

BK PRECISION®

Model 9129B

Triple Output Programmable DC Power Supply

Programming Manual




TABLE OF CONTENTS

1 Remote Operation	4
1.1 Interface Connection	4
Communication cables	4
<i>USB to TTL Adapter IT-E132B</i>	4
RS-232 Settings	5
2 Remote Commands	6
2.1 Parameter Definition	6
2.2 IEEE488.2 Common Commands	6
2.3 SCPI STATUS REGISTER	9
2.4 SYSTem Subsystem	14
2.5 OUTPut Subsystem	16
2.6 SOURce Subsystem	17
2.7 MEASure Subsystem	20
2.8 INSTRument Subsystem	21

1 Remote Operation

1.1 Interface Connection

Users can program the power supply by using the SCPI (Standard Commands for Programmable Instruments) commands over the TTL remote interface (via USB virtual com with IT-E132B).

Note: The RMT indicator will appear on display when the power supply is successfully connected to a PC remotely through any remote interface. Keys on the front panel will be locked until the instrument is in LOCAL mode. To return to LOCAL mode from the front panel, press the  key. The RMT indicator will disappear when the instrument is in LOCAL mode.

Communication cables

The power supply has a DB9 connector on the rear panel that allows remote communication via the IT-E132B adapter.



Do not connect the DB9 connector to a standard RS-232 serial port. Doing so may damage the instrument. The instrument outputs TTL signal levels only, and must be used with IT-E132B adapter to connect to a USB port (virtual com) on a PC.

USB to TTL Adapter IT-E132B

Use the IT-E132B USB to TTL adapter to connect a PC to the instrument for remote communication.

Connect the INSTRUMENT side of the adapter to the DC load DB9 connector. Connect the COMPUTER end to a USB port on your computer.

When connected, the computer will automatically install the appropriate drivers (For Windows® 7 or above). The driver is also available for download at www.bkprecision.com.

Upon successful installation of the driver, the USB connection will be registered as a virtual COM device. It can be accessed and control via the assigned COM port (This can be verified in “Device Manager”) the same way as you would with RS-232 serial interfaces.



RS-232 Settings

In order for the computer to communicate with the power supply, both must be set to the same RS-232 settings. These communication settings are:

- Baud rate must be one of 4800, 9600, or 38400
- 8 data bits
- One stop bit
- No parity

Refer to the Error! Reference source not found. section in the Menu in the USER MANUAL for instructions on how to select a different baud rate.

Note: The remote interface does not have a flow control mechanism. The programmer should be aware of this limitation and allow sufficient command process time. If the remote commands are sent too fast the power supply's internal buffer may overrun and cause a communication error. Therefore, adding a delay between commands to allow for sufficient time to process commands is required.

2 Remote Commands

2.1 Parameter Definition

The 9129B Power Supply supports the standard SCPI commands and a few proprietary commands that follow the SCPI convention. The SCPI interface of the 9129B Power Supply enables users to operate using a computer or a terminal equipped with RS-232. SCPI IEEE-488.2 also supports multi-unit control allowing a user to control up to 32 power supplies.

The following table lists all of the numerical parameters.

Symbol	Response Formats
<bool>	Boolean value, can be 1 or "ON", 0 or "OFF"
<NR1>	Integer value, can be zero, positive or negative integer number
<NRf>	Flexible numerical value, can be zero, positive or negative float point numeric value
<string>	String value, characters enclosed in single or double
<LF>	New line, hex code 0x0Ah
<CR>	Carriage return, hex code 0x0Dh

All commands should be terminated with the <LF> or <CR> and there should be a space between command and numerical parameter <s>.

2.2 IEEE488.2 Common Commands

Below is a list and description of all common SCPI commands supported by the instrument.

***CLS**

This command clears the following registers:

Standard event register

Query event register

Operation event register

Status byte register

Error code

Command syntax: *CLS

Parameter: None

***ESE**

This command can set the parameter of the standard event enable register. Setting parameter can determine which bit of standard event register is 1, and the byte will enable ESB of status

byte register as 1.

Command syntax: *ESE <NR1>

Parameter: 0~255

The value when power on: Refer to *PSC command

Example: *ESE 128

Query syntax: *ESE?

