and queries for the channel range. (Contents of Range:) | |
| Syntax | :OCP[:CHANnel]:RANGe {LOW MIDDLE HIGH} | |
| Query Syntax | :OCP[:CHANnel]:RANGe? | |
| Parameter | LOW | CC Mode Low range. |
| | MIDDLE | CC Mode Middle range. |
| | HIGH | CC Mode High range. |
| Return | Return the OCPtest current range. | |
| example | :OCP:RANG LOW Set CC Mode Low range. | |
| Query example | :OCP:RANG? Low The range is CC Mode Low range. | |

Set →
 → Query

:OCP[:CHANnel]:STARt

| | | |
|-------------------|--|---|
| Description | Sets and queries for the starting current value. (Contents of Start C:) | |
| Setting syntax | :OCP[:CHANnel]:STARt {<NR2> MINimum MAXimum} | |
| Setting parameter | <NR2> | The START current value in Amps. |
| | MINimum | Minimum starting current value. |
| | MAXimum | Maximum starting current value. |
| Setting example | :OCP:STAR 2 Set the start current to 2A. | |
| Query syntax | :OCP[:CHANnel]:STARt? | |
| Return parameter | <NR2> | Return the START current value in Amps. |

Query example :OCP:STAR?
0.1000
Returns the starting current as 0.1A.

Set →
 → Query

:OCP[:CHANnel]:END

Description Sets and queries for the ending current value of the test. The value must be higher than the DUT OCP value. (Contents of End C:)

Setting syntax :OCP[:CHANnel]:END
{<NR2>|MINimum|MAXimum}

| | | |
|-------------------|---------|--------------------------------|
| Setting parameter | <NR2> | The END current value in Amps. |
| | MINimum | Minimum ending current value. |
| | MAXimum | Maximum ending current value. |

Setting example :OCP:END 2
Set the END current to 2A.

Query syntax :OCP[:CHANnel]:END?

Return parameter <NR2> Return the END current value in Amps.

Query example :OCP:END?
0.1000
Returns the END current as 0.1A.

Set →
 → Query

:OCP[:CHANnel]:STEP:CURRent

Description Sets and queries for the current step resolution of the OCP Test Automation. (Contents of Step C:)

Setting syntax :OCP[:CHANnel]:STEP:CURRent
{<NR2>|MINimum|MAXimum}

| | | |
|-------------------|---------|----------------------------------|
| Setting parameter | <NR2> | The current value in Amps. |
| | MINimum | Minimum current step resolution. |
| | MAXimum | Maximum current step resolution. |

Setting example :OCP:STEP:CURR 0.1
 Set the step resolution as 0.1A.

Query syntax :OCP[:CHANnel]: STEP:CURRent?

Return parameter <NR2> Returns the current step resolution in Amps.

Query example :OCP:STEP:CURR?
 0.1000
 Returns the step resolution as 0.1A.

:OCP[:CHANnel]:STEP:TIME

Set →
 → Query

Description Sets and queries for how long the step times of the OCP Test Automation function. (Contents of Step T:)

Setting syntax :OCP[:CHANnel]:STEP:TIME
 {<NR2>|MINimum|MAXimum}

| | |
|-------------------------|---------------------------|
| Setting parameter <NR2> | The step time in seconds. |
| MINimum | Minimum step times. |
| MAXimum | Maximum step times. |

Setting example :OCP:STEP:TIME 2
 Set the step time to 2second.

Query syntax :OCP[:CHANnel]:STEP:TIME?

Return parameter <NR2> Return the step time in seconds.

Set →
 → Query

:OCP[:CHANnel]:DELay

Description Sets and queries for the test delay time of the OCP Test Automation function. (Contents of Delay:)

Setting syntax :OCP[:CHANnel]:DELay
 {<NR2>|MINimum|MAXimum}

| | |
|-------------------------|---------------------------|
| Setting parameter <NR2> | The delay time in seconds |
| MINimum | Minimum test delay time. |

MAXimum Maximum test delay time.

Setting example :OCP:DEL 2
Set the delay time to 2s.

Query syntax :OCP[:CHANnel]:DElAY?

Return parameter <NR2> Return the delay time in seconds.

Query example :OCP:DEL?
0.10
Returns the delay time in seconds

Set →
→ Query

:OCP[:CHANnel]:TRIGger

Description Sets and queries for the voltage trigger for when the power supply OCP has been triggered. (Contents of Trig V:)

Setting syntax :OCP[:CHANnel]:TRIGger
{<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The trigger voltage level.
MINimum Minimum voltages trigger level.
MAXimum Maximum voltage triggers level.

Setting example :OCP:TRIG 2
Set the trigger voltage level to 2V.

Query syntax :OCP[:CHANnel]:TRIGger?

Return parameter <NR2> Return the trigger voltage level.

Query example :OCP:TRIG?
2.0
Returns the trigger level.

Set →
→ Query

:OCP[:CHANnel]:LAST

Description Sets and queries for the current value of after the DUT OCP protection has been activated. (Contents of last C:)

Setting syntax :OCP[:CHANnel]:LAST
 {<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The current value in Amps.
 MINimum Minimum current value.
 MAXimum Maximum current value.

Setting example :OCP:LAST 2
 Set the current value to 2A.

Query syntax :OCP[:CHANnel]:LAST?

Return parameter <NR2> Returns the current value in Amps.

Query example :OCP:LAST?
 0.1000
 Returns the current value.

:OCP:CHANnel:STATus → Query

Description Queries the status of the OCP Test Automation function.

Query Syntax :OCP:CHANnel:STATus?

Return Parameter 0 Test ended
 1 OCP test active

Query example :OCP:CHAN:STAT?
 0
 The test has ended.

:OCP:RESult → Query

Description Returns the OCP Test Automation results.

Query Syntax :OCP:RESult?

Return Parameter <NR2>,<NR2> Returns the OCP current, voltage.

Query example :OCP:RES?
 3.6750,0.10
 OCP Current:3.675A, OCP Voltage:0.10V

:OCP:RUN**Set** →

Description Turns the load on for the OCP Test Automation function.

Same as :INPUT ON command.

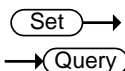
Syntax :OCP:RUN

OPP test Commands

To execute / stop the test, use: INPUT command.

| | |
|--------------------------------|-----|
| :OPP:STATe | 158 |
| :OPP:EDIT[:CHANnel] | 159 |
| :OPP[:CHANnel]:NUMBer | 160 |
| :OPP:MEMO | 160 |
| :OPP[:CHANnel]:RANGe | 161 |
| :OPP[:CHANnel]:STARt | 161 |
| :OPP[:CHANnel]:END | 162 |
| :OPP[:CHANnel]:STEP:WATT | 163 |
| :OPP[:CHANnel]:STEP:TIME | 163 |
| :OPP[:CHANnel]:DELay | 164 |
| :OPP[:CHANnel]:TRIGger | 164 |
| :OPP[:CHANnel]:LAST | 165 |
| :OPP:CHANnel:STATus | 166 |
| :OPP:RESult | 166 |
| :OPP:RUN | 166 |

:OPP:STATe



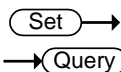
| | | |
|-------------------|---|---|
| Description | Sets and queries for the state of the OPP function. | |
| Setting syntax | :OPP:STATe {<bool> OFF ON} | |
| Setting parameter | OFF 0 | Turns the OPP function off. |
| | ON 1 | Turns the OPP function on. |
| Setting example | :OPP:STATe ON | |
| | Turns the OPP function on. | |
| Query syntax | :OPP:STATe? | |
| Return Parameter | Return the OPP function state. | |
| | ON, | OPP function is on. |
| | STOP RUN END | “STOP” is OPP test stop, “RUN” is OPP test running, “END” is OPP test finished. |
| | OFF | OPP function is off. |

Query example :OPP:STATE?

OFF

Indicates that the OPP function is turned off.

:OPP:EDIT[:CHANnel]



Description Sets and queries for the settings of the selected OPP memory number.

Parameters of "OPP Function"

Setting syntax :OPP:EDIT[:CHANnel](1),(2),(3),(4),(5),(6),(7),(8),(9)

| | | |
|-------------------|--------------------|---|
| Setting parameter | (1) <NR1> | Memory Number (OPP.No:1~12) |
| | (2) <ASCII string> | Current range (Range:) LOW MIDDLE HIGH |
| | (3) <NR2> | Start watt value (Start W: unit W) |
| | (4) <NR2> | End watt value (End W: unit W) |
| | (5) <NR2> | Watt step value (Step W: unit W) |
| | (6) <NR2> | Time step value (Step T:unit s) |
| | (7) <NR2> | Delay time value (Delay: unit s) |
| | (8) <NR2> | Trigger voltage (Trig V:unit V) |
| | (9) <NR2> | Steady state watt after test has finished (last W:unit W) |

Setting example :OPP:EDIT 1, LOW, 0.5, 5.0, 0.1, 0.25, 0, 5.0, 1.0

Query syntax :OPP:EDIT[:CHANnel]?

Return Parameter In the following order, return all the settings for the selected OPP memory number.

| | |
|--------------------------------------|-------------------|
| (1) Temp:Seq_<NR1> | OPP.No:1~12. |
| (2) Range: High Middle Low | Current range. |
| (3) Start W:<NR2> | Start watt value. |
| (4) End W:<NR2> | End watt value. |

| | |
|------------------|--|
| (5) Step W:<NR2> | Watt step value. |
| (6) Step T:<NR2> | Time step value. |
| (7) Delay: <NR2> | Delay time value. |
| (8) TrigV :<NR2> | Trigger voltage. |
| (9) Last W:<NR2> | Steady state watt after test has finished. |

Query example :OPP:EDIT?
 Temp:Seq_1, Range: High, Start W:0.36749, End W:0.36750, Step W:0.00001, Step T:5.00, Delay:0.00,Trig V:0.01, Last W:0.00000

Set →
 → Query

:OPP[:CHANnel]:NUMBer

Description Sets and queries for the OPP memory number. (Contents of OPP.No:)

Setting syntax :OPP[:CHANnel]:NUMBer <NR1>

Setting parameter <NR1> Sets the OPP memory number

Setting example :OPP:NUMB 1
 Sets the OPP number is 1.

