Faithtech

Wide range high power Bi-directional programmable DC power supply

SCPI Programming manual

(FTB9000 series)

VERSION V1.00

2022-01

Introduction

This manual is applicable to FTB9000 series bidirectional high power programmable source load system. Unless otherwise stated, references to "switching power supply" or "power supply" in the manual refer to the product series.

Related Information

This manual assumes that you know how to connect a power supply to a computer. Please refer to the user manual for detailed on-line method.

Some of the content in the manual is related to specific attachments. If you need some special accessories or random accessories are not enough to meet your needs, please contact Faith Technology sales or after-sales service department

Notice

For more information about our products, please visit our website http://en.faithtech.cn.

Version revision record

Date	version	Revised Chapters
Jan,2022	1.00	Complete this manual

Directory

1. Overview

1.1. Introduction

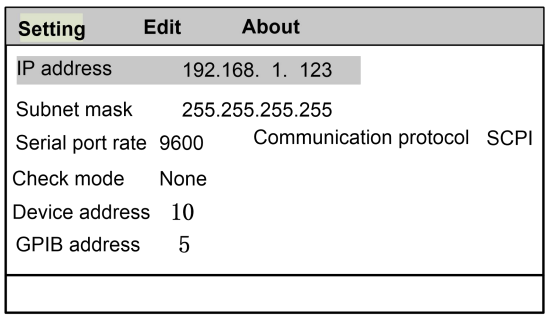
FTB9000 Bi-directional high power programmable DC switching power supply provides USB serial port (standard), LAN (standard), RS485 (optional),CAN (optional) and other remote communication interface, you can connect the power supply and the computer through a special cable, to achieve computer control of the source.

表格 1-1Power communication port

Remote control equipment	Communication interface	overview
computer	USB port	USB port Analog serial port
	LAN	Standard Ethernet communication cable

1.2. Remote interface configuration

This section describes the configuration method of each communication interface in detail. These configurations can only be set through the power front panel keyboard. For more details about the configuration, see the user manual. Press the “Menu” key to Enter the menu. Under the "Settings" column, select "System" and press "Enter" to enter the system parameter setting interface.



Photot 1-1System parameter setting screen

Use the knob or arrow keys to move the cursor to the setting item and press “Enter “to enter the parameter editing mode. Enter the number key and decimal key to edit the IP address, rotate the knob to select the baud rate and verification mode. You can press “Enter”

to confirm the editing parameters, or press “Esc” to exit the editing mode.

The communication configuration information is stored in the host's non-volatile memory and is not affected by shutting down or calling preset Settings. After modifying the communication parameters, restart the device. The power supply will configure the communication interface according to the preset parameters

Connecting a USB serial Port

USB serial ports are universal USB communication ports that comply with USB flat specifications. The USB serial port uses the 9600 baud rate by default. The value can be 4800, 9600, 19200, 38400, or 115200bps. The baud rate of the power supply and the computer must be consistent. The factory default check bit function is off (no check). If the parity bit function is enabled, the RS232 interface uses odd or even check to verify data.

Connect to LAN

FTB9000 series power supply has Ethernet communication interface, using UDP communication mode, the default port number is 7000.

Before the communication starts, set the IP address and subnet mask, and ensure that the IP addresses of the PC and the power supply are on the same network segment.

1.3. Enter the remote control mode

After receiving any correct SCPI instruction, the power supply enters the remote control mode.

In remote control mode: the local keyboard is locked, the key operation is invalid, and the power supply can only be controlled through programming commands; The screen on the front panel displays real-time status information such as voltage, current, and power. There are two ways to exit remote control mode:

press "Enter" button, the system back to the local operating mode.

send programming commands "SYSTem: the LOCal" can make the power back to LOCal mode.

1.4. Additional information

For details about software operations, driver installation, and communication operations, see the user manual and delivered software manuals. For the latest information on the software and drivers, please visit the Faith Technology website at http://en.faithtech.cn.

2. Overview of programming commands

2.1. Introduction to programming commands

There are two types of power commands: IEEE488.2 public commands and device-specific SCPI commands.

IEEE488.2 Common commands define a number of control and query commands common to instrumentation. You can use public commands to perform basic operations on the power supply, such as reset and status query. All IEEE 488.2 public commands consist of "*" and three-letter mnemonics, such as: *RST, *IDN? , * ESE? And so on.

The SCPI command implements most of the testing, setting, calibration and measurement functions of the power supply. Such commands are organized as command trees. Each command can contain multiple mnemonics, command each node of the tree in character between ":" space, as shown in figure 2 - 1. The top ROOT of the command tree is called the root. The full path from the root to the leaf is a complete programming command.

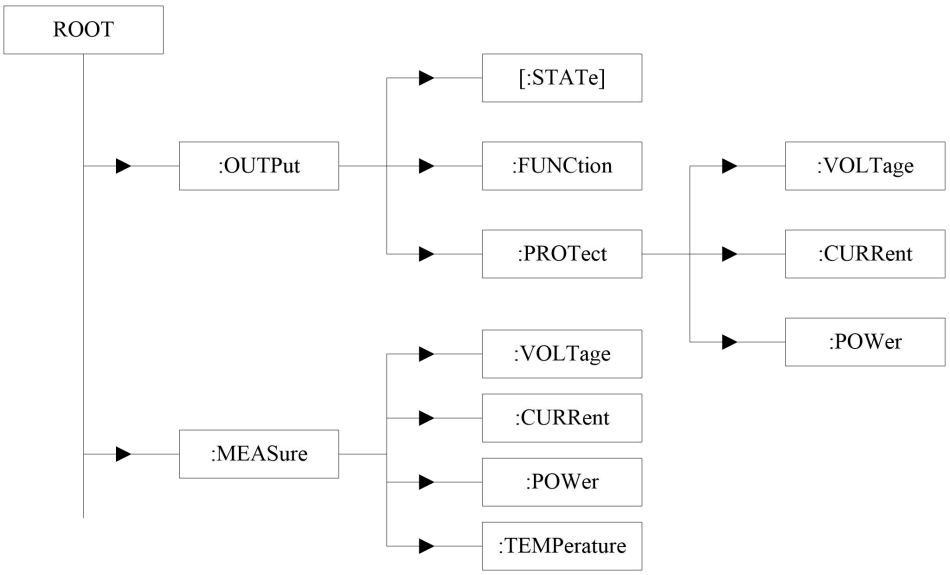


图 2- 1 Example command tree

2.2. Programming command syntax

The SCPI command of FTB9000 series Bi-directional high power programmable DC

switching power supply is the inheritance and extension of IEEE488.2 command specification. The SCPI command consists of command keywords, delimiters, parameter fields, and terminators. The following command is used as an example:

CURRent:STATic:L110.0A

In this command, CURRent, STATic, and L1 are command keywords, the characters: and Spaces are delimiters, 10.0 is parameter (some commands have multiple parameters separated by commas), and the carriage return character after the command is the end character of the command.