Returned parameter: <NR1>

The bit definition of the standard event enabled register:

Bit Position	7	6	5	4	3	2	1	0
Bit Name	PON	Not used	CME	EXE	DDE	QYE	Not used	OPC
Bit Weight	128		32	16	8	4		1

PON Power-on

CME Command error

EXE Execution error

DDE Device-dependent error

QYE Query error

OPC Operation complete

*ESR?

This command can read the value of standard event status register. After executing this command, standard event status register is reset. Bit definition of standard event status register is the same as the standard event status enable register.

Query syntax: *ESR?

Parameter: None

Returned parameter: <NR1>

*IDN?

This command can read the power supply information. The returned string contains 4 segments divided by comma.

Query syntax: *IDN?

Parameter: None

Returned parameter: <AARD> segment description

Manufacturer, Model, Serial number, Software version

Example: B&K Precision, 9205, 602203010697410001, V1.09-V1.04

*OPC

When all commands before this command are executed, bit OPC in standard event register will be set to 1.

Command syntax: *OPC

Parameter: None

Query syntax: *OPC?

Returned parameter: <NR1>

***PSC**

This command controls whether the power supply sends a query or not when it is reset. If set to 1 or ON, the operation event enable register, quest event enable register, and standard event status register will be reset when power supply is reset. When set to 0 or OFF, the data of the status byte register, operation event enable register, quest event enable register and standard event status enable register will be stored in non-volatile memory and recalled when power supply is reset.

Command syntax: *PSC <bool>

Parameter: 0|1|ON|OFF

Query syntax: *PSC?

Returned parameter: <NR1>

***RST**

This command resets the power supply to default settings.

Command syntax: *RST

Parameter: None

***SRE**

This command can set the parameter of state byte enable register. Setting parameter can determine which byte value of state byte register is 1, and the byte will set RQS of state byte register to 1. Bit definition of state byte enable register is the same as the state byte register.

Command syntax: *SRE <NR1>

Parameter: 0~255

Query syntax: *SRE?

Returned parameter: <NR1>

***STB?**

This command can read the data from the status byte register.

Query syntax: *STB?

Parameter: None

Returned parameter: <NR1>

***SAV**

This command saves the current setups of the power supply to a specified memory location . These setups contain set current, set voltage, and set voltage limits.

The memory is divided into 8 groups, and each group can store 1~9 (9 total) setups. Up to 72

setups can be saved in total.

Command syntax: *SAV<NRf>

Parameter: 1~9

Returned parameter: None

***RCL**

This command recalls the setups you saved previously from the specified memory location.

Command syntax: *RCL<NRf>

Parameter: 1~9

Returned parameter: None

***TST?**

This command will query the self-test results during power up.

Query Syntax: * TST?

Parameters: None

Return parameter: <NR1>

0 indicates that the unit passed self-test. A non-zero value indicates an error.

***WAI**

The Wait-to-Continue (WAI) command causes the instrument to wait until all pending commands are completed, before executing any other commands.

Parameters: None

Related command: * OPC

***PSC**

The Power-On Status Clear (PSC) command controls the automatic power-on clearing of the Service Request Enable Register, the Standard Event Status Enable Register, and device-specific event enable registers

1 / ON: When the power is turned on, all enable register will be cleared.

0 / OFF: When the power is turned on, the values of all the registers remain consistent with the last saved value.

Command syntax: * PSC <NR1>

Parameters: 0/1 / ON / OFF

Query syntax: * PSC?

2.3 SCPI STATUS REGISTER

The 9129B power supply stores the different states of the instrument through four status registers. The four status register groups are status byte register, standard event register, query status register, and operation status register. The status byte register stores the information of the other registers. The following tables provide the details on each of the registers.

STATus:QUESTionable[:EVENT]?

This command can be used to read the value in query event register. After executing this command, the query event register will be cleared.

Query syntax: STATus:QUESTionable[:EVENT]?

Parameter: None

Returned parameter: <NRf>

Relative command: STATus:QUESTionable:ENABle

STATus:QUESTionable:ENABle

This command can set the parameter of quest event enable register. Setting parameter can determine which bit value of questionable event register is 1 and the bit will enable QUES.

Command syntax: STATus:QUESTionable:ENABle <NRf>

Parameter: 0~65535

Default set: Refer to *PSC command

Example: STATus:QUESTionable:ENABle 128

Query syntax: STATus:QUESTionable:ENABle?