Query syntax :OPP[:CHANnel]:NUMBer?

Return parameter <NR1> Returns the OPP memory number.

Query example :OPP:NUMB?
 1
 The OPP number is 1.

Set →
 → Query

:OPP:MEMO

Description Sets and queries for user-created note of the currently selected OPP function.

Setting syntax :OPP:MEMO <string>

Setting parameter <string> Set the OPP note.

Setting example :OPP:MEMO abc
 Set the OPP note is “abc”.

Query syntax :OCP:MEMO?
 Return parameter <string> Return the OCP note.
 Query example :OCP:MEMO?
 abc
 The OCP note is “abc”.

 →
 → 

:OPP[:CHANnel]:RANGe

Description Sets and queries for the channel range. (Contents of Range:)

Setting syntax :OPP[:CHANnel]:RANGe {LOW|MIDDLE|HIGH}

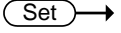
Setting parameter LOW CC Mode Low range.
 MIDDLE CC Mode Middle range.
 HIGH CC Mode High range.

Setting example :OPP:RANG LOW
 Set CP Mode Low range.

Query syntax :OPP[:CHANnel]:RANGe?

Return parameter LOW | MIDDLE | HIGH Return the OPP test current range.

Query example :OPP:RANG?
 Low
 The range is CP Mode Low range.

 →
 → 

:OPP[:CHANnel]:START

Description Sets and queries for the starting current value. (Contents of Start W:)

Setting syntax :OPP[:CHANnel]:START
 {<NR2>|MINimum|MAXimum}

| | | |
|-------------------|---------|--------------------------------|
| Setting parameter | <NR2> | The START watt value in Watts. |
| | MINimum | Minimum starting watt value. |
| | MAXimum | Maximum starting watt value. |

Setting example :OPP:STAR 2
 Set the start watt to 2W.

| | |
|------------------|--|
| Query syntax | :OPP[:CHANnel]:STARt? |
| Return parameter | <NR2> Return the START watt value in Watts. |
| Query example | :OPP:STAR? 0.1000 Returns the starting watt as 0.1W. |

Set →
 → Query

:OPP[:CHANnel]:END

Description Sets and queries for the ending watt value of the test. The value must be higher than the DUT OPP value. (Contents of End W:)

| | |
|-------------------|---|
| Setting syntax | :OPP[:CHANnel]:END {<NR2> MINimum MAXimum} |
| Setting parameter | <NR2> The END watt value in Amps. MINimum Minimum starting watt value. MAXimum Maximum starting watt value. |

Setting example :OPP:END 2
 Set the END watt to 2A.

| | |
|------------------|--|
| Query syntax | :OPP[:CHANnel]:END? |
| Return parameter | <NR2> Return the END watt value in Watts. |
| Query example | :OPP:END? 0.1000 Returns the END watt as 0.1W. |

:OPP[:CHANnel]:STEP:WATT




| | | |
|-------------------|--|--|
| Description | Sets and queries for the current step resolution of the OPP Test Automation. (Contents of Step W:) | |
| Setting syntax | :OPP[:CHANnel]:STEP:WATT {<NR2> MINimum MAXimum} | |
| Setting parameter | <NR2> | The watt value in Watts. |
| | MINimum | Minimum watt step resolution. |
| | MAXimum | Maximum watt step resolution. |
| Setting example | :OPP:STEP:WATT 0.1 Set the step resolution as 0.1W. | |
| Query syntax | :OPP[:CHANnel]:STEP:WATT? | |
| Return parameter | <NR2> | Returns the Watt step resolution in Watts. |
| Query example | :OPP:STEP:WATT? 0.1000 Returns the step resolution as 0.1W. | |

:OPP[:CHANnel]:STEP:TIME




| | | |
|-------------------|---|---------------------------|
| Description | Sets and queries for how long the step times of the OPP Test Automation function. (Contents of Step T:) | |
| Setting syntax | :OPP[:CHANnel]:STEP:TIME {<NR2> MINimum MAXimum} | |
| Setting parameter | <NR2> | The step time in seconds. |
| | MINimum | Minimum step times. |
| | MAXimum | Maximum step times. |
| Setting example | :OPP:STEP:TIME 2 Set the step time to 2second. | |
| Query syntax | :OPP[:CHANnel]:STEP:TIME? | |

Return parameter <NR2> Return the step time in seconds.

Query example :OPP:STEP:TIME?

0.10

Returns the step time.

:OPP[:CHANnel]:DELay

Set →

→ Query

Description Sets and queries for the test delay time of the OPP Test Automation function. (Contents of Delay:)

Setting syntax :OPP[:CHANnel]:DELay
{<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The delay time in seconds

MINimum Minimum test delay time.

MAXimum Maximum test delay time.

Setting example :OPP:DEL 2

Set the delay time to 2s.

Query syntax :OPP[:CHANnel]:DELay?

Return parameter <NR2> Return the delay time in seconds.

Query example :OPP:DEL?

0.10

Returns the delay time in seconds.

:OPP[:CHANnel]:TRIGger

Set →

→ Query

Description Sets and queries for the voltage trigger for when the power supply OPP has been triggered. (Contents of Trig V:)

Setting syntax :OPP[:CHANnel]:TRIGger
{<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The trigger voltage level.

MINimum Minimum voltages trigger level.

| | | |
|------------------|-------------------------|--------------------------------------|
| | MAXimum | Maximum voltage triggers level. |
| Setting example | :OPP:TRIG 2 | |
| | | Set the trigger voltage level to 2V. |
| Query syntax | :OPP[:CHANnel]:TRIGger? | |
| Return parameter | <NR2> | Return the trigger voltage level. |
| Query example | :OPP:TRIG? | |
| | 2.0 | |
| | | Returns the trigger level. |

Set →
 → Query

:OPP[:CHANnel]:LAST

| | | |
|-------------------|---|----------------------------------|
| Description | Sets and queries for the watt value of after the DUT OPP protection has been activated. (Contents of last W:) | |
| Setting syntax | :OPP[:CHANnel]:LAST {<NR2> MINimum MAXimum} | |
| Setting parameter | <NR2> | The watt value in Watts. |
| | MINimum | Minimum watt value. |
| | MAXimum | Maximum watt value. |
| Setting example | :OPP:LAST 2 | |
| | | Set the watt value to 2W. |
| Query syntax | :OPP[:CHANnel]:LAST? | |
| Return parameter | <NR2> | Returns the watt value in Watts. |
| Query example | :OPP:LAST? | |
| | 0.1000 | |
| | | Returns the watt value. |

:OPP:CHANnel:STATus → Query

| | |
|------------------|---|
| Description | Queries the status of the OPP Test Automation function. |
| Query syntax | :OPP:CHANnel:STATus? |
| Return parameter | 0 Test ended |
| | 1 OPP test active |
| Query example | :OPP:CHAN:STAT? 0 The test has ended. |

:OPP:RESult → Query

| | |
|------------------|---|
| Description | Returns the OPP Test Automation results. |
| Query syntax | :OPP:RESult? |
| Return parameter | <NR2>,<NR2> Returns the OPP current, voltage. |
| Query example | :OCP:RES? 3.6750,0.10 OPP Current:3.675A, OPP Voltage:0.10V |

:OPP:RUN Set →

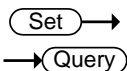
| | |
|----------------|---|
| Description | Turns the load on for the OPP Test Automation function. Same as :INPUT ON command. |
| Setting syntax | :OPP:RUN |

BATT test Commands


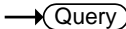
To execute / stop the test, use: INPUT command.

| | |
|---------------------------------|-----|
| :BATTery:STATe..... | 167 |
| :BATT:EDIT | 168 |
| :BATTery [:CHANnel]:NUMBer..... | 169 |
| :BATTery:MEMO..... | 170 |
| :BATTery:MODE..... | 170 |
| :BATTery:RANGe..... | 171 |
| :BATTery:VALue..... | 171 |
| :BATTery:RISE | 172 |
| :BATTery:FALL..... | 172 |
| :BATTery:STOP:VOLTage..... | 173 |
| :BATTery:STOP:TIME | 174 |
| :BATTery:STOP:AH..... | 174 |
| :BATTery:DATalog:TIMer | 175 |
| :BATT:CHANnel:STATus..... | 175 |
| :BATT:RESult..... | 176 |
| :BATT:RUN..... | 176 |

:BATTery:STATe



| | | |
|-------------------|--|--|
| Description | Sets and queries for the state of the BATT function. | |
| Setting syntax | :BATTery:STATe {<Boolean> OFF ON} | |
| Setting parameter | OFF 0 | Turns the BATT function off. |
| | ON 1 | Turns the BATT function on. |
| Setting example | :BATT:STATe ON Turns the BATT function on. | |
| Query syntax | :BATTery:STATe? | |
| Return Parameter | Return the BATT function state. | |
| | ON, STOP RUN END | BATT function is on. “STOP” is BATT test stop, “RUN” is BATT test running, “END” is BATT test finished. |

| | | |
|-------------------|---|---|
| | OFF | BATT function is off. |
| Query example | :BATT:STATE? | |
| | OFF | |
| | Indicates that the BAtt function is turned off. | |
| | |  |
| | |  |
| :BATT:EDIT | | |
| Description | Sets and queries for the settings of the selected BATT memory number. | |
| | Parameters of “BATT Function”. | |
| Setting syntax | :BATTery:EDIT(1),(2),(3),(4),(5),(6),(7),(8),(9),(10) | |
| Setting parameter | (1) <NR1> | Memory Number (BATT.No:1~12). |
| | (2) <ASCII string> | Operation mode (Mode :). CC CR CP |
| | (3) <ASCII string> | I and V range (Range :) ILVL IMVL IHVL ILVH IMVH IHVH |
| | (4) <NR2> | Setting value of operation mode (Setting :). |
| | (5) <NR2> | Rising slew rate (Slew Rate↑ : unit mA/us). |
| | (6) <NR2> | Falling slew rate (Slew Rate↓ : unit mA/us). |
| | (7) <NR2> | Stop Voltage (Stop Volt: unit V). |
| | (8) <NR2> | Stop Time (Stop Time: unit s). |
| | (9) <NR2> | Stop AH (Stop AH: unit Ah) |
| | (10) <NR2> | The time interval for data capture (Datalog timer: unit s). |
| Setting example | :BATT:EDIT 1, CC, ILVL, 0.1, 8.4, 8.4, 5, 50, 1 | |
| Query syntax | :BATTery:EDIT? | |

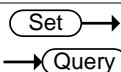
Return Parameter In the following order, return all the settings for the selected BATT memory number.

| | |
|---|-------------------------------------|
| (1) No:<NR1> | BATT.No:1~12. |
| (2) /sMemo:<string> | BATT note. |
| (3) /sMode:CC CR CP | Operation mode. |
| (4) /sRange: ILVL IMVL IHVL ILVH IMVH IHVH | I and V range |
| (5) /sSet:<NR2> | Setting value. |
| (6) /sSRUP :<NR2> | Rising slew rate. |
| (7) /sSRDW:<NR2> | Falling slew rate. |
| (8) /sStop/sV:<NR2> | Stop Voltage. |
| (9) /sStop/sT:<NR2> | Stop Time. |
| (10) /sStop/sAH :<NR2> | Stop AH |
| (11) /sDatalog:<NR2>/n | The time interval for data capture. |

Query example :BATT:EDIT?