For the convenience of description, the following conventions are adopted for each symbol in the following sections:

square brackets ([]) optional keyword or parameters, can be omitted.

curly braces ({}) indicates that the command string of the parameters in the options.

Angle brackets (< >) must provide a numerical parameters.

vertical line (|) used to separate multiple options of optional parameters

Command keywords

Each command keyword has two formats: long mnemonic and short mnemonic. Short mnemonic is short for long mnemonic. Each mnemonic is no more than 12 characters long (including any numeric suffixes that may occur). The power supply accepts only precise long or short mnemonics. The rules for generating mnemonics are as follows:

A long mnemonic consists of a word or phrase. If it is a word, the whole word forms a mnemonic; In the case of a phrase, the first character of each word and the entire last word form a mnemonic.

CONFIGURE —— CONFigure

Main Value —— MVALue

The short mnemonics generally consist of the first four characters of the long mnemonics.

CONFigure —— CONF

If the length of the long mnemonics is less than or equal to 4, the long and short mnemonics are the same; If the length of the long mnemonic is greater than 4 and the fourth character is a vowel, the short mnemonic will drop the vowel and become 3 characters.

SAVE —— SAVE

TIMer —— TIM

Mnemonic is not case sensitive.

Delimiter character

Colon ":"

Separate two adjacent keywords in the command, for example, CURR and MVAL in the CURR:MVAL 10 command. It can also be the first character of a command, indicating that the path is to be found from the top node of the command tree.

Spaces

Separate command fields from parameter fields.

semicolon ";"

If a command contains multiple command units, separate multiple command units. Using a semicolon does not change the hierarchy of the current path. For example, the following command:

"CURR: reaching 0; MVAL 10"

Set the range of the constant current function to the maximum range and the main current value to 10A. It is equivalent to the following two commands:

"CURR: reaching 0"

"CURR: MVAL 10"

comma ","

Separate multiple parameters in the parameter field. For example, in the CAL:STAT ON,6900 command, use commas (,) to separate parameters ON and 6900.

Query indicator

The question mark "? Used to mark the query function of a command. The tag position is the last keyword immediately following the command domain. For example, to query the primary value of the constant current function, run the CURR? Command. . Assuming a primary value of 10A, the power supply returns the string "10.000".

After receiving a query command and parsing it, the PSU executes the command and generates a response string. The response string is first written to the output buffer. If the current remote interface is a GPIB interface, it waits for the controller to read the response. Otherwise, the response string is immediately sent to the interface.

Most setup commands have corresponding query syntax. If a command that cannot be queried is received, the PSU reports an error message "-115Command can not query", and nothing is returned.

The command terminator

Command terminators are available in two formats: newline (ASCII character LF with a value of 10) and EOI (only available for the GPIB interface). The terminator ends the current command string and resets the command path to the root path.

2.3. Parameter Format

Programming parameters are numeric, character, a variety of types such as Boolean, see table 2 - 1 parameter format, but no matter what type, are expressed in ASCII string.

表格 2-1Parameter Format

symbol	Make it clear	example
<NR1>	Integer number value。	123
<NR2>	Floating point value。	123.，12.3，0.12，1.23E4
<NRf>	The number may be NR1 or NR2.	
<NRf+>	Extended numerical formats, including < the NRf >, MIN and MAX.	
<Bool>	Boolean data	1|0|ON|OFF
<CRD>	Character data such as CURR.	
<AARD>	Returns ASCII data. Allows the undefined 7-bit ASCII code to be returned. This data format implies a command terminator.	

2.4. Status system

The state system records various conditions and states of the power supply in each state register group. The structure of the state system is shown in the figure below. The state system includes standard event register group, channel state register group and state byte register.

Each register group consists of several registers, including the status register, the event register, and the enable register.

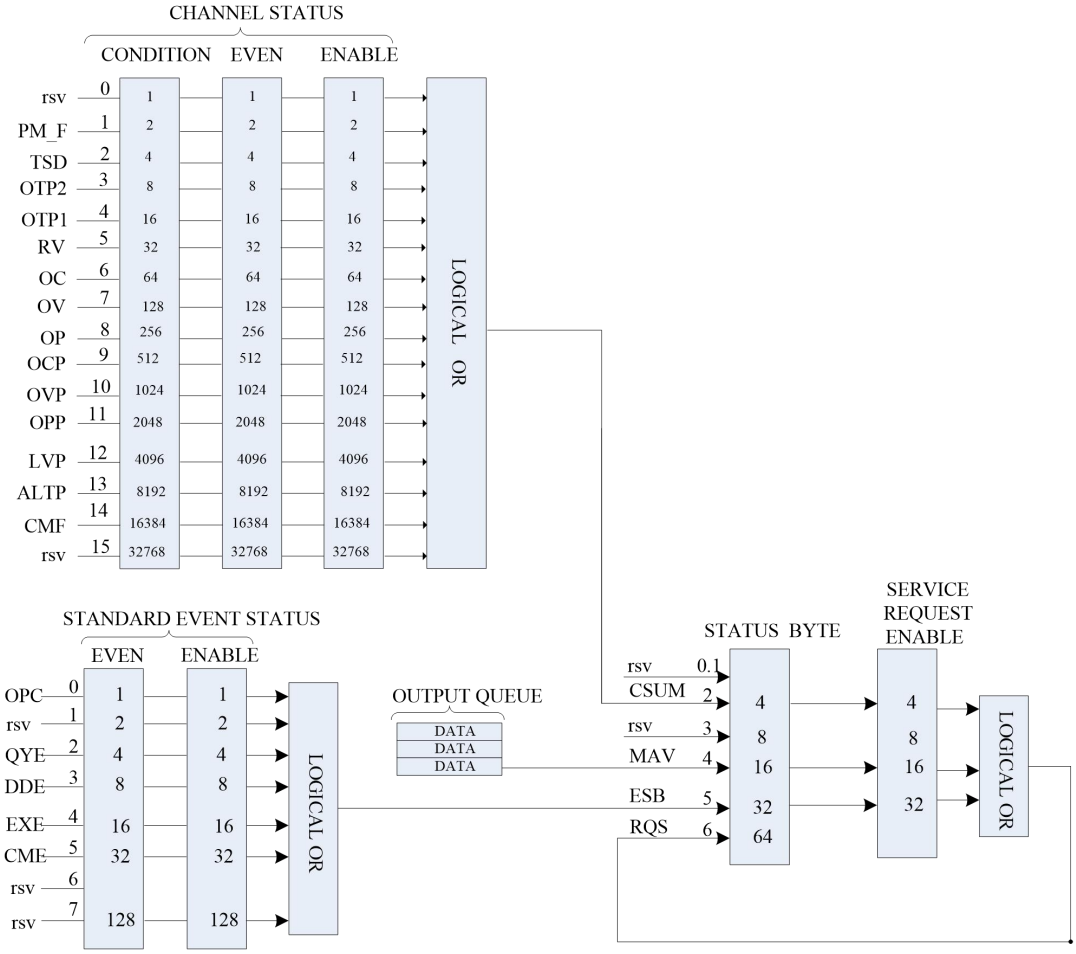


Photo 2-2State system of power supply

Channel status register group

Channel Status register group (Channel Status Registers) is reflected power real-time Status and register set of events, The system includes Channel Condition register, PTR filter register, NTR filter register, Channel Event register and Channel Event Enable register.