Returned parameter: <NRf>

Related command: *PSC

STATus: PRESet

This command clears the status event

Command syntax: STATus: PRESet

Parameters: None

Example: **STAT:PRESet**

STATus: QUESTionable: INSTRument [: EVENT]?

This command is to query the status event register of the instrument. The register is cleared after the query returns a value.

Query syntax: STATus: QUESTionable: INSTRument [: EVENT]?

Parameters: None

Return parameter: <NRf>

STATus: QUESTionable: INSTRument: ENABle

This command sets the parameter of questionable event enable register. Setting parameter can determine which bit value of quest event register is 1 and the bit will enable QUES.

Command syntax: STATus:QUESTionable:ENABle <NRf>

Parameter: 0~255

Example: STATus:QUESTionable:ENABle 128

Query syntax: STATus:QUEStionable:ENABle?

Returned parameter: <NRf>

STATus: QUEStionable: INSTrument: ISUMmary1 [: EVENT]?

This command queries the operation event register of Channel 1, and after the query returns a value, the value of this register is cleared.

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary1 [: EVENT]?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM1:EVEN?

STATus: QUEStionable: INSTrument: ISUMmary1: ENABle

This command modifies the value of the operation enable register summary of Channel 1.
rewrite value channel enable register 1

Command syntax: STATus: QUEStionable: INSTrument: ISUMmary1: ENABle

Parameters: register values

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary1: ENABle?

Example: STAT:QUES:INST:ISUM1:ENAB 12

STATus: QUEStionable: INSTrument: ISUMmary1: CONDition?

This command queries the questionable condition register summary of channel 1

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary1: CONDition?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM2:COND?

STATus: QUEStionable: INSTrument: ISUMmary2 [: EVENT]?

This command queries the operation event register of Channel 2, and after the query returns a value, the value of this register is cleared.

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary2 [: EVENT]?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM1:EVEN?

STATus: QUEStionable: INSTrument: ISUMmary2: ENABle

This command modifies the value of the operation enable register summary of Channel 2.
rewrite value channel enable register 2

Command syntax: STATus: QUEStionable: INSTrument: ISUMmary2: ENABle

Parameters: register values

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary2: ENABle?

Example: STAT:QUES:INST:ISUM2:ENAB 12

STATus: QUEStionable: INSTrument: ISUMmary2: CONDition?

This command queries the questionable condition register summary of channel 2

Query syntax: STATus: QUEStionable: INSTrument: ISUMmary2: CONDition?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM2:COND?

STATus: QUEStionable: INSTRument: ISUMmary3 [: EVENT]?

This command queries the operation event register of Channel 3. After the query returns a value, the value of this register is cleared.

Query syntax: STATus: QUEStionable: INSTRument: ISUMmary3 [: EVENT]?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM1:EVEN?

STATus: QUEStionable: INSTRument: ISUMmary3: ENABle

This command modifies the value of the operation enable register summary of Channel 3.

Command syntax: STATus: QUEStionable: INSTRument: ISUMmary3: ENABle

Parameters: register values

Query syntax: STATus: QUEStionable: INSTRument: ISUMmary1: ENABle?

Example: STAT:QUES:INST:ISUM3:ENAB 12

STATus: QUEStionable: INSTRument: ISUMmary3: CONDition?

This command queries the questionable condition register summary of channel 3

Query syntax: STATus: QUEStionable: INSTRument: ISUMmary3: CONDition?

Returned parameter: <NRf>

Example: STAT:QUES:INST:ISUM3:COND?

STATus: OPERation [: EVENT]?

This command reads the value of the operation event register. After this command is executed, the operation event register is cleared.

Query syntax: STATus: OPERation [: EVENT]?

Parameters: None

Return parameter: <NR1>

Related commands: STATus: OPERation: ENABle

STATus: OPERation: ENABle

This command is used to set the contents of the operation enable register.

Command syntax: STATus: OPERation: ENABle <NRf>

Parameters: 0 to 255

Reset value: Reference *PSC command

For example: STATus: OPERation: ENABle 128

Query syntax: STATus: OPERation: ENABle?

Return parameter: <NR1>

STATus: OPERation: INSTRument [: EVENT]?

This command queries the value of the channel operation event register. After the command is

executed, the value of this register is cleared.

Query syntax: STATus: OPERation: INSTRument [: EVENT]?