No:01,/sMemo:./sMode:CC,/sRange:ILVL,/sSet:0.000
0,/sSRUP:250,/sSRDW:250,/sStop/sV:1.500,/sStop/s
T:0,/sStop/sAH:0.20,/sDatalog:1/n

:BATTery [:CHANnel]:NUMBER



Description Sets and queries for the BATT memory number.
(Contents of BATT.No:)

Setting syntax :BATTery[:CHANnel]:NUMBER <NR1 >

Setting parameter <NR1> Sets the BATT memory number

Setting example :BATT:NUMB 1
Sets the BATT number is 1.

Query syntax :BATTery[:CHANnel]:NUMBER?

Return parameter <NR1> Returns the BATT memory number.

Query example :BATT:NUMB?
 1
 The BATT number is 1.

Set →

:BATTery:MEMO

→ Query

Description Sets and queries for user-created note of the currently selected BATT function.

Setting syntax :BATTery:MEMO <string>

Setting parameter <string> Set the BATT note.

Setting example :BATT:MEMO abc
 Set the BATT note is "abc".

Query syntax :BATTery:MEMO?

Return parameter <string> Return the BATT note.

Query example :BATT:MEMO?
 abc
 The BATT note is "abc".

Set →

:BATTery:MODE

→ Query

Description Sets and queries for the operation mode.
 (Contents of Mode:)

Setting syntax :BATTery:MODE {CC|CR|CP}

Setting parameter CC Set the CC mode.

CR Set the CR mode.

CP Set the CP mode.

Setting example :BATT:MODE CC
 Set the operation mode to CC.

Query syntax :BATTery:MODE?

Return parameter CC | CR | CP Return the operation mode.

Query example : BATT:MODE?
 CC
 The operation mode is CC.

:BATTery:RANGe

Set →
 → Query

Description Sets and queries for the channel range. (Contents of Range:)

Setting syntax :BATTery:RANGe
 {ILVL|IMVL|IHVL|ILVH|IMVH|IHVH}

| | | |
|-------------------|------|---|
| Setting parameter | ILVL | Set the I range is L, and V range is L. |
| | IMVL | Set the I range is M, and V range is L. |
| | IHVL | Set the I range is H, and V range is L. |
| | ILVH | Set the I range is L, and V range is H. |
| | IMVH | Set the I range is M, and V range is H. |
| | IHVH | Set the I range is H, and V range is H. |

Setting example :BATT:RANG ILVL
 Set the range are I range L, and V range L.

Query syntax :BATTery:RANGe?

Return parameter ILVL|IMVL|IHVL|ILVH|IMVH|IHVH Return the BATT test I and V range.

Query example :BATT:RANG?
 ILVL
 The setting range is I range L, and V range L.

:BATTery:VALue

Set →
 → Query

Description Sets and queries for the setting value of the selected operation mode. (Contents of Setting:)

Setting syntax :BATTery:VALue {<NR2>|MINimum|MAXimum}

| | | |
|-------------------|---------|------------------------|
| Setting parameter | <NR2> | The setting value |
| | MINimum | Minimum setting value. |

MAXimum Maximum setting value.

Setting example :BATT:VAL 2
Set the value to 2A.

Query syntax :BATTery:VALue?

Return parameter <NR2> Return the setting value.

Query example :BATT:VAL?
2
Returns the starting current as 2A.

:BATTery:RISE

Set →

→ Query

Description Sets and queries for the test rising slew rate in mA/us. (Contents of Slew Rate↑:)

Setting syntax :BATTery:RISE {<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The test rising slew rate.

MINimum Minimum test rising slew rate.

MAXimum Maximum test rising slew rate.

Setting example :BATT:RISE 8.4
Set the test rising slew to 8.4mA/us.

Query syntax :BATTery:RISE?

Return parameter <NR2> Return the test rising slew.

Query example :BATT:RISE?
8.4
Returns the test rising slew as 8.4mA/us.

Set →

→ Query

:BATTery:FALL

Description Sets and queries for the test falling slew rate in mA/us. (Contents of Slew Rate↓:)

Setting syntax :BATTery:FALL {<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The test falling slew rate.

| | |
|---------|---------------------------------|
| MINimum | Minimum test falling slew rate. |
|---------|---------------------------------|

| | |
|---------|---------------------------------|
| MAXimum | Maximum test falling slew rate. |
|---------|---------------------------------|

Setting example :BATT:FALL 8.4

Set the test falling slew to 8.4mA/us.


Query syntax :BATTery:FALL?

Return parameter <NR2> Return the test falling slew.

Query example :BATT:FALL?

8.4

Returns the test falling slew as 8.4mA/us.

:BATTery:STOP:VOLTage



Description Sets and queries for the voltage at which the test should be interrupted. (Contents of Stop Volt:)

Setting syntax :BATTery:STOP:VOLTage
{<NR2>|MINimum|MAXimum}

Setting parameter <NR2> The stop voltage in V.

| | |
|---------|-----------------------|
| MINimum | Minimum stop voltage. |
|---------|-----------------------|

| | |
|---------|-----------------------|
| MAXimum | Maximum stop voltage. |
|---------|-----------------------|

Setting example :BATT:STOP:VOLT 5

Set the stop voltage to 5V.

Query syntax :BATTery:STOP:VOLTage?

Return parameter <NR2> Return the stop voltage in V.

Query example :BATT:STOP:VOLT?

5

Returns the stop voltage as 5V.

Set →

:BATTery:STOP:TIME

→ Query

Description Sets and queries for the time after which the test should be interrupted. (Contents of Stop Time:)

Setting syntax :BATTery:STOP:TIME
{<NR1>|MINimum|MAXimum|OFF}

| | |
|--------------------------------|--|
| Setting parameter <NR1> | The stop time in seconds. (0~3599999) |
| MINimum | Minimum stop times. |
| MAXimum | Maximum stop times. |
| OFF | “OFF” is 0 second. |

Setting example :BATT:STOP:TIME 2
Set the stop time to 2second.

Query syntax :BATTery:STOP:TIME?

Return parameter <NR1> Return the stop time in seconds.

Query example :OPP:STOP:TIME?
2
Returns the step time as 2 seconds.

Set →

:BATTery:STOP:AH

→ Query

Description Sets and queries for the discharged energy rate at which the test should be interrupted. (Contents of Stop AH:)

Setting syntax :BATTery:STOP:AH
{<NR2>|MINimum|MAXimum|OFF}

| | |
|--------------------------------|-----------------------------------|
| Setting parameter <NR2> | The stop AH in Ah. (0~9999.99) |
| MINimum | Minimum stop AH. |
| MAXimum | Maximum stop AH. |
| OFF | “OFF” is 0 AH. |

Setting example :BATT:STOP:AH 2
 Set the stop AH to 2 Ah.

Query syntax :BATTery:STOP:AH?

Return parameter <NR2> Return the stop AH in Ah.

Query example :OPP:STOP:AH?
 2
 Returns the step AH as 2 Ah.

Set →
 → Query

:BATTery:DATalog:TIMer

Description Sets and queries for the time interval for data capture. (Contents of Datalog timer:)

Setting syntax :BATTery:DATalog:TIMer
 {<NR1>|MINimum|MAXimum}

Setting parameter <NR1> The interval time in seconds.
 (1 ~ 120)

| | |
|---------|-------------------------|
| MINimum | Minimum interval times. |
| MAXimum | Maximum interval times. |

Setting example :BATT:DAT:TIM 2
 Set the interval time to 2 seconds.

Query syntax :BATTery:DATalog:TIMer?

Return parameter <NR1> Return the interval time in seconds.

Query example :OPP:DAT:TIM?
 2
 Returns the interval time as 2 seconds.

:BATT:CHANnel:STATus → Query

Description Queries the status of the BATT Test Automation function.

Query syntax :BATT:CHANnel:STATus?

Return parameter 0 Test ended

1 BATT test active

Query example :BATT:CHAN:STAT?

0

The test has ended.

:BATT:RESult

→ Query

Description Returns the BATT Test Automation results.

Query syntax :BATT:RESult?

Return parameter <NR2>,<NR2> Returns the BATT current, voltage.

Query example :BATT:RES?

3.6750,0.10

BATT Current:3.675A, BATT Voltage:0.10V

:BATT:RUN

Set →

Description Turns the load on for the BATT Test Automation function.

Same as :INPUT ON command.