The channel status register records the real-time state of the power supply. The main content is the alarm information of the power supply, including overcurrent state, overvoltage state, etc. The detailed definition is shown in the following table.

表格 2-2Channel status bit definition

Bit	0	1	2	3	4	5	6	7

Instruction	rsv	PM_F	TSD	OTP2	OTP1	RV	OC	OV
rsv：reservation PM_F：PFC Module fault TSD：Temperature sensor is damaged OTP2：Transformer overtemperature protection OTP1：Module overtemperature protection RV：Output reverse connection protection OC：The output current exceeds the rated value OV：The output voltage exceeds the rated value								
Bit	8	9	10	11	12	13	14	15
Instruction	OP	OCP	OVP	OPP	LVP	ALTP	CMF	rsv
OP：The output power exceeds the rated value OCP：The output current exceeds the rated value OVP：The output voltage exceeds the set value OPP：The output power exceeds the set value LVP：The output voltage is lower than the set value ALTP：The output mode has changed CMF：External communication protection								

The channel event register records the state change events of the power supply, and the meaning of each binary bit corresponds to the bit of the channel state register one-to-one. The channel event register can be cleared by the relevant query command or by the "*cls" command, which will restart the recording of new events.

Standard event register group

The Standard Event Register is the register group that records the important events occurring in the process of parsing programming commands or performing operations of the power supply. It includes the Standard Event register and the Standard Event Enable register.

Each definition of the standard event register is compatible with the IEEE 488.2 standard and is defined as follows:

Table 2-3 Standard event register bit definition

Bit	7	6	5	4	3	2	1	0
Name	rsv	rsv	CME	EXE	DDE	QYE	rsv	OPC
OPC All operations or				EXE Execution error				

commands have been completed QYE Query error DDE Equipment related error	CME Command parsing errors

The value of the standard event register can be cleared by the relevant query command or by the "*CLS" command, after which the recording of new events will start again. Standard events that can be used to set the standard events registers the events of a byte is reported to the state register of the ESB, as shown in figure 2 - 2 state of the power supply system.

Status byte register

The Status Byte Register (Status Byte Register) records the important states that the IEEE 488.2 bus compatible device needs to support. The status bit records whether the power supply currently has unserved events, errors, standard events, etc., and its definition is fully compatible with the IEEE 488.2 specification. The status byte registers are defined as follows:

Table 2-4 Status byte register bit definition

bit			6	5	4	3	2	0-1
name			RQS	ESB	MAV	rsv	CSUM	rsv
CSUM There is a channel state event MAV There is available information, that is, the output queue is not empty ESB There are standard events RQS Request service				rsv Reserved bit				

2.5. Command version information

The version information of a programming command can only be queried remotely. Programming commands can be sent through a remote interface

SYSTem:VERSion?

To query the programming command version of the power supply.

The return string format is "YYYY.v", where "YYYY" represents the version year and "V" represents the version number.

2.6. Detailed explanation of commands

2.7. IEEE488.2 Public order

Public commands are the common commands required by the IEEE 488.2 specification to be supported by the instrument, which are used to control the common functions of the instrument, such as reset and status query, etc. The syntax and semantics of the commands follow the IEEE 488.2 protocol specification. IEEE 488.2 public commands have no hierarchy.

*CLS

Clear the following registers. Please refer to the state system section for the meaning of each register:

standard event status register

channel events registers

status byte registers

error queue

Command syntax: *CLS

Parameter: None

Query syntax: None

*ESE

Set the Standard Event Status Enable register. The parameter is the enable status of each of the standard event registers. Setting the 1 bit will enable the corresponding event recording function.

Command syntax: *ESE

Parameter: 0 to 255

Reset value: related to the *PSC command, please refer to the *PSC command.

For example, *ESE 128

Query syntax: *ESE?

Return value: (register value)

Related commands: *ESR? *PSC, *STB?

*ESR?

Read the Standard Event register. After executing this command, the standard event register will be cleared to zero. Refer to table 2 - 3 standard events registers a definition.

Query syntax: *ESR?

Parameter: None

Return value: (register value)

Related commands: *CLS, *ESE, *OPC

*IDN?

Read information about the power supply. The return value of this command contains three comma-separated strings whose meanings are, in order, manufacturer, product model, retention, and software version number.

Query syntax: *IDN?

Parameter: None

Returned value: String description

Faith Manufacturer

9000A Product model

0 Retention

XX.XX software version

return value sample：Faith,9000A,0,V1.00

*OPC

Set the OPC position of the Standard Event register to "1". If the OPC bit is 1, all operations and commands initiated before the command are completed.

Command syntax: *OPC

Parameter: None

Query syntax: *OPC?

Return value:

Related commands: *TRG *WAI

*PSC

Control whether the service request enable register and the standard event enable register are cleared when the power is turned on, defined as follows:
1: disables saving the values of the service request enable register and the standard event enable register. On startup, the power supply zeroes out the values of the service request enable register and the standard event enable register.
0: Save the values of the service request enable register and the standard event enable register and invoke them at the next startup.

Command syntax: *PSC
Parameters: 0 | 1
For example: *PSC 1
Query syntax: *PSC?
The return value: OFF | ON the current state of the PSC (*)
Related commands: *ESE, *SRE

*RST

factory data reset
Command syntax: *RST
Parameter: None
Returned value: None
Related commands: None

*SRE

Set the Service Request Enable Register. The enable bit of this Register is used to enable the generation condition of the RQS bit in the Status Byte Register. Please refer to the Status byte Register section. If one of the status bytes is "1" and the corresponding service request enablement register bit is also "1", then the RQS bit in the status byte register will be set to "1".

Command syntax: *SRE

Parameter: 0 to 255

Query syntax: *SRE?

Return value: (* Current status of SRE command)

Related commands: *ESE *ESR *PSC

*STB?

Query the Status Byte register. Executing this command will clear the status byte register.

Please refer to table 2 - state 4 byte register a definition.

Command syntax: *STB?

Parameter: None

Return value:

Related commands: *CLS *ESE *ESR

*SAV

Save the configuration parameters to the specified storage locations (storage locations 1 to 20).

Command syntax: *SAV

Parameter: 1 to 20

Example: *SAV 3

Query syntax: None

Related commands: *RCL

◎**Note:**

*** The SAV command takes about 500 milliseconds to execute. Do not perform any operation when executing this command.**

---

*RCL

Read the configuration parameters from the specified storage location (storage locations 1 to 20).