Parameters: None

STATus: OPERation: INSTRument: ENABLE

This command sets the value of the channel operation enable register

Command syntax: STATus: OPERation: INSTRument: ENABLE

Parameters: register values

Query syntax: STAT:OPERation:INSTRument: ENABLE?

STATus: OPERation: INSTRument: ISUMmary1 [: EVENT]?

This command registers a channel 1 query event, and after the query returns a value, the value of this register is cleared.

Query syntax: STATus: OPERation: INSTRument: ISUMmary1 [: EVENT]?

Parameters: None

Example: OPERation:INSTRument:ENABLE?

STATus: OPERation: INSTRument: ISUMmary1: ENABLE

This command re-writes the value of channel 1 operation enable register

Command syntax: STATus: OPERation: INSTRument: ISUMmary1: ENABLE

Parameters: register values

Query syntax: STATus: OPERation: INSTRument: ISUMmary1: ENABLE?

STATus: OPERation: INSTRument: ISUMmary1: CONDition?

This command queries the status of the channel value register 1.

Query command: STATus: OPERation: INSTRument: ISUMmary1: CONDition?

Parameters: None

Example: STAT:OPER:INST:ISUM1:COND?

STATus: OPERation: INSTRument: ISUMmary2 [: EVENT]?

This command registers a channel 2 query event, and after the query returns a value, the value of this register is cleared.

Query syntax: STATus: OPERation: INSTRument: ISUMmary2 [: EVENT]?

Parameters: None

Example: STAT:OPER:INST:ISUM2?

STATus: OPERation: INSTRument: ISUMmary2: ENABLE

This command overrides value of the channel 2 operation enable register

Command syntax: STATus: OPERation: INSTRument: ISUMmary2: ENABLE

Parameters: register values

Query syntax: STATus: OPERation: INSTRument: ISUMmary2: ENABLE?

STATus: OPERation: INSTRument: ISUMmary2: CONDition?

This command queries the status of the channel value register 2.

Query command: STATus: OPERation: INSTRument: ISUMmary2: CONDition?

Parameters: None

Example: STAT:OPER:INST:ISUM1:COND?

STATus: OPERation: INSTRument: ISUMmary3 [: EVENT]?

This command registers a channel 3 event query, and after the query returns a value, the value of this register is cleared.

Query syntax: STATus: OPERation: INSTRument: ISUMmary3 [: EVENT]?

Parameters: None

Example: STAT:OPER:INST:ISUM2?

STATus: OPERation: INSTRument: ISUMmary3: ENABLE

This command overrides value channel 3 operation enable register

Command syntax: STATus: OPERation: INSTRument: ISUMmary3: ENABLE

Parameters: register values

Query syntax: STATus: OPERation: INSTRument: ISUMmary3: ENABLE?

STATus: OPERation: INSTRument: ISUMmary3: CONDition?

This command queries the status the channel value of register 3.

Query command: STATus: OPERation: INSTRument: ISUMmary3: CONDition?

Parameters: None

Example: STAT:OPER:INST:ISUM3:ENAB?

2.4 SYSTEM Subsystem

SYSTEM:ERRor?

This command is used to query the error information of the power supply. When the error LED indicator appears on the front panel, one or more hardware or command syntax error has occurred. There are at most 20 groups of error message. Sending the command once will read one error message from the error queue.

1. Error messages follow the FIFO (first-in-first-out) principle. The first error will be returned first. When all the error information in the error queue is read, the ERROR LED will disappear. When an error occurs, the buzzer of the power supply will beep once.
2. If more than 20 errors occur, the last one stored in the queue will be replaced by "-350", meaning "too many errors". If the error messages in the queue are not read, the subsequent errors will not be saved in the queue. If there is no error message when reading error information, it will return "0", meaning "no error".

3. Turning off the power supply or sending the "CLS" (clear status) command will clear all errors in the queue. *RST command will not clear the error messages in the error queue.

Command syntax: SYST:ERR?