Setting syntax :BATT:RUN

Utility Commands

| | |
|----------------------------|-----|
| :UTILity:SYSTem | 177 |
| :UTILity:LOAD | 178 |
| :UTILity:LOAD:MODE | 178 |
| :UTILity:LOAD:RANGe | 179 |
| :UTILity:TIME | 179 |
| :UTILity:KNOB | 180 |
| :UTILity:SPEAker | 181 |
| :UTILity:ALARm | 181 |
| :UTILity:UNReg | 182 |
| :UTILity:GNG | 183 |
| :UTILity:CONTRast | 183 |
| :UTILity:BRiGhtness | 184 |
| :UTILity:LANGuage | 184 |
| :UTILity:REMOte | 185 |
| :UTILity:REMOte:MODE | 185 |

:UTILity:SYSTem

→ **Query**

| | | |
|------------------|---|------------------------------|
| Description | Query for model number, serial number, and firmware version. | |
| Query Syntax | :UTILity:SYSTem? | |
| Return parameter | <ASCII string> | Return the model name. |
| | <NR1> | Return the serial number. |
| | <ASCII string> | Return the firmware version. |
| Query example | :UTIL:SYST? >PEL-3021H,12345678,V1.01.001 Return the model number, serial number, and firmware version. | |

Set →

→ Query

:UTILity:LOAD

Description Sets and queries for Auto Load and load function at power on. (Contents of Auto Load and Auto Load On)

Parameters of “Load Setting For Power On”

Syntax :UTILity:LOAD {ON | OFF | LOAD | PROG | NSEQ | FSEQ}

Query Syntax :UTILity:LOAD?

| | | |
|------------------|------|-----------------------------------|
| Parameter | ON | Turn on Auto Load. |
| | OFF | Turn off Auto Load. |
| | LOAD | Sets to manual operation. |
| | PROG | Sets to program function. |
| | NSEQ | Sets to normal sequence function. |
| | FSEQ | Sets to fast sequence function. |

Return parameter <ASCII string> Return the Auto Load and load function at power on, by the “Load:{On | Off}, Load On:{Load | Prog | NSeq | FSeq}” string.

Example 1 :UTIL:LOAD ON
Sets on the manual operation at power on.

Example 2 :UTIL:LOAD PROG
Sets to the program function at power on.

Query example :UTIL:LOAD?
>Load:Off, Load On:Prog
Return the Auto Load and load function at power on.

Set →

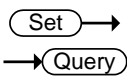
→ Query

:UTILity:LOAD:MODE

Description Sets and queries for the Load Off (Mode) setting.
Parameters of “Load Setting For Power On”

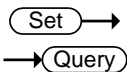
Syntax :UTILity:LOAD:MODE {< Boolean > | ON | OFF}

| | | |
|------------------|---|------------------------------|
| Query Syntax | :UTILity:LOAD:MODE? | |
| Parameter | ON / 1 | Sets Load Off (Mode) to ON. |
| | OFF / 0 | Sets Load Off (Mode) to OFF. |
| Return parameter | Return the Load Off (Mode) setting. | |
| Example | :UTIL:LOAD:MODE ON Turns Load Off (Mode) to on. | |
| Query example | :UTIL:LOAD:MODE? >Off Load Off (Mode) is off. | |



:UTILity:LOAD:RANGe

| | | |
|------------------|---|-------------------------------|
| Description | Sets and queries for the Load Off (Range) setting. Parameters of "Load Setting For Power On" | |
| Syntax | :UTILity:LOAD:RANGe {<Boolean> OFF ON} | |
| Query Syntax | :UTILity:LOAD:RANGe? | |
| Parameter | ON / 1 | Sets Load Off (Range) to OFF. |
| | OFF / 0 | Sets Load Off (Range) to ON. |
| Return parameter | Return the Load Off (Range) setting. | |
| Example | :UTILity:LOAD:RANGe ON Turns Load Off (Range) to on. | |
| Query example | :UTILity:LOAD:RANGe? > Off Load Off (Range) is off. | |



:UTILity:TIME

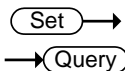
| | | |
|--------------|--|--|
| Description | Sets and queries for the date and time. Parameters of "Date/Time" | |
| Syntax | :UTILity:TIME (1),(2),(3),(4),(5) | |
| Query Syntax | :UTILity:TIME? | |

| | | |
|-----------|-----------|-------------------|
| Parameter | (1) <NR1> | Sets the month. |
| | (2) <NR1> | Sets the day. |
| | (3) <NR1> | Sets the year. |
| | (4) <NR1> | Sets the hour. |
| | (5) <NR1> | Sets the minutes. |

| | | |
|------------------|-----------|---------------------|
| Return parameter | (1) <NR1> | Return the month. |
| | (2) <NR1> | Return the day. |
| | (3) <NR1> | Return the year. |
| | (4) <NR1> | Return the hour. |
| | (5) <NR1> | Return the minutes. |

Example :UTIL:TIME 9,1,2013,10,11
Sets the date and time.

Query example :UTIL:TIME?
>Month:9, Day:1, Year:2013, Hour:10, Minute:11
Return the date and time.



:UTILity:KNOB

Description Sets and queries for operational setting of the knob.
Parameters of "Other Setting"

Syntax :UTILity:KNOB { UPDated | OLD }

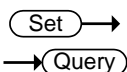
Query Syntax :UTILity:KNOB?

| | | |
|-----------|---------|--|
| Parameter | UPDated | Sets real-time updates. |
| | OLD | Sets update at the settlement after Enter. |

Return parameter Return the operational settings of the knob, by the "{Updated | Old}" string.

Example :UTIL:KNOB UPD
Sets the operation of the knob.

Query example :UTIL:KNOB?
 >Updated
 Return the operational settings of the knob.



:UTILity:SPEAker

Description Sets and queries for the speakers sound during scrolling and key input.

Parameters of “Other Setting”

Syntax :UTILity:SPEAker {< Boolean > | ON | OFF}

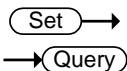
Query Syntax :UTILity: SPEAker?

| | | |
|-----------|----------|---|
| Parameter | ON or 1 | Sets on the speakers sound during scrolling and key input. |
| | OFF or 0 | Sets off the speakers sound during scrolling and key input. |

Return parameter Return the setting value of speaker sound during scrolling and key input, by the "{On | Off}" string.

Example :UTIL:SPEA ON
 Sets on the speakers sound during scrolling and key input.

Query example :UTIL:SPEA?
 >On
 Return the setting value of speaker sound during scrolling and key input.



:UTILity:ALARm

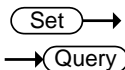
Description Sets and queries for the speaker sound of the alarm.

Parameters of “Other Setting”

Syntax :UTILity:ALARm {< Boolean > | ON | OFF}


Query Syntax :UTILity:ALARm?

| | | |
|------------------|--|---|
| Parameter | ON or 1 | Sets ON of the speaker sound of the alarm. |
| | OFF or 0 | Sets OFF of the speaker sound of the alarm. |
| Return parameter | Return the speaker sound of the alarm, by the "{On Off}" string. | |
| Example | :UTIL:ALAR ON Sets the speaker sound of the alarm. | |
| Query example | :UTIL:ALAR? >On Return the speaker sound of the alarm. | |




:UTILity:UNReg

| | | |
|------------------|---|--|
| Description | Sets and queries for the speaker sound of Anne-regulation. Parameters of "Other Setting" | |
| Syntax | :UTILity:UNReg {< Boolean > ON OFF} | |
| Query Syntax | :UTILity:UNReg? | |
| Parameter | ON or 1 | Sets on the speaker sound of Anne-regulation. |
| | OFF or 0 | Sets off the speaker sound of Anne-regulation. |
| Return parameter | Return the setting value of speaker sound of Anne-regulation, by the "{On Off}" string. | |
| Example | :UTIL:UNR ON Sets on the speaker sound of Unregulation. | |
| Query example | :UTIL:UNR? >On Return the setting value of speaker sound of Unregulation. | |

:UTILity:GNG




| | | |
|------------------|---|---|
| Description | Sets and queries for the speaker sound of the "Go-NoGo" judgment time. Parameters of "Other Setting" | |
| Syntax | :UTILity:GNG {< Boolean > ON OFF} | |
| Query Syntax | :UTILity:GNG? | |
| Parameter | ON or 1 | Sets ON of the speaker sound of the "Go-NoGo" judgment time. |
| | OFF or 0 | Sets OFF of the speaker sound of the "Go-NoGo" judgment time. |
| Return parameter | Return the speaker sound of the "Go-NoGo" judgment time, by the "{On Off}" string. | |
| Example | :UTIL:GNG ON Sets the speaker sound of the "Go-NoGo" judgment time. | |
| Query example | :UTIL:GNG? >On Return the setting of the speaker sound of the "Go-NoGo" judgment time. | |

:UTILity:CONTRast




| | | |
|--------------|--|--|
| Description | Sets and queries for the contrast of the LCD display. Parameters of "Other Setting" | |
| Syntax | :UTILity:CONTRast <NR1> | |
| Query Syntax | :UTILity:CONTRast? | |
| Parameter | <NR1> | Sets contrast of the LCD display. 3~13 (low ~ high) |
| Example | :UTIL:CONT 8 Sets the 8 at contrast of the LCD display. | |

Query example :UTIL:CONT?
 >8
 Return contrast of the LCD display.

Set →

:UTILity:BRiGhtness

→ Query

Description Sets and queries for brightness of the LCD display.

Parameters of "Other Setting"

Syntax :UTILity:BRiGhtness <NR1>

Query Syntax :UTILity:BRiGhtness?

Parameter <NR1> Sets brightness of the LCD display.
 50~90 (low ~ high)

Example :UTIL:BRI 70
 Sets the 70 at brightness of the LCD display.

Query example :UTIL:BRI?
 >70
 Return brightness of the LCD display.

Set →

:UTILity:LANGuage

→ Query

Description Sets and queries for the language of the operation panel. Note: Language is English only.

Parameters of "Other Setting"

Syntax :UTILity:LANGuage ENGLISH

Query Syntax :UTILity:LANGuage?

Parameter ENGLISH Sets to English language.

Return parameter Return the language of the operation panel, by the "English" string.

Example :UTIL:LANG ENGL
 Sets to English language.

Query example :UTIL:LANG?
 >English
 Return the language.

:UTILity:REMOte



Description Turns the remote control on or off.

Syntax :UTILity:REMOte {OFF | 0 | ON | 1}

| | | |
|-----------|-------|--------------------------|
| Parameter | OFF/0 | Turns Remote control off |
| | ON/1 | Turns remote control on |

Example :UTIL:REM 1
 Turns remote control on.

:UTILity:REMOte:MODE



Description Sets the remote mode to fast or normal.
 When in fast mode, the panel interface is deactivated with an interface time of no more than 10ms. Normal mode has an interface time of 30~130ms. In normal mode the display interface continues to update the screen in real-time.