Command syntax: *RCL

Parameter: 1 to 20

Example: *RCL 3

Query syntax: None

Related commands: *SAV

◎**Note:**

*** The RCL command takes about 500 milliseconds to execute. Do not perform any operation when executing this command.**

---

*TST?

The operation performs a self-check and returns the self-check information. If the command output is -1, it indicates that the power self-test fails. In this case, run the SYSTem:ERRor? Command. Query self-check error codes. The specific meanings of error codes and error messages are described in the appendix.

Command syntax: *TST?

Parameter: None

Return value:

Related commands: None

*WAI

After this command is executed, the power supply does not process other commands until the current operation is complete.

Command syntax: *WAI

Parameter: None

Related commands: *OPC

2.8. CONFigure COMAMND

CONFigure:APG:VOLTage

Turn on or off the power supply analog programming voltage.

Command syntax: CONFigure:APG:VOLTage

Parameters: 0 ON | OFF | | 1

Unit: None

Example: CONF:APG:VOLT 1

Query syntax: CONFigure:APG:VOLTage?

Return value:<NR1>

CONFigure:APG:CURRent

Turn on or off power to simulate programming current.

Command syntax: CONFigure:APG:CURRent

Parameters: 0 ON | OFF | | 1

Unit: None

Example: CONF:APG:CURR 1

Query syntax: CONFigure:APG:CURRent?

Return value:<NR1>

CONFigure:APG:POWer

Turn on or off power analog programming power.

Command syntax: CONFigure:APG:POWer

Parameters: 0 ON | OFF | | 1

Unit: None

Example: CONF:APG:POW 1

Query syntax: CONFigure:APG:POWer?

Return value:<NR1>

CONFigure:INHibit

Set the external control mode for power output.

Command syntax: CONFigure:INHibit

Parameters: 0 | 1 | 2 | TRIGGER | TOGGLE | HOLD

Unit: None

Example: CONF:INH 1

Query syntax: CONFigure:INHibit?

Return value:<NR1>

CONFigure:PRIor

Select voltage and current preferred.

Command syntax: CONFigure:PRIor

Parameters: 0 | 1 | VOLTAGE | CURRENT

Unit: None

For example, CONF:PRI 1

Query syntax: CONFigure:PRIor?

Return value:<NR1>

CONFigure:MSSL:ID

Set the identity attribute of the power supply when cascading. This parameter also serves as the cascade address.

CONFigure:MSSL:ID

Parameters: 0 | 1 | 2 | 3 | 4 | MASTER | SLAVE1 | SLAVE2 | SLAVE3 | SLAVE4

Example: CONF:MSSL:ID 1

Query syntax: CONFigure:MSSL:ID?

The return value: MASTER | SLAVE1 | SLAVE2 | SLAVE3 | SLAVE4

CONFigure:MSSL:NUMSLV

Set the number of slave machines when cascading.

Syntax: CONFigure:MSSL:NUMSLV

Parameter: 1 to 4

Example: CONF:MSSL:NUMSLV 1

Query syntax: CONFigure:MSSL:NUMSLV?

Return value:<NR1>

CONFigure:MSSL:CONTrol

Enable or disable cascade expansion.

Command syntax: CONFigure:MSSL:CONTrol

Parameters: 0 ON | OFF | | 1

Example: CONF:MSSL:CONT 1

Query syntax: CONFigure:MSSL:CONTrol?

The return value: OFF | ON

2.9. OUTPut COMMAND

OUTPut[:STATe]

Turn output on or off. The start and stop of the test function is also controlled by this command.

Command syntax: OUTPut[:STATe]

Parameters: 0 ON | OFF | | 1

For example, OUTP 1

Query syntax: OUTPut[:STATe]?

The return value: OFF | ON

OUTPut:FUNCtion

Switch power test function.

Command syntax: OUTPut:FUNCtion

Parameters: 0 | 1 | 2 | 3 | 4 VIP | | SEQ | CR | CHARGER | DISCHARGER

Example: OUTP:FUNC VIP

Query syntax: OUTPut:FUNCtion?

Return values: 0 | 2 | 3 | 1

OUTPut:PROTection:VOLTage[:LEVel]

Set the overvoltage protection value of the power supply.

Command syntax: OUTPut: PROTection: VOLTage: [LEVel] < the NRf >

Parameter: MIN to MAX

Unit: V (volt)

Example: OUTP:PROT:VOLT 10

The query syntax: OUTPut: PROTection: VOLTage: [LEVel]?

Return value:<NRf> [Unit=V]

OUTPut:PROTection:CURRent[:LEVel]

Set the overcurrent protection value of the power supply.

Command syntax: OUTPut: PROTection: the CURRent [: LEVel] < the NRf >

Parameter: MIN to MAX

Unit: A (ampere)

Example: OUTP:PROT:CURR 20

The query syntax: OUTPut: PROTection: the CURRent [LEVel]?

Return value:<NRf> [Unit=A]

OUTPut:PROTection:POWer[:LEVel]

Set the overpower protection value for the PSU.

Command syntax: OUTPut:PROTection:POWer[:LEVel]

Parameter: MIN to MAX

Unit: W (watt)

Example: OUTP:PROT:POW 1000

OUTPut:PROTection:POWer[:LEVel]?

Return value:<NRf> [Unit=W]

OUTPut:PROTection:LVP[:LEVel]

Set the undervoltage protection value of the power supply.

Command syntax: OUTPut:PROTection:LVP[:LEVel]

Parameter: MIN to MAX

Unit: V (volt)

For example, OUTP:PROT:LVP 20

Query syntax: OUTPut:PROTection:LVP[:LEVel]?

Return value: <NRf>[Unit=V]

OUTPut:PROTection:CMF[:LEVel]

Set the communication timeout time of the power supply.

Command syntax: OUTPut:PROTection:CMF[:LEVel]

Parameter: MIN to MAX

Unit: s (second)

Example: OUTP:PROT:CMF 5

OUTPut:PROTection:CMF[:LEVel]?

Return value: <NRf>[Unit=s]

OUTPut:PROTection:RVP[:LEVel]

Turn on or off reverse connection detection.

Command syntax: OUTPut:PROTection:RVP[:LEVel]

Parameters: 0 ON | OFF | | 1

For example, OUTP:PROT:RVP 1

OUTPut:PROTection:RVP[:LEVel]?

The return value: OFF | ON

OUTPut:PROTection:CLEar

Clear the protection or fault of the power supply.

Command syntax: OUTPut:PROTection:CLEar

Parameter: None

Example: OUTP:PROT:CLE

Query syntax: None

2.10. FUNCtion COMMAND

2.11. FUNCtion

Switch power test function.

Command syntax: FUNCtion

Parameters: 0 | 1 | 2 | 3 | 4 VIP | | SEQ | rsei | the RAMP

Example: FUNCtion VIP

Query syntax: FUNCtion?