Parameter: None

Returned parameter: <NR1>,<SRD>

The following is a complete list of error codes and their descriptions:

- (0) No error
- (1) Module Initialization Lost
- (2) Mainframe Initialization Lost
- (3) Module Calibration Lost
- (4) EEPROM failure
- (101) Too many numeric suffices
- (110) No input command
- (114) Invalid Numeric suffix
- (116) Invalid value
- (117) Invalid dimensions
- (120) Parameter overflowed
- (140) Wrong type of parameter
- (150) Wrong number of parameter
- (160) Unmatched quotation mark
- (165) Unmatched bracket
- (170) Invalid command
- (180) No entry in list
- (190) Too many dimensions
- (191) Too many char
- (-200) Execution error
- (-221) Settings conflict
- (-222) Data out of range
- (-223) Too much data
- (-224) Illegal parameter value
- (-225) Out of memory
- (-230) Data Corrupt or Stale
- (-310) System error
- (-350) Too many errors
- (-400) Query error
- (-410) Query INTERRUPTED
- (-420) Query UNTERMINATED
- (-430) Query DEADLOCKED
- (223) Front panel buffer overrun
- (224) Front panel timeout
- (225) Front Crc Check error
- (401) CAL switch prevents
- (402) CAL password is incorrect

- (403) CAL not enabled
- (404) Readback cal are incorrect
- (405) Programming cal are incorrect

SYSTem:VERSion?

This command is used to query the current SCPI version. The returned parameter is a string in the format "YYYY.V", where YYYY is the year of the version, and V is the software version of that year.

Command syntax: SYST:VERS?

Parameter: None

Returned parameter: <NRf>

SYSTem:REMOte

This command is used to enable remote mode. Except for the Local key on the front panel, all keys are locked and cannot be used when in remote mode. Sending or receiving most commands without first sending this command to enable remote mode will cause errors.

Command syntax: SYST:REM

Parameter: None

SYSTem:LOCal

This command is used to switch the instrument to local control mode (front panel control).

Command syntax: SYST:LOCal

Parameter: None

SYSTem:BEEPer

This command allows the instrument to emit a beep sound.

Command Syntax: SYSTem:BEEPer

Parameters: None

2.5 OUTPUT Subsystem

OUTPut [:STATe][:ALL]

This command sets the state of the output, on or off, of all the channels.

Command Syntax: OUTPut[:STATe] <bool>

Parameters: 0 | 1 | ON | OFF

* RST value: OFF

Query syntax: OUTPut:STATe[:ALL]?

Return parameter: 0 | 1

Example: OUTP:STAT 0

OUTPut:SERies [:STATe]

This command sets the state of channels CH1 + CH2 combined in series

Command syntax: `OUTPut:SERies[:STATe] <bool>`

Parameters: 0 | 1 | ON | OFF

Query syntax: `OUTPut:SERies[:STATe]?`

Return parameter: 0 | 1

Example: `OUTp:SER 1`

OUTPut:PARAllel [:STATe]

This command sets the state of channels CH1 + CH2 combined in parallel

Command syntax: `OUTPut:PARAllel[:STATe] <bool>`

Parameters: 0 | 1 | ON | OFF

Query syntax: `OUTPut:PARAllel[:STATe]?`

Return parameter: 0 | 1

Example: `OUTP:PARA 1`

2.6 SOURce Subsystem

[SOURce:]CHANnel:OUTPut [:STATe]

This command is used to modify the output of the selected channel on or off.

Command syntax: `[SOURce:]CHANnel:OUTPut [:STATe] <bool>`

Parameters: 0 | 1 | ON | OFF

* RST value: OFF

Query syntax: `[SOURce:]CHANnel:OUTPut[:STATe]?`

Return parameter: 0 | 1

Example: `SOUR:CHAN:OUTP:STAT 0`

[SOURce:]OUTPut:PROTection:CLEar

This instruction is used to clear protected power supply, such as the protection OVP and OTP protection

Command syntax: `[SOURce:]OUTPut:PROTection:CLEar`

Parameters: None.

[SOURce:]CURRent[:LEVel][:IMMEDIATE][:AMPLitude]

This command is used to set the current value of the selected channel.

Command syntax: `[SOURce:]CURRent[:LEVel] <NRf>`

Parameters: num | MIN | MAX | UP | DOWN | DEF

Unit: A mA

* RST value: MIN

Examples: CURR 3A, CURR 30mA, CURR MAX, CURR MIN

Query syntax: [SOURce:] CURRent[:LEVel][:IMMEDIATE][:AMPLitude]?

Parameters: [MIN | MAX]

Examples: CURR? , CURR? MAX, CURR? MIN

Return parameter: <NR2>

[SOURce:]CURRent[:LEVel]:UP[:IMMEDIATE][:AMPLitude]

This command will increase the value of the current in steps.