Syntax :UTILity:REMOte:MODE {NORMAL | 0 | FAST | 1}

| | | |
|-----------|----------|--------|
| Parameter | FAST/1 | FAST |
| | NORMAL/0 | NORMAL |

Example :UTIL:REM:MODE 1
 Turns remote mode to fast.

Interface Commands


| | |
|-------------------------|-----|
| :UTILity:INTerface..... | 186 |
| :UTILity:BRATe..... | 186 |
| :UTILity:SBIT | 187 |
| :UTILity:PARity | 187 |

:UTILity:INTerface

Set →

→ Query

Description Sets and queries for the interface.

 **Note** Command is only valid for RS-232Control and USB.

Syntax :UTILity:INTerface {USB | RS232}

Query Syntax :UTILity:INTerface?

Parameter Sets after the transmission, but require power cycle to enable the feature.

USB Sets the USB interface.

RS232 Sets the RS-232C interface.

Example :UTIL:INT RS232
Sets the RS-232C interface.


Query example :UTIL:INT?
>RS232
Return the interface setting value.

Set →

→ Query

:UTILity:BRATe

Description Sets and queries for the baud rate of RS-232C.

 **Note** Command is only valid for interface setting of RS-232C

Syntax :UTILity:BRATe {2400 | 4800 | 9600 | 19200 | 38400}

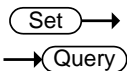
Query Syntax :UTILity:BRATe?

Parameter <NR1> Sets the baud rate.

| | | |
|------------------|--------------------------|-----------------------|
| Return parameter | <code><NR1></code> | Return the baud rate. |
|------------------|--------------------------|-----------------------|

| | | |
|---------|------------------|---------------------|
| Example | :UTIL:BRAT 38400 | Sets the baud rate. |
|---------|------------------|---------------------|

| | | |
|---------------|-----------------------|-----------------------|
| Query example | :UTIL:BRAT? >38400 | Return the baud rate. |
|---------------|-----------------------|-----------------------|



:UTILity:SBIT

| | |
|-------------|---|
| Description | Sets and queries for the stop bit of the RS-232C interface. |
|-------------|---|

| | |
|------|---|
| Note | Command is only valid for interface setting of RS232C |
|------|---|

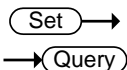
| | |
|--------|-----------------------|
| Syntax | :UTILity:SBIT {1 2} |
|--------|-----------------------|

| | |
|--------------|----------------|
| Query Syntax | :UTILity:SBIT? |
|--------------|----------------|

| | | |
|-----------|--------------------------|--------------------|
| Parameter | <code><NR1></code> | Sets the stop bit. |
|-----------|--------------------------|--------------------|

| | | |
|---------|--------------|--------------------|
| Example | :UTIL:SBIT 1 | Sets the stop bit. |
|---------|--------------|--------------------|

| | | |
|---------------|-------------------|----------------------|
| Query example | :UTIL:SBIT? >1 | Return the stop bit. |
|---------------|-------------------|----------------------|



:UTILity:PARity

| | |
|-------------|---|
| Description | Sets and queries for the parity bit of RS-232C interface. |
|-------------|---|

| | |
|------|--|
| Note | Command is only valid for interface setting of RS-232C |
|------|--|

| | |
|--------|---------------------------------------|
| Syntax | :UTILity:PARity { NONE ODD EVEN } |
|--------|---------------------------------------|

| | |
|--------------|------------------|
| Query Syntax | :UTILity:PARity? |
|--------------|------------------|

| | | |
|-----------|------|----------------------|
| Parameter | NONE | Sets no parity. |
| | ODD | Sets the odd parity. |

| | | |
|------------------|-------------|--|
| | EVEN | Sets the even parity. |
| Return parameter | | Return the parity bit of RS-232C interface, by the "{None Odd Even}" string. |
| Example | | :UTIL:PAR NONE Sets no parity. |
| Query example | | :UTIL:PAR? >None Return the setting of parity. |

File Commands

| | |
|-----------------------------|-----|
| :MEMory:SAVE..... | 189 |
| :MEMory:RECall..... | 189 |
| :PREset:SAVE..... | 190 |
| :PREset:RECall..... | 190 |
| :SETup:SAVE..... | 190 |
| :SETup:RECall..... | 190 |
| :FACTory[:RECall]..... | 191 |
| :USER[:DEFault]:SAVE..... | 191 |
| :USER[:DEFault]:RECall..... | 191 |

:MEMory:SAVE

Set →

Description Save in the internal memory of the specified.

Syntax :MEMory:SAVE <NR1>

Parameter <NR1> Specify the internal memory number.
(1 to 256)

Example :MEM:SAVE 20
Saves the current setting to internal memory 20.

Same function command *SAV

:MEMory:RECall

Set →

Description Recall settings from the internal memory.

Syntax :MEMory:RECall <NR1>

Parameter <NR1> Specify the internal memory number.
(1 to 256)

Example :MEM:REC 20
Recall setting internal memory 20.

Same function command *RCL

:PREset:SAVE (Set) →

| | | |
|-------------|--|--|
| Description | Save to the preset memory of the specified. | |
| Syntax | :PREset:SAVE < NR1 > | |
| Parameter | <NR1> | Specify the preset memory number. (0 to 9) |
| Example | :PRE:SAVE 1 Saves the setting value to preset memory 1. | |

:PREset:RECall (Set) →

| | | |
|-------------|---|--|
| Description | Recall settings from the preset memory. | |
| Syntax | :PREset:RECall <NR1 > | |
| Parameter | <NR1> | Specify the preset memory number. (0 to 9) |
| Example | :PRE:REC 1 Recall setting preset memory 1. | |

:SETup:SAVE (Set) →

| | | |
|-------------|---|---|
| Description | Save to the setup data of the specified. | |
| Syntax | :SETup:SAVE < NR1 > | |
| Parameter | <NR1> | Specify the setup data number. (1 to 100) |
| Example | :SET:SAVE 1 Saves the setting value to setup data 1. | |


:SETup:RECall (Set) →

| | | |
|-------------|--------------------------------------|---|
| Description | Recall settings from the setup data. | |
| Syntax | :SETup:RECall <NR1 > | |
| Parameter | <NR1> | Specify the setup data number. (1 to 100) |

Example :SET:REC 1
 Recall setting setup data 1.

:FACTory[:RECall] Set →

Description Sets factory defaults.

 Note Interface will switch to as follows. Interface:RS-232C,
 Baud rate:38400, Stop bit:1, parity :None

Syntax :FACTory[:RECall]

Example :FACT
 Sets factory defaults.

:USER[:DEFault]:SAVE Set →

Description Save to the default settings for the user.

Syntax :USER[:DEFault]:SAVE

Example :USER:SAVE
 Save as user default settings the current settings
 value.

:USER[:DEFault]:RECall Set →

Description Recall the default settings for the user.

Syntax :USER[:DEFault]:RECall

Example :USER:REC
 Recall the default settings for the user.

SCPI Status Commands

| | |
|----------------------|-----|
| :SYSTem:ERRor..... | 192 |
| :STATus:PRESet | 192 |

:SYSTem:ERRor → Query

Description Queries the error queue. The last error message is return. A maximum of 32 errors are stored in the error queue.

Query Syntax :SYSTem:ERRor?

Return parameter Returns the error queue, by the "<NR1>,<string>" string.

Query example :SYST:ERR?
 >-113, "Undefined header"

:STATus:PRESet Set →

Description Is the setting of the initial value for the Csummary status and the Questionable status and the Operation status. PTR (positive transition) filter is set, enable register and (negative transition) NTR filter is reset.

| Initial value of Register / Filter | Setting |
|--|---------|
| Csummary status enable | 0x0000 |
| Csummary status PTR (positive transition) | 0x7FFF |
| Csummary status NTR (negative transition) | 0x0000 |
| Operation status enable | 0x0000 |
| Operation status PTR (positive transition) | 0x7FFF |
| Operation status NTR (negative transition) | 0x0000 |
| Questionable status enable | 0x0000 |

| | | |
|---------|---|--------|
| | Questionable status PTR (positive transition) | 0x7FFF |
| | Questionable status NTR (negative transition) | 0x0000 |
| Syntax | :STATus:PRESet | |
| Example | :STAT:PRES | |

It set the initial value for the Csummary status and the Questionable status and the Operation status.

Csummary Status Commands

| | |
|-----------------------------------|-----|
| :STATus:CSUMmary:CONDition | 194 |
| :STATus:CSUMmary:ENABle | 194 |
| :STATus:CSUMmary[:EVENT] | 195 |
| :STATus:CSUMmary:NTRansition..... | 195 |
| :STATus:CSUMmary:PTRansition..... | 195 |

:STATus:CSUMmary:CONDition → Query

| | |
|------------------|---|
| Description | Query the Csummary Status Condition register. |
| Query Syntax | :STATus:CSUMmary:CONDition? |
| Return parameter | Return the Csummary Status Condition register, by the "<NR1>" string. |
| Query example | :STAT:CSUM:COND? >1 Return the Csummary Status Condition register |

Set →

:STATus:CSUMmary:ENABle → Query

| | |
|---------------|---|
| Description | Sets and queries for the Event Enable register of Csummary. |
| Syntax | :STATus:CSUMmary:ENABle <NR1> |
| Query Syntax | :STATus:CSUMmary:ENABle? |
| Parameter | <NR1> Sets the Event Enable register of Csummary. |
| Example | :STAT:CSUM:ENAB 1 Sets the Event Enable register of Csummary. |
| Query example | :STAT:CSUM:ENAB? >1 Return the Event Enable register of Csummary. |

:STATus:CSUMmary[:EVENT] → Query

| | |
|------------------|---|
| Description | Query and setting for the Event register of Csummary. |
| Query Syntax | :STATus:CSUMmary[:EVENT]? |
| Return parameter | Return the Event register of Csummary, by the "<NR1>" string. |
| Query example | :STAT:CSUM? >1 Return the Event register of Csummary. |

Set →

:STATus:CSUMmary:NTRansition → Query

| | |
|---------------|--|
| Description | Sets and queries for detection bit of Csummary status of changes of from positive to negative. |
| Syntax | :STATus:CSUMmary:NTRansition <NR1> |
| Query Syntax | :STATus:CSUMmary:NTRansition? |
| Parameter | <NR1> Sets the detection bit of Csummary status of changes from positive to negative. |
| Example | :STAT:CSUM:NTR 1 Sets the detection bit of Csummary status of changes from positive to negative. |
| Query example | :STAT:CSUM:NTR? >1 Return the detection bit of Csummary status of changes from positive to negative. |