Return values: 0 | 2 | 3 | 1

2.12. SOURce COMMAND

SOURce:VOLTage[:LEVel]

Set the output voltage.

Command syntax: SOURce:VOLTage[:LEVel]

Parameter: MIN to MAX

Unit: V (volt)

Example: SOUR:VOLT 30

Query syntax: SOURce:VOLTage[:LEVel]?

Return value: <NRf>[Unit=V]

SOURce:VOLTage:SLEW:RISE

Set the voltage rise slope.

Command syntax: SOURce:VOLTage:SLEW:RISE

Parameter: MIN to MAX

Unit: V/s (volt/second)

Example: SOUR:VOLT:SLEW:RISE 5000

Query syntax: SOURce:VOLTage:SLEW:RISE?

Return value: <NRf>[Unit=V/s]

SOURce:VOLTage:SLEW:FALL

Set the voltage drop slope.

Command syntax: SOURce:VOLTage:SLEW:FALL

Parameter: MIN to MAX

Unit: V/s (volt/second)

Example: SOUR:VOLT:SLEW:FALL 1000

Query syntax: SOURce:VOLTage:SLEW:FALL?

Return value:<NRf> [Unit=V/s]

SOURce:VOLTage:LIMit:HIGH

The upper limit of the voltage output is set to prevent the user from misinput to protect the device under test.

Command syntax: SOURce, VOLTage: LIMit: HIGH < the NRf >

Parameter: MIN to MAX

Unit: V (volt)

Example: SOUR:VOLT:LIM:HIGH 25

The query syntax: SOURce: VOLTage: LIMit: HIGH?

Return value: <NRf>[Unit=V]

SOURce:VOLTage:LIMit:LOW

The lower limit of the voltage output is set to protect the device under test from incorrect input.

Command syntax: SOURce:VOLTage:LIMit:LOW

Parameter: MIN to MAX

Unit: V (volt)

Example: SOUR:VOLT:LIM:LOW 10

Query syntax: SOURce:VOLTage:LIMit:LOW?

Return value: <NRf>[Unit=V]

SOURce:CURRent[:LEVel]

Set the output current.

Command syntax: SOURce:CURRent[:LEVel]

Parameter: MIN to MAX

Unit: A (ampere)

Example: SOUR:CURR 50.0

Query syntax: SOURce:CURRent[:LEVel]?

Return value: <NRf>[Unit=A]

SOURce:CURRent:SLEW:RISE

Set the current rising slope.

Command syntax: SOURce:CURRent:SLEW:RISE

Parameter: MIN to MAX

Unit: A/s (ampere/second)

Example: SOUR:CURR:SLEW:RISE 2000

Query syntax: SOURce:CURRent:SLEW:RISE?

Return value: <NRf>[Unit=A/s]

SOURce:CURRent:SLEW:FALL

Set the current drop slope.

Command syntax: SOURce:CURRent:SLEW:FALL

Parameter: MIN to MAX

Unit: A/s (ampere/second)

Example: SOUR:CURR:SLEW:FALL 500.0

Query syntax: SOURce:CURRent:SLEW:FALL?

Return value: <NRf>[Unit=A/s]

SOURce:CURRent:LIMit:HIGH

The upper limit of current output is set to prevent users from misinput to protect the device under test.

Command syntax: SOURce: CURRent: LIMit: HIGH < the NRf >

Parameter: MIN to MAX

Unit: A (ampere)

Example: SOUR:CURR:LIM:HIGH 200

The query syntax: SOURce: CURRent: LIMit: HIGH?

Return value: <NRf>[Unit=A]

SOURce:CURRent:LIMit:LOW

The lower limit of current output is set to prevent users from misinput to protect the device under test.

Command syntax: SOURce:CURRent:LIMit:LOW

Parameter: MIN to MAX

Unit: A (ampere)

Example: SOUR:CURR:LIM:LOW 100

Query syntax: SOURce:CURRent:LIMit:LOW?

Return value: <NRf>[Unit=A]

SOURce:POWer[:LEVel]

Set the output power.

Command syntax: SOURce:POWer[:LEVel]

Parameter: MIN to MAX

Unit: W (watt)

Example: SOUR:POW 1000

Query syntax: SOURce:Power[:LEVel]?

Return value: <NRf>[Unit=W]

SOURce:POWer:LIMit:HIGH

The upper limit of the power output is set to protect the device under test from misinput.

Command syntax: SOURce:POWer:LIMit:HIGH

Parameter: MIN to MAX

Unit: W (watt)

Example: SOUR:POW:LIM:HIGH 3000

Query syntax: SOURce:POWer:LIMit:HIGH?

Return value:<NRf> [Unit=W]

SOURce:POWer:LIMit:LOW

Set the lower limit for power output to protect the device under test.

Command syntax: SOURce:POWer:LIMit:LOW

Parameter: MIN to MAX

Unit: W (watt)

Example: SOUR:POW:LIM:LOW 2500

Query syntax: SOURce:POWer:LIMit:LOW?

Return value: <NRf>[Unit=W]

LOAD COMMAND

LOAD:CURRent[:LEVel]

Set the load output current.

Command syntax: LOAD:CURRent[:LEVel]

Parameter: MIN to MAX

Unit: A (ampere)

Example: LOAD:CURR 50.0

Query syntax: LOAD:CURRent[:LEVel]?

Return value:<NRf> [Unit=A]

LOAD:CURRent:SLEW:RISE

Set the rising slope of the load current.

Command syntax: LOAD:CURRent:SLEW:RISE

Parameter: MIN to MAX

Unit: A/s (ampere/second)

Example: LOAD:CURR:SLEW:RISE 2000

Query syntax: LOAD:CURRent:SLEW:RISE?

Return value: <NRf>[Unit=A/s]

LOAD:CURRent:SLEW:FALL

Set the downward slope of load current.

Command syntax: LOAD:CURRent:SLEW:FALL

Parameter: MIN to MAX

Unit: A/s (ampere/second)

Example: LOAD:CURR:SLEW:FALL 500.0

Query syntax: LOAD:CURRent:SLEW:FALL?

Return value:<NRf> [Unit=A/s]

LOAD:CURRent:LIMit:HIGH

The upper limit of the load current output is set to prevent users from misinput, so as to protect the device under test.

Command syntax: LOAD:CURRent:LIMit:HIGH

Parameter: MIN to MAX

Unit: A (ampere)

Example: LOAD:CURR:LIM:HIGH 200

Query syntax: LOAD:CURRent:LIMit:HIGH?

Return value:<NRf> [Unit=A]

LOAD:CURRent:LIMit:LOW

Set the lower limit of the load current output to prevent users from misinput and protect the device under test.

Command syntax: LOAD:CURRent:LIMit:LOW

Parameter: MIN to MAX

Unit: A (ampere)

For example, LOAD:CURR:LIM:LOW 100

Query syntax: LOAD:CURRent:LIMit:LOW?