Command syntax: [SOURce:]CURRent[:LEVel]:UP[:IMMEDIATE][:AMPLitude] <NRf>

Parameters: None

Related commands: [SOURce:]CURRent[:LEVel][:IMMEDIATE]: STEP[:INCRement]

Example: CURR: UP

SOURce:]CURRent[:LEVel]:DOWN[:IMMEDIATE][:AMPLitude]

This command will step will decrease the current value of the current channel.

Command syntax: [SOURce:]CURRent[:LEVel]:DOWN[:IMMEDIATE][:AMPLitude] <NRf>

Parameters: None

Related commands: [SOURce:]CURRent[:LEVel][:IMMEDIATE]:STEP[:INCRement]

Example: CURR:DOWN

[SOURce:]CURRent[:LEVel][:IMMEDIATE]:STEP[:INCRement]

This command sets the step current value.

Command syntax: [SOURce:]CURRent[:LEVel][:IMMEDIATE]:STEP[:INCRement] <NRf>

Parameters: current value

Unit: A | mA | uA

Query syntax: [SOURce:]CURRent[:LEVel][:IMMEDIATE]:STEP[:INCRement]?

Example: SOUR:CURR:LEV:IMM:STEP:INCR

[SOURce:]APPLY:CURRent[:LEVel][:IMMEDIATE][:AMPLitude]

This command allows setting the output current of all three channels simultaneously.

Command syntax: [SOURce:]APPLY:CURRent[:LEVel][:IMMEDIATE][:AMPLitude]

<Nr1>, <nr2>, <nr3>

Parameter 1: The current value of the first channel

Parameter 2: The current value of the second channel

Parameter 3: The current value of the third channel

Unit: A, mA, uA

Example: APP:CURR 1,0,.4

Query syntax: [SOURce:]APPLY:CURRent[:LEVel][:IMMEDIATE][:AMPLitude]?

Parameters: None

Return Value: <NRf>,<NRf>,<NRf>

[SOURce:]APPLY:OUTput[:STATE]

Controls the output state of all 3 channels simultaneously.

Command syntax: [SOURce:]APPLY:OUTput[:STATe] <b1>,<b2>,<b3>

Parameter 1: 1 | 0 | ON | OFF

Parameter 1: The output state of the first channel

Parameter 2: The output state of the second channel

Parameter 3: The output state of the third channel

Example: APP: OUT 0, 1, 0

Query syntax: [SOURce:]APPLY:OUTput[:STATe]?

Parameters: None

Returns: <bool>,<bool>,<bool>+

[SOURce:]APPLY:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]

Sets the voltage value of all the channels simultaneously.

Command syntax: [SOURce:]APPLY:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]
<Nr1>,<Nr2>,<Nr3>

Parameter 1: The voltage value of the first channel

Parameter 2: The voltage value of the second channel

Parameter 3: The voltage value of the third channel

Unit: V, kV, mV, uV

Example: APP:VOLT 3 modifying channel 1

APP:VOLT 2,1,2 modifying channels 1-3

Query syntax: [SOURce:]APPLY:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]?

Parameters: None

Return Value: <NRf>,<NRf>,<NRf>

[SOURce:]VOLTage[:LEVel]:UP[:IMMEDIATE][:AMPLitude]

This command will increase the voltage value by one step.

Command syntax: [SOURce:]VOLTage[:LEVel]:UP[:IMMEDIATE][:AMPLitude] <NRf>

Parameters: None

Related commands: [SOURce:]VOLTage[:LEVel][:IMMEDIATE]:STEP[:INCRement]

Example: VOLT:UP

[SOURce:] VOLTage[:LEVel]:DOWN[:IMMEDIATE][:AMPLitude]

This command will decrease the voltage value by one step.

Command syntax: [SOURce:]VOLTage[:LEVel]:DOWN[:IMMEDIATE][:AMPLitude] <NRf>

Parameters: None

Related commands: [SOURce:]VOLTage[:LEVel][:IMMEDIATE]:STEP[:INCRement]

VOLT:DOWN

[SOURce:] VOLTage[:LEVel][:IMMEDIATE]:STEP[:INCRement]

This command sets the value of the step voltage.

Command syntax: [SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement] <NRf>

Parameters: Voltage value

Unit: kV | V | mV | uV

Query syntax: [SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]?