Set →

:STATus:CSUMmary:PTRansition → Query

| | |
|-------------|--|
| Description | Sets and queries for detection bit of Csummary status of changes of from negative to positive. |
| Syntax | :STATus:CSUMmary:PTRansition <NR1> |

| | | |
|---------------|--|---|
| Query Syntax | :STATus:CSUMmary:PTRansition? | |
| Parameter | <NR1> | Sets the detection bit of Csummary status of changes from negative to positive. |
| Example | :STAT:CSUM:PTR 1 Sets the detection bit of Csummary status of changes from negative to positive. | |
| Query example | :STAT:CSUM:PTR? >1 Return the detection bit of Csummary status of changes from negative to positive. | |

Operation Status Commands

| | |
|-------------------------------------|-----|
| :STATus:OPERation:CONDition | 197 |
| :STATus:OPERation:ENABle | 197 |
| :STATus:OPERation[:EVENT] | 198 |
| :STATus:OPERation:NTRansition | 198 |
| :STATus:OPERation:PTRansition | 198 |

:STATus:OPERation:CONDition → **Query**

| | |
|------------------|---|
| Description | Query the Operation Status Condition register. |
| Query Syntax | :STATus:OPERation:CONDition? |
| Return parameter | Return the Operation Status Condition register, by the "<NR1>" string. |
| Query example | :STAT:OPER:COND? >1 Return the Operation Status Condition register. |

→ **Set**

:STATus:OPERation:ENABle → **Query**

| | |
|---------------|--|
| Description | Sets and queries for the Event Enable register of Operation. |
| Syntax | :STATus:OPERation:ENABle <NR1> |
| Query Syntax | :STATus:OPERation:ENABle? |
| Parameter | <NR1> Sets the Event Enable register of Operation. |
| Example | :STAT:OPER:ENAB 1 Sets the Event Enable register of Operation. |
| Query example | :STAT:OPER:ENAB? >1 Return the Event Enable register of Operation. |

:STATus:OPERation[:EVENT] → Query

| | |
|------------------|--|
| Description | Query for the Event register of Operation. |
| Query Syntax | :STATus:OPERation[:EVENT]? |
| Return parameter | Return the Event register of Operation, by the "<NR1>" string. |
| Query example | :STAT:OPER? >1 Return the Event register of Operation. |

Set →

:STATus:OPERation:NTRansition → Query

| | |
|---------------|---|
| Description | Sets and queries for detection bit of Operation status of changes of from positive to negative. |
| Syntax | :STATus OPERation:NTRansition <NR1> |
| Query Syntax | :STATus:OPERation:NTRansition? |
| Parameter | <NR1> Sets the detection bit of Operation status of changes from positive to negative. |
| Example | :STAT:OPER:NTR 1 Sets the detection bit of Operation status of changes from positive to negative. |
| Query example | :STAT:OPER:NTR? >1 Return the detection bit of Operation status of changes from positive to negative. |

Set →

:STATus:OPERation:PTRansition → Query

| | |
|--------------|---|
| Description | Sets and queries for detection bit of Operation status of changes of from negative to positive. |
| Syntax | :STATus:OPERation:PTRansition <NR1> |
| Query Syntax | :STATus OPERation:PTRansition? |

| | | |
|---------------|-----------------------|--|
| Parameter | <NR1> | Sets the detection bit of Operation status of changes from negative to positive. |
| Example | :STAT:OPER:PTR 1 | Sets the detection bit of Operation status of changes from negative to positive. |
| Query example | :STAT:OPER:PTR? >1 | Return the detection bit of Operation status of changes from negative to positive. |

Questionable Status Commands

| | |
|--|-----|
| :STATus:QUEStionable:CONDition | 200 |
| :STATus:QUEStionable:ENABle | 200 |
| :STATus:QUEStionable[:EVENT] | 201 |
| :STATus:QUEStionable:NTRansition | 201 |
| :STATus:QUEStionable:PTRansition | 201 |

:STATus:QUEStionable:CONDition → Query

Description Query the Questionable Status Condition register.

Query Syntax :STATus:QUEStionable:CONDition?

Return parameter Return the Questionable Status Condition register, by the "<NR1>" string.

Query example :STAT: QUES:COND?
>1

Return the Questionable Status Condition register.

Set →

:STATus:QUEStionable:ENABle → Query

Description Sets and queries for the Event Enable register of Questionable.

Syntax :STATus:QUEStionable:ENABle <NR1>

Query Syntax :STATus:QUEStionable:ENABle?

Parameter <NR1> Sets the Event Enable register of Questionable.

Example :STAT:QUES:ENAB 1
Sets the Event Enable register of Questionable.

Query example :STAT:QUES:ENAB?
>1
Return the Event Enable register of Questionable.

:STATus:QUEStionable[:EVENT] → Query

| | |
|------------------|---|
| Description | Query for the Event register of Questionable. |
| Query Syntax | :STATus:QUEStionable[:EVENT]? |
| Return parameter | Return the Event register of Questionable, by the "<NR1>" string. |
| Query example | :STAT:QUES? >1 Return the Event register of Questionable. |

:STATus:QUEStionable:NTRansition Set →
→ Query

| | |
|---------------|--|
| Description | Sets and queries for detection bit of Questionable status of changes of from positive to negative. |
| Syntax | :STATus:QUEStionable:NTRansition <NR1> |
| Query Syntax | :STATus:QUEStionable:NTRansition? |
| Parameter | <NR1> Sets the detection bit of Questionable status of changes from positive to negative. |
| Example | :STAT:QUES:NTR 1 Sets the detection bit of Questionable status of changes from positive to negative. |
| Query example | :STAT:QUES:NTR? >1 Return the detection bit of Questionable status of changes from positive to negative. |

:STATus:QUEStionable:PTRansition Set →
→ Query

| | |
|--------------|--|
| Description | Sets and queries for detection bit of Questionable status of changes of from negative to positive. |
| Syntax | :STATus:QUEStionable:PTRansition <NR1> |
| Query Syntax | :STATus:QUEStionable:PTRansition? |

| | | |
|---------------|-----------------------|---|
| Parameter | <NR1> | Sets the detection bit of Questionable status of changes from negative to positive. |
| Example | :STAT:QUES:PTR 1 | Sets the detection bit of Questionable status of changes from negative to positive. |
| Query example | :STAT:QUES:PTR? >1 | Return the detection bit of Questionable status of changes from negative to positive. |

Status Register Overview

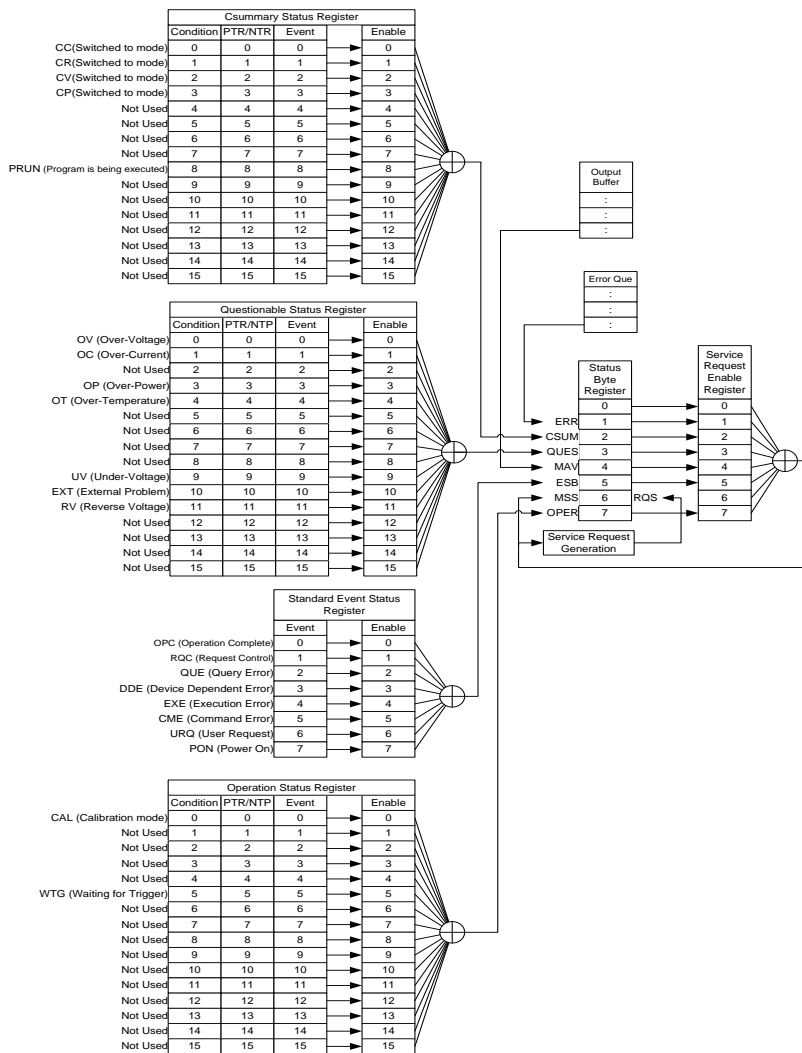
This section explains in detail how the Status registers are used and how to configure them.

It will also be able to create an effective program by understand the Status register.