Return value: <NRf>[Unit=A]

LOAD:POWer[:LEVel]

Load set output power.

Command syntax: LOAD:POWer[:LEVel]

Parameter: MIN to MAX

Unit: W (watt)

Example: LOAD:POW 1000

Query syntax: LOAD:Power[:LEVel]?

Return value: <NRf>[Unit=W]

LOAD:POWer:LIMit:HIGH

The upper limit of the load power output is set to prevent users from misinput to protect the device under test.

Command syntax: LOAD:POWer:LIMit:HIGH

Parameter: MIN to MAX

Unit: W (watt)

Example: LOAD:POW:LIM:HIGH 3000

Query syntax: LOAD:POWer:LIMit:HIGH?

Return value: <NRf>[Unit=W]

LOAD:POWer:LIMit:LOW

Set the lower limit of the load power output to prevent users from misinput and protect the device under test.

Command syntax: LOAD:POWer:LIMit:LOW

Parameter: MIN to MAX

Unit: W (watt)

Example: LOAD:POW:LIM:LOW 2500

Query syntax: LOAD:POWer:LIMit:LOW?

Return value:<NRf> [Unit=W]

2.13. SEQuence COMAMND

2.14. SEQuence:STATus

To query the running status of the current sequence, the return result is the number of running sequence file cycles and running steps.

Command syntax: SEQuence:STATus?

Parameter: None

For example: SEQ:STAT?

Return values: <NR1>,<NR1>,

2.15. SEQuence:RUN:NUMBer

Set the sequence function run file number.

Command syntax: SEQuence:RUN:NUMBer

Parameter: 1 to 20

For example, SEQ:RUN:NUMB 1

Query syntax: SEQuence:RUN:NUMBer?

Return value:<NR1>

2.16. SEQuence:EDIT:NUMBer

Set sequence function Edit file number.

Command syntax: SEQuence:EDIT:NUMBer

Parameter: 1 to 20

For example, SEQ:EDIT:NUMB 1

Query syntax: SEQuence:EDIT:NUMBer?

Return value:<NR1>

2.17. SEQuence:EDIT:COUNt

Sets the file length of the current edit sequence.

Command syntax: SEQuence:EDIT:COUNt

Parameter: 1 to 20

Example: SEQ:EDIT:COUN 10

Query syntax: SEQuence:EDIT:COUNt?

Return value:<NR1>

2.18. SEQuence:EDIT:LINK

Set the link file of the current edit sequence. 0 indicates no link file.

Command syntax: SEQuence:EDIT:LINK

Parameter: 0 to 20

For example, SEQ:EDIT:LINK 0

Query syntax: SEQuence:EDIT:LINK?

Return value:<NR1>

2.19. SEQuence:EDIT:CYCLe

Set the number of runs of the current edit sequence, with 0 indicating infinite runs.

Command syntax: SEQuence:EDIT:CYCLe

Parameter: 0 to 60000

Example: SEQ:EDIT:CYCL 1

Query syntax: SEQuence:EDIT:CYCLe?

Return value:<NR1>

2.20. SEQuence:EDIT:SAVE

Save the sequence file you are editing.

Command syntax: SEQuence:EDIT:SAVE

Parameter: None

For example, SEQ:EDIT:SAVE

Query syntax: None

2.21. SEQuence:EDIT:STEP

Set the editing step of the current edit file.

Command syntax: SEQuence:EDIT:STEP

Parameter: 1 to 20

Example: SEQ:EDIT:STEP 1

Query syntax: SEQuence:EDIT:STEP?

Return value:<NR1>

2.22. SEQuence:EDIT:VOLTage

Sets the output voltage for the current editing step of the sequence file.

Command syntax: SEQuence:EDIT:VOLTage

Parameter: MIN to MAX

Unit: V (volt)

Example: SEQ:EDIT:VOLT 12.0

Query syntax: SEQuence:EDIT:VOLTage?

Return value: <NRf>[Unit=V]

2.23. SEQuence:EDIT:SOURce:CURRent

Sets the source output current for the current editing step of the sequence file.

Command syntax: SEQuence: EDIT: SOURce: CURRent < the NRf >

Parameter: MIN to MAX

Unit: A (ampere)

Example: SEQ:EDIT:SOUR:CURR 50

The query syntax: SEQuence: EDIT: SOUR: CURRent?

Return value:<NRf> [Unit=A]

2.24. SEQuence:EDIT:SOURce:POWer

Sets the source output power for the current editing step of the sequence file.

Command syntax: SEQuence: EDIT: SOURce: POWer < the NRf >

Parameter: MIN to MAX

Unit: W (watt)

Example: SEQ:EDIT:SOUR:POW 50

Query syntax: SEQuence:EDIT:SOUR:POWer?

Return value: <NRf>[Unit=W]

2.25. SEQuence:EDIT:LOAD:CURRent

Sets the load input current for the current editing step of the sequence file.

Command syntax: SEQuence: EDIT: LOAD: CURRent < the NRf >

Parameter: MIN to MAX

Unit: A (ampere)

Example: SEQ:EDIT:LOAD:CURR 50

The query syntax: SEQuence: EDIT: LOAD: CURRent?

Return value:<NRf> [Unit=A]

2.26. SEQuence:EDIT:LOAD:POWer

Sets the load input power for the current edit step of the sequence file.

Command syntax: SEQuence:EDIT:LOAD:POWer

Parameter: MIN to MAX

Unit: W (watt)

For example, SEQ:EDIT:LOAD:POW 50

Query syntax: SEQuence:EDIT:LOAD:POWer?

Return value:<NRf> [Unit=W]

2.27. SEQuence:EDIT:DELay

Set the delay time of the current editing step (delay function).

Command syntax: SEQuence:EDIT:DELay

Parameter: 0.001 to 86400

Unit: s (second)

For example, SEQ:EDIT:DEL 10

Query syntax: SEQuence:EDIT:DELay?

Return value: <NRf>[Unit=s]

2.28. CHG COMMAND

CHG:VOLTage

Set the charging voltage in charging mode.

Command syntax: CHG:VOLTage[:LEVel]

Parameter: MIN to MAX

Unit: V (volt)

Example: CHG:VOLT 20.0

Query syntax: CHG:VOLT[:LEVel]?

Return value: <NRf>[Unit=V]

CHG:CURRent

Set the charging current in charging mode.

Command syntax: CHG:CURRent[:LEVel]

Parameter: MIN to MAX

Unit: A (Ann)

For example, CHG:CURR 10.0

Query syntax: CHG:CURR [:LEVel]?

Return value: <NRf>[Unit=A]

CHG:POWer

Set the charging power in charging mode.

Command syntax: CHG:POWer[:LEVel]

Parameter: MIN to MAX

Unit: W (watt)

For example, CHG:POW100

Query syntax: CHG:POW [:LEVel]?