[SOURce:]VOLTage:LIMIT[:LEVel]

This command sets the output voltage limit.

Command syntax: [SOURce:]VOLTage:LIMIT[:LEVel] <NRf>

Parameters: num | Min | Max | def

Unit: kV | V | mV | uV

Query: [SOURce:]VOLTage:LIMIT[:LEVel]?

Example: SOUR:VOLT:LIMIT 31

2.7 MEASure Subsystem

MEASure[:SCALar]:CURRent[:DC]?

FETCh:CURRent [: DC]?

This command is used to read the power supply output current.

Command Syntax: MEASure[:SCALar]:CURRent? [:DC]<nr>

Optional parameter CH1 | CH2 | CH3 | ALL

Return parameter:

If you use the optional parameter CH1 | CH2 | CH3 designated measurement channel, the actual value for the specified channel is returned. If you use the optional parameter ALL, the measured current of all channels are returned.

If you do not use the optional parameters, the selected channel's actual current is returned/

Return parameters Unit: A

Example: MEAS:CURR:ALL?

MEASure[:SCALar][:VOLTage][:DC]?

FETCh [: VOLTage] [: DC]?

This command is used to read the output voltage of the power supply.

Command Syntax: MEASure[:SCALar][:VOLTage]? [:DC]<nr>

Optional parameter CH1 | CH2 | CH3 | ALL

Return parameter:

If you use the optional parameter CH1 | CH2 | CH3 designated measurement channel, the actual value for the specified channel is returned. If you use the optional parameter ALL, the measured voltage of all channels are returned.

If you do not use the optional parameters, the selected channel's actual voltage is returned.

Return parameters Unit: V

Example: MEAS?

MEASure[:SCALar]:POWer[:DC]?

FETCh: POWer [: DC]?

This command is used to read the output power of the power supply.

Command Syntax: MEASure[:SCALar]:POWer? [:DC]<nr>

Optional parameter CH1 | CH2 | CH3 | ALL

Return parameter:

If you use the optional parameter CH1 | CH2 | CH3 designated measurement channel, the actual value for the specified channel is returned. If you use the optional parameter ALL, the measured power of all channels are returned.

If you do not use the optional parameters, the selected channel's actual power is returned.

Return parameters Unit: W

Example: MEAS:POW?

MEASure[:SCALar]:CURRent:ALL[:DC]?

Returns the actual current value of all 3 channels simultaneously.

Command Syntax: MEASure[:SCALar]:CURRent: ALL[:DC]?

Parameters: None

Return Value Format: X.XXX, Y.YYY, Z.ZZZ

X.XXX: CH1 actual current

Y.YYY: CH2 actual current

Z.ZZZ: CH3 actual current

The return value units, A

The same command: MEAS:CURR:ALL?

MEASure[:SCALar][:VOLTage]:ALL[:DC]?

Returns the actual voltage value of all 3 channels simultaneously.

Command Syntax: MEASure[:SCALar][:VOLTage]:ALL[:DC]?

Parameters: None

Return Value Format: X.XXX, Y.YYY, Z.ZZZ

X.XXX: CH1 actual voltage

Y.YYY: CH2 actual voltage

Z.ZZZ: CH3 actual voltage

The return value units, V

The same command: MEAS:ALL?

2.8 INSTRUMENT Subsystem

INSTRument[:SElect]

This instruction is used to select a channel.

Command Syntax: INSTRument[:SElect] <channel>
Parameters: CH1 | CH2 | CH3
Query syntax: INSTRument[:SElect]?
Example: INST:SEL Ch1

INSTRument:NSElect

This instruction is used to select a channel.
Command syntax: INSTRument:NSElect <NR1>
Parameters 1-3
Query syntax: INSTRument:NSElect?

INSTRument:COMbine:SERies

Sets the supply to series mode.
Command syntax: INSTRument:COMbine:SERies
Parameters: None
Query syntax: INSTRument:COMbine?

INSTRument:COMbine:PARAllel

Sets the supply to parallel mode.
Command syntax: INSTRument:COMbine:PARAllel
Parameters: None
Query syntax: INSTRument:COMbine?

INSTRument:COMbine:OFF

Disable series or parallel mode.
Command syntax: INSTRument:COMbine:OFF
Parameters: None
Query syntax: INSTRument:COMbine?