Introduction to the Status Registers

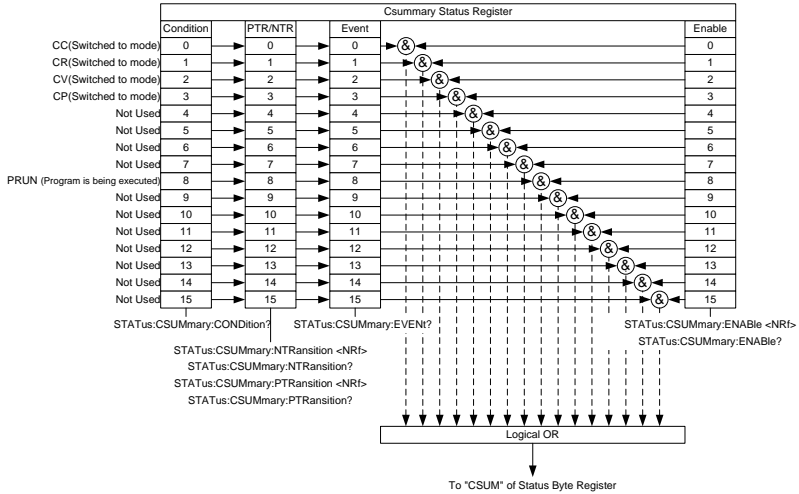
| | |
|----------|---|
| Overview | <p>The status registers are used to determine the status of the electronic load. The status registers maintain the status of the protection conditions, load conditions and channel conditions of the load modules.</p> <p>The PEL-3000H series have a number of register groups:</p> <ul style="list-style-type: none">● CSummary Registers● Operation Status Register Group● Questionable Status Register Group● Standard Event Status Register Group● Status Byte Register Group |
|----------|---|

Configuration in the Status register



Csummary Status Register Group

Overview Csummary Status register group, you can check the operating status of the load mode and sequence or program mode.

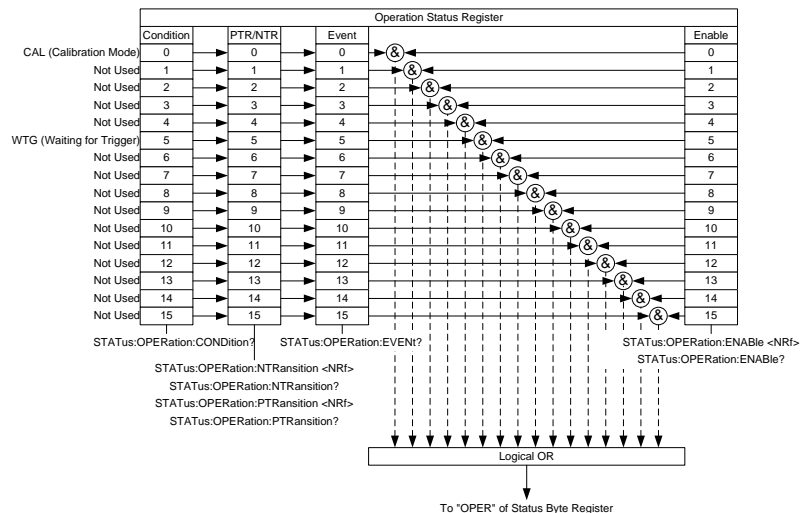


| Bit Summary | Bit name | Event | Bit # | Bit Weight |
|-------------|----------|---|-------|------------|
| | CC | Indicate the Constant Current mode setting. | 0 | 1 |
| | CR | Indicate the Constant Resistance mode setting. | 1 | 2 |
| | CV | Indicate the Constant Voltage mode setting. | 2 | 4 |
| | CP | Indicate the Constant Power mode setting. | 3 | 8 |
| | PRUN | Indicate the operation mode of Sequence or Program. | 8 | 256 |

| | | | | | |
|---------------------|---|---------------------|-----|---------------------|-----|
| Condition Register | Csummary Status Condition register can be read the current state of the load mode and program mode or sequence operation mode. | | | | |
| PTR/NTR Filter | The PTR/NTR (Positive/Negative transition) register determines the type of transition conditions that will set the corresponding bit in the Event Registers. Use the Positive transition filter to view events that change from false to positive, and use the negative transition filter to view events that change from positive to negative. | | | | |
| | <table> <tr> <td>Positive Transition</td> <td>0→1</td> </tr> <tr> <td>Negative Transition</td> <td>1→0</td> </tr> </table> | Positive Transition | 0→1 | Negative Transition | 1→0 |
| Positive Transition | 0→1 | | | | |
| Negative Transition | 1→0 | | | | |
| Event Register | The PTR/NTR Register will dictate the type of transition conditions will set the corresponding bits in the Event Register. If the Event Register is read, it will be cleared to 0. | | | | |
| Enable Register | The Enable register determines which Events in the Event Register will be used to set the CSUM bit in the Status Byte Register. | | | | |

Operation Byte Register Group

Overview Operation Status register group, you can check the operating status of the Trigger wait or the Calibration mode.



| Bit name | Event | Bit # | Bit Weight |
|----------|--------------------------------|-------|------------|
| CAL | Indicate the Calibration mode. | 0 | 1 |
| WTG | Indicate the Trigger wait. | 5 | 32 |

Condition Register Operation Status Condition register can be read by the current state of the Trigger waiting and Calibration mode.

PTR/NTR Filter The PTR/NTR (Positive/Negative transition) register determines the type of transition conditions that will set the corresponding bit in the Event Registers. Use the Positive transition filter to view events that change from false to positive, and use the negative transition filter to view events that change from positive to negative.

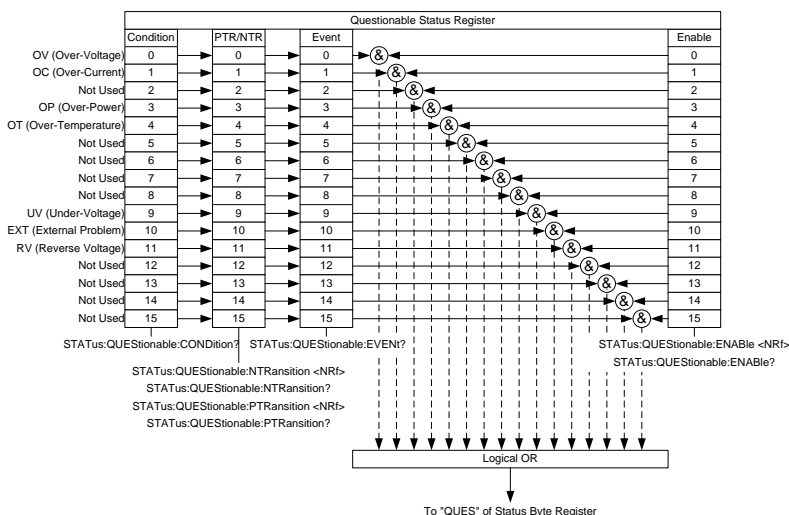
Positive Transition 0→1
 Negative Transition 1→0

Event Register The PTR/NTR Register will dictate the type of transition conditions will set the corresponding bits in the Event Register. If the Event Register is read, it will be cleared to 0.

Enable Register The Enable register determines which Events in the Event Register will be used to set the OPER bit in the Status Byte Register.

Questionable Status Register Group

Overview Questionable Status register group, you can check the operating status of the protection function.



Bit Summary

| Bit name | Event | Bit # | Bit Weight |
|----------|--------------------------------------|-------|------------|
| OV | Indicate the Over-Voltage condition. | 0 | 1 |
| OC | Indicate the Over-Current condition. | 1 | 2 |

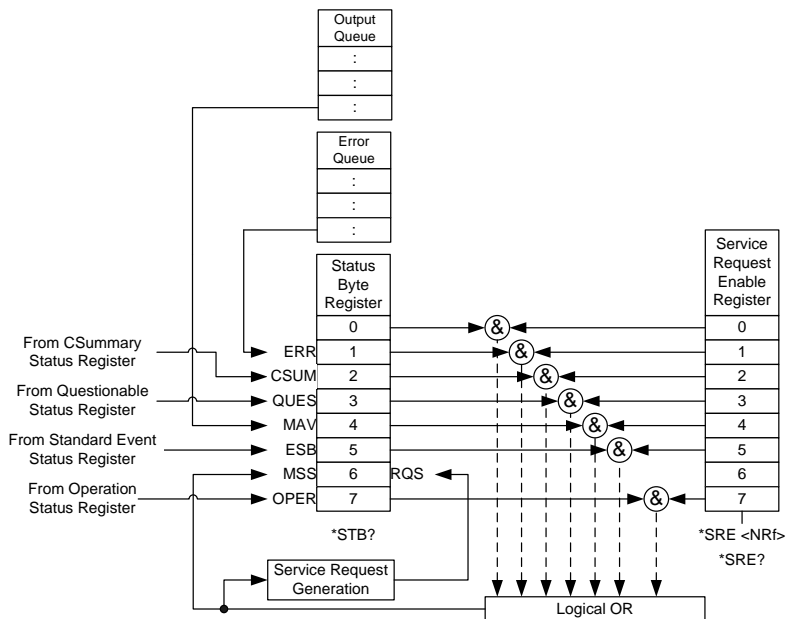
| | | | |
|--------------------|---|-----|------|
| OP | Indicate the Over-Power condition. | 3 | 8 |
| OT | Indicate the Over-Temperature condition. | 4 | 16 |
| PRUN | Indicate the operation mode of Sequence or Program. | 8 | 256 |
| UV | Indicate the Under-Voltage condition. | 9 | 512 |
| EXT | Indicate the abnormality state of the external control. | 10 | 1024 |
| RV | Indicate the reverse connection state. | 11 | 2048 |
| Condition Register | Questionable Status Condition register can be read the current state of the protection function. | | |
| PTR/NTR Filter | The PTR/NTR (Positive/Negative transition) register determines the type of transition conditions that will set the corresponding bit in the Event Registers. Use the Positive transition filter to view events that change from false to positive, and use the negative transition filter to view events that change from positive to negative. | | |
| | Positive Transition | 0→1 | |
| | Negative Transition | 1→0 | |
| Event Register | The PTR/NTR Register will dictate the type of transition conditions will set the corresponding bits in the Event Register. If the Event Register is read, it will be cleared to 0. | | |
| Enable Register | The Enable register determines which Events in the Event Register will be used to set the QUES bit in the Status Byte Register. | | |

| | | | | |
|-----------------|--|---|---|-----|
| | DDE | Device Dependent Error Device specific error. | 3 | 8 |
| | EXE | Execution Error The EXE bit indicates an execution error due to one of the following: illegal command parameter, parameter out of range, invalid parameter, the command didn't execute due to an overriding operation condition. | 4 | 16 |
| | CME | Command Error The CME bit is set when a syntax error has occurred. The CME bit can also be set when a <GET> command is received within a program message. | 5 | 32 |
| | URQ | User Request | 6 | 64 |
| | PON | Power On Indicates the power is turned on. | 7 | 128 |
| Event Register | Any bits set in the event register indicate that an error has occurred. Reading the Event register will reset the register to 0. | | | |
| Enable Register | The Enable register determines which Events in the Event Register will be used to set the ESB bit in the Status Byte Register. | | | |