Return value: <NRf>[Unit=W]

CHG:STOP:VOLTage

Set the stop voltage in charging mode.

Command syntax: CHG:STOP:VOLTage

Parameter: MIN to MAX

Unit: V (volt)

For example, CHG:STOP:VOLT 25.0

Query syntax: CHG:STOP:VOLT?

Return value:<NRf> [Unit=V]

CHG:STOP:CURRent

Set the stop current in charging mode.

Command syntax: CHG:STOP:CURRent

Parameter: MIN to MAX

Unit: A (Ann)

For example, CHG:STOP:CURR 5.0

Query syntax: CHG:STOP:CURR?

Return value: <NRf>[Unit=A]

CHG:STOP:CAPacity

Set the stop voltage in charging mode.

Command syntax: CHG:STOP:CAPacity

Parameter: MIN to MAX

Unit: AH (Ampere)

For example, CHG:STOP:CAP 2500

Query syntax: CHG:STOP:CAP?

Return value: <NR1>[Unit=AH]

CHARge:STOP:TIME

Set the stop current in charging mode.

Command syntax: CHG:STOP:TIME

Parameter: MIN to MAX

Unit: S (second)

For example, CHG:STOP:TIME 10

Query syntax: CHG:STOP:TIME?

Return value:<NR1> [Unit=S]

2.29. DISCharge COMMAND

2.30. DISCharge:MODE

Set the mode selection in discharge mode.

DISCharge:MODE

Parameters:

Unit: None

Example: DISC:MODE 1

Query syntax: DISC:MODE?

Return value:

2.31. DISCharge:CURRent

Set the discharge current in discharge mode.

Command syntax: DISCharge:CURRent[:LEVel]

Parameter: MIN to MAX

Unit: A (Ann)

Example: DISC:CURR 10.0

Query syntax: DISC:CURR [:LEVel]?

Return value: <NRf>[Unit=A]

2.32. DISCharge:RESistance

Simulated resistance value in discharge mode.

DISCharge:RESistance

Parameter: MIN to MAX

Unit: Ω (ohm)

Example: DISC:RES1.0

Query syntax: DISC:RES?

Return value: <NRf>[Unit=Ω]

2.33. DISCharge:POWer

Set the discharge power in discharge mode.

Command syntax: DISCharge:POWer[:LEVel]

Parameter: MIN to MAX

Unit: W (watt)

Example: DISC:POW100

Query syntax: DISC:POW [:LEVel]?

Return value:<NRf> [Unit=W]

2.34. DISCharge:STOP:VOLTage

Set the stop voltage in discharge mode.

DISCharge:STOP:VOLTage

Parameter: MIN to MAX

Unit: V (volt)

Example: DISC:STOP:VOLT 25.0

Query syntax: DISC:STOP:VOLT?

Return value: <NRf>[Unit=V]

2.35. DISCharge:STOP:CAPacity

Set the stop voltage in discharge mode.

DISCharge:STOP:CAPacity

Parameter: MIN to MAX

Unit: AH (Ampere)

Example: DISC:STOP:CAP 2500

Query syntax: DISC:STOP:CAP?

Return value:<NR1> [Unit=AH]

2.36. DISCharge:STOP:TIME

Set the stop current in discharge mode.

DISCharge:STOP:TIME

Parameter: MIN to MAX

Unit: S (second)

Example: DISC:STOP:TIME 10

Query syntax: DISC:STOP:TIME?

Return value: <NR1>[Unit=S]

2.37. RAMP COMMAND

RAMP:VOLTage:BEGin

Set the voltage starting point of the slow rise and slow fall function.

RAMP:VOLTage:BEGin

Parameter: MIN to MAX

Unit: V (volt)

Example: RAMP:VOLT:BEG 55.0

Query syntax: RAMP:VOLTage:BEGin?

Return value:<NRf> [Unit=V]

RAMP:VOLTage:MIDDle

Set the voltage intermediate point of the slow rise and slow fall function.

RAMP:VOLTage:MIDDle

Parameter: MIN to MAX

Unit: V (volt)

For example, RAMP:VOLT:MIDD 75.0

RAMP:VOLTage:MIDDle?

Return value: <NRf>[Unit=V]

RAMP:VOLTage:END

Set the voltage end point of the slow rise and slow fall function.

RAMP:VOLTage:END

Parameter: MIN to MAX

Unit: V (volt)

Example: RAMP:VOLT:END 35.0

Query syntax: RAMP:VOLTage:END?

Return value:<NRf> [Unit=V]

RAMP:CURRent

Set the output current of the slow rise and slow fall function.

RAMP:CURRent

Parameter: MIN to MAX

Unit: A (ampere)

For example, RAMP:CURR 100.0

Query syntax: RAMP:CURRent?

Return value:<NRf> [Unit=A]

RAMP:POWer

Set the output power of the slow rise and slow fall function.

RAMP:POWer

Parameter: MIN to MAX

Unit: W (watt)

Example: RAMP:POW 1000

Query syntax: RAMP:POWer?

Return value: <NRf>[Unit=W]

RAMP:SLEW:BTOM

Set the voltage slope from the start point to the midpoint of the slow rise and slow fall function.

Command syntax: RAMP:SLEW:BTOM

Parameter: MIN to MAX

Unit: V/s (volts per second)

Example: RAMP:SLEW:BTOM 100

Query syntax: RAMP:SLEW:BTOM?

Return value:<NRf> [Unit=V/s]

RAMP:SLEW:MTOE

Set the voltage slope from the midpoint to the end point of the slow rise and slow fall function.

Command syntax: RAMP:SLEW:MTOE

Parameter: MIN to MAX

Unit: V/s (volts per second)

Example: RAMP:SLEW:MTOE 100

Query syntax: RAMP:SLEW:MTOE?

2.38. MEASure COMMAND

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

Query the sampling voltage.

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

Parameter: None

Example: MEAS:VOLT?

Return value: <NR2>[Unit=V]

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

Query the echo current.

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

Parameter: None

For example: MEAS:CURR?

Return value: <NR2>[Unit=A]

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

Query echo power.

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

Parameter: None

For example: MEAS:POW?

Return value:<NR2> [Unit=W]

2.42. FETCh COMMAND

2.43. FETCh:VOLTage?

Query the sampling voltage.

Query syntax: MEASure:VOLTage?

Return value:<NR2> [Unit=V]

2.44. FETCh:CURRent?

Query the sampling current.

Query syntax: MEASure:CURRent?

Return value: <NR2>[Unit=A]

2.45. FETCh:STATus?

Query the power status.

Query syntax: MEASure:STATus?

Return value:<NR1>

2.46. STATus COMMAND

2.47. STATus:QUEStionable:CONDition?

Query the power channel status register.

The query syntax: STATus: QUEStionable: CONDition?

Example: STAT:QUES:COND?