Status Byte Register Group

Overview

The Status Byte register groups, you can check the status of the event in the status register of all. You can read the status byte register in "`*STB?`" Query command.



| Bit name | Event | Bit # | Bit Weight |
|----------|--|-------|------------|
| ERR | Error Event/Queue If data is present in the Error queue, the ERR bit will be set. | 1 | 2 |
| CSUM | Csummary Status Register The summary bit for the Csummary Status Register group. | 2 | 4 |

| | | | |
|----------------------|--|---|-----|
| QUES | Questionable Status Register The summary bit for the Questionable Status Register group. | 3 | 8 |
| MAV | Message Available This is set when there is data in the Output Queue waiting to be read. | 4 | 16 |
| ESB | Event Summary Bit. The ESB is the summary bit for the Standard Event Status Register group. | 5 | 32 |
| MSS /RQS | The MSS Bit is the summary of the Status Byte Register and Service Request register (bits 1-5, 7). This will be set to 1. | 6 | 64 |
| OPER | Operation Status Register OPER bit is the summary bit for the Operation Status Register Group. | 7 | 128 |
| Status Byte Register | Any bits set in the Status byte register acts as a summary register for all the four other status registers and indicates if there is a service request, an error in the Error Queue or data in the Output Queue. Reading the Status Byte register will reset the register to 0. | | |
| Status Byte Register | Any bits set in the Status byte register acts as a summary register for all the four other status registers and indicates if there is a service request, an error in the Error Queue or data in the Output Queue. Reading the Status Byte register will reset the register to 0. | | |

Service Request Enable Register Service Request Enable register specifies the bits in the Status Byte register for setting the MSS / RQS bit in the Status Byte register.

Also, MSS bit, can see in the "*STB?" Query command.

RQS bit is used to RQS bit of the GP-IB interface managed by the service request generator a bit of MSS. RQS bit is initialized after reading.

Error Messages

The following error messages may be encountered when reading the error queue.

| Error Code and string | Description |
|--------------------------------|---|
| Command Errors | |
| 0 NoError | No error |
| -100 Command Error | This is the generic syntax error for devices that cannot detect more specific errors. |
| -101 Invalid character | A syntactic element contains a character which 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 the 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 received than required for the header. |
| -110 Command header error | An error was detected in the header. |
| -111 Header separator error | A character which is not a legal header separator was encountered while parsing the header. |
| -112 Program mnemonic too long | The header contains more than twelve characters. |
| -113 Undefined header | The header is syntactically correct, but it is undefined for this specific 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 which appears to be numeric, including the nondecimal numeric types. |
| -121 Invalid character in number | An invalid character for the data type being parsed was encountered. |
| -123 Exponent too large | The magnitude of the exponent was larger than 32000. |
| -124 Too many digits | The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros. |
| -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, as well as errors -131 through -139, are generated when parsing a suffix. |
| -131 Invalid suffix | The suffix does not follow the syntax described in IEEE 488.2 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. |
| -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. |
| -161 Invalid block data | A block data element was expected, but was invalid for some reason. |
| -168 Block data not allowed | A legal block data element was encountered but was not allowed by the device at this point in parsing. |
| -170 Expression error | This error is generated when parsing an expression data element. |
| -171 Invalid expression | The expression data element was invalid. |
| -178 Expression data not allowed | A legal expression data was encountered but was not allowed by the device at this point in parsing. |
| -180 Macro error | This error is generated when defining a macro or executing a macro. |
| -181 Invalid outside macro definition | Indicates that a macro parameter placeholder (\$<number>) was encountered outside of a macro definition. |
| -183 Invalid inside macro definition | Indicates that the program message unit sequence, sent with a *DDT or *DMC command, is syntactically invalid. |
| -184 Macro parameter error | Indicates that a command inside the macro definition had the wrong number or type of parameters. |

Execution Errors

| | |
|-------------------------------|--|
| -200 Execution error | This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that an Execution Error as defined in IEEE 488.2 has occurred. |
| -201 Invalid while in local | Indicates that a command is not executable while the device is in local due to a hard local control |
| -202 Settings lost due to rtl | Indicates that a setting associated with a hard local control was lost when the device changed to LOCS from REMS or to LWLS from RWLS. |
| -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 | |
| -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. |
| -212 Arm ignored | Indicates that an arming signal was received and recognized by the device but was ignored. |
| -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 subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error. |

| | |
|------------------------------|--|
| -215 Arm deadlock | Indicates that the arm source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error. |
| -220 Parameter error | Indicates that a program data element related error occurred. |
| -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 possibilities, was expected. |
| -225 Out of memory. | The device has insufficient memory to perform the requested operation. |
| -226 Lists not same length. | Attempted to use LIST structure having individual LIST's of unequal lengths. |
| -230 Data corrupt or stale | Possibly invalid data; new reading started but not completed since last access. |
| -231 Data questionable | Indicates that measurement accuracy is suspect. |
| -232 Invalid format | Indicates that a legal program data element was parsed but could not be executed because the data format or structure is inappropriate. |

| | |
|---------------------------|--|
| -233 Invalid version | Indicates that a legal program data element was parsed but could not be executed because the version of the data is incorrect to the device. |
| -240 Hardware error | Indicates that a legal program command or query could not be executed because of a hardware problem in the device. |
| -241 Hardware missing | Indicates that a legal program command or query could not be executed because of missing device hardware. |
| -250 Mass storage error | Indicates that a mass storage error occurred. |
| -251 Missing mass storage | Indicates that a legal program command or query could not be executed because of missing mass storage. |
| -252 Missing media | Indicates that a legal program command or query could not be executed because of a missing media. |
| -253 Corrupt media | Indicates that a legal program command or query could not be executed because of corrupt media. |
| -254 Media full | Indicates that a legal program command or query could not be executed because the media was full. |
| -255 Directory full | Indicates that a legal program command or query could not be executed because the media directory was full. |
| -256 File name not found | Indicates that a legal program command or query could not be executed because the file name on the device media was not found. |
| -257 File name error | Indicates that a legal program command or query could not be executed because the file name on the device media was in error. |

| | |
|--------------------------------|---|
| -258 Media protected | Indicates that a legal program command or query could not be executed because the media was protected. |
| -260 Expression error | Indicates that a expression program data element related error occurred. |
| -261 Math error in expression | Indicates that a syntactically legal expression program data element could not be executed due to a math error. |
| -270 Macro error | Indicates that a macro-related execution error occurred. |
| -271 Macro syntax error | Indicates that that a syntactically legal macro program data sequence, according to IEEE 488.2 could not be executed due to a syntax error within the macro definition. |
| -272 Macro execution error | Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition. |
| -273 Illegal macro label | Indicates that the macro label defined in the *DMC command was a legal string syntax, but could not be accepted by the device. |
| -274 Macro parameter error | Indicates that the macro definition improperly used a macro parameter placeholder. |
| -275 Macro definition too long | Indicates that a syntactically legal macro program data sequence could not be executed because the string or block contents were too long for the device to handle. |
| -276 Macro recursion error | Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive. |

| | |
|-------------------------------------|---|
| -277 Macro redefinition not allowed | Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined. |
| -278 Macro header not found | Indicates that a syntactically legal macro label in the *GMC? query could not be executed because the header was not previously defined. |
| -280 Program error | Indicates that a downloaded program-related execution error occurred. |
| -281 Cannot create program | Indicates that an attempt to create a program was unsuccessful. A reason for the failure might include not enough memory. |
| -282 Illegal program name | The name used to reference a program was invalid. |
| -283 Illegal variable name | An attempt was made to reference a nonexistent variable in a program. |
| -284 Program currently running | Certain operations dealing with programs may be illegal while the program is running. |
| -285 Program syntax error | Indicates that a syntax error appears in a downloaded program. |
| -286 Program runtime error | |
| -290 Memory use error | Indicates that a user request has directly or indirectly caused an error related to memory or <data_handle>s, this is not the same as "bad" memory. |
| -291 Out of memory | |
| -292 Referenced name does not exist | |
| -293 Referenced name already exists | |

-294 Incompatible type Indicates that the type or structure of a memory item is inadequate.

Device Specific Errors

-300 Device-specific error This is the generic device-dependent error for devices that cannot detect more specific errors.

-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.

-312 PUD memory lost Indicates that the protected user data saved by the *PUD command has been lost.

-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.

-320 Storage fault [Indicates that the firmware detected a fault when using data storage. This error is not an indication of physical damage or failure of any mass storage element.

-321 Out of memory An internal operation needed more memory than was available.

-330 Self-test failed Fail of Self Test.

-340 Calibration failed Fail of Calibration.

-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 | This is the generic communication error. |
| -361 Parity error in program message | Parity bit not correct when data received. |
| -362 Framing error in program message | A stop bit was not detected when data was received. |
| -363 Input buffer overrun | Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing. |
| -365 Time out error | This is a generic device-dependent error. |

Query Errors

| | |
|---|--|
| -400 Query error | This is the generic query error. |
| -410 Query INTERRUPTED | Indicates that a condition causing an INTERRUPTED Query error occurred. |
| -420 Query UNTERMINATED | Indicates that a condition causing an UNTERMINATED Query error occurred. |
| -430 Query DEADLOCKED | Indicates that a condition causing an DEADLOCKED Query error occurred. |
| -440 Query UNTERMINATED after indefinite response | Indicates that a query was received in the same program message after a query requesting an indefinite response was executed |

Power On Event Commands

| | |
|---------------|--|
| -500 Power on | The instrument has detected an off to on transition in its power supply. |
|---------------|--|

User Request Event

| | |
|-------------------|---|
| -600 User request | The instrument has detected the activation of a user request local control. |
|-------------------|---|

Request Control Event

-700 Request control The instrument requested to become the active IEEE 488.1 controller-in-charge.

Operation Complete Event

-800 Operation complete The instrument has completed all selected pending operations in accordance with the IEEE 488.2 synchronization protocol.