Return value:<NR1>

2.48. STATus:QUEStionable:ENABle

Sets the mask for the channel event bit to be reported to the status byte.

Command syntax: STATus: QUEStionable: ENABle < NR1 >

Parameter:

Unit: None

For example, STAT:QUES:ENAB 24

The query syntax: STATus: QUEStionable: ENABle?

Return value:<NR1>

2.49. STATus:QUEStionable[:EVENt]?

Query the power channel event register.

Query syntax: STATus:QUEStionable[:EVENt]?

For example: STAT:QUES?

Return value:<NR1>

2.50. STATus:CHANnel:PTRansition/NTRansition

The programmable filter, which determines the type of conversion in the status register(0 to 1 or 1 to 0), sets the relevant bits of the Event register.

Command syntax: STATus: CHANnel: PTRansition/NTRansition < NR1 >

Parameter: 0 to 65535

For example: STAT:CHAN:PTR 4 Set the channel event bits to 0 to 1

STAT:CHAN:NTR 4 Sets the channel event bits to 1-0

Both PTR and NTR are set to 4, indicating that 0 to 1 and 1 to 0 are set

The query syntax: STATus: CHANnel: PTRansition?

STATus:CHANnel:NTRansition?

Return parameter:

Query example: STAT:CHAN:PTR? Ask about the PTRansition channel Settings

Return example: 4

2.51. SYSTem COMMAND

2.52. SYSTem:ERRor?

Read information from the error queue.

Command syntax: SYSTem:ERRor?

Parameter: None

Example: SYST:ERR?

Return values:<NR1>,<SRD> ,

2.53. SYSTem:LOCal

Exit the remote control mode and return the power supply to the local control mode.

Command syntax: SYSTem:LOCal

Parameter: None

For example, SYST:LOC

Query syntax: None

2.54. SYSTem:VERSion?

Read device version information.

Command syntax: SYSTem:VERSion?

Parameter: None

For example: SYST:VERS?

Return value:

2.55. Error message

2.56. Introduction

Any errors that occur during work are recorded in the error queue until the error queue is full. The error message can be read through the panel menu or a programming command.
Errors are retrieved in first-in, first-out order, with the first error returned as the earliest error. Each read removes one from the error queue. If No error exists, that is, the error queue is empty, the power supply returns +0 No error when the query command is sent.

2.57. View error messages

In remote control mode, run the following command to read and clear an error message from the queue:
SYSTem:ERRor?
The command returns a string, such as:
+101 Invalid character
This error message indicates that there are invalid characters in the command string received by the power supply. If all ERRor information is read or no error occurs during the reading, run the SYSTem:ERRor? The command will return the following information:
"+0 No error"
This information indicates that there is no error or all error information has been cleared.
The following sections describe the error codes and meanings of the power supply error messages.

2.58. Grammatical errors

-100 Command error Undefined syntax error

-101 Invalid character Invalid character in program information string

-102 Syntax error Unrecognized commands or data types exist

-103 Invalid separator A separator is requested, but a character that is not a separator is sent

-104 Data type error The current data type does not match the required type

-105 GET not allowed Receive group execution trigger (GET) in program message

-106 Semicolon unwanted One or more redundant semicolons exist

-107 Comma unwanted One or more extra commas in the argument list

-108 Parameter not allowed The number of parameters is beyond the order quantity

-109 Missing parameter The number of parameter is less than the order quantity, or input parameters

-110 Command header error Undefined command header error

-111 Header separator error A non-delimiter character is used where the command header separator is

-112 Program mnemonic too long The command mnemonic is longer than 12 characters

-113 Undefined header The received commands are not defined in this instrument, although they are syntactically structured

-114 Header suffix out of rang The command header suffix is out of scope

-115 Command can not query Commands do not exist in query form

-116 Command must query Commands must be in query form

-120 Numeric data error Undefined numeric data error

-121 Invalid character in number A data character was present in the numeric data that was not accepted by the preceding command

-123 Exponent too large The absolute value of the index exceeds 32,000

-124 Too many digits Decimal data is more than 255 characters long, excluding the leading 0

-128 Numeric data not allowed Receive properly formatted numeric data elements at a location that does not accept them

-130 Suffix error Undefined suffix error

-131 Invalid suffix The suffix does not follow the syntax defined in IEEE 488.2, or the suffix does not fit into E5071C

-134 Suffix too long The suffix is longer than 12 characters

-138 Suffix not allowed Append suffixes to numeric elements that do not allow suffixes

-140 Character data error Undefined keyword data error

-141 Invalid character data Invalid characters were found in the keyword data element, or invalid keywords were received

-144 Character data too long The keyword data contains more than 12 characters

-148 Character data not allowed Receive the correctly formatted keyword data element in the location where the instrument does not accept the keyword data element

-150 String data error Undefined string data error

-151 Invalid string data The string data should have been present, but for some reason, the present string data was invalid

-158 String data not allowed Receive string data elements where this instrument does not accept string data elements

-160 Block data error Undefined block data error

-161 Invalid block data Data blocks are expected, but for some reason they are invalid

-168 Block data not allowed Receive data block elements where this instrument does not accept them

-170 Expression error Undefined expression error

-171 Invalid expression The expression data element is invalid. For example, parentheses are not paired or illegal characters are used

-178 Expression data not allowed Receive expression data elements where this instrument does not accept them

-180 Macro error Undefined macro error

-181 Invalid outside macro definition Macro parameter placeholder "$" encountered outside macro definition

-183 Invalid inside macro definition Macro definition (*DDT,*DMC), the content of the macro syntax error

-184 Macro parameter error The command in the macro definition has the wrong number or type of arguments

2.59. Execution error

-200 Execution error An error is generated that is relevant to the execution and this instrument cannot specify an error message

-220 Parameter error Undefined parameter error

-221 Setting conflict The command parameters were parsed but could not be executed due to the current device status

-222 Data out of range Data out of range

-224 Illegal paramter value Parameter is not one of the list of optional parameters for the current command

-225 Out of memory The available memory in this instrument is not sufficient to perform the selected action

-232 Invalid format Illegal data format

-240 Hardware error Undefined hardware error

-242 Calibration data lost Calibration data loss

-243 NO reference No reference voltage

-256 File name not found File name not found

-259 Not selected file No file is available

-295 Input buffer overflow Input cache overflow

-296 Output buffer overflow Output cache overflow

2.60. Query error

-350 Query overflow Query overflow

-400 Query error Query error

2.61. Programming examples

2.62. Stationary VIP output operation

```
OUTP OFF                    //Switching test functions is only allowed when the output is off
OUTP:FUNC VIP               //Switch to steady-state VIP output
SOUR:VOLT:SLEW:RISE 5000    //Set the voltage rise slope to 5000V/s
SOUR:VOLT:SLEW:FALL 5000    //Set the voltage drop slope to 5000V/s
SOUR:CURR:SLEW:RISE 2000    //Set the current rise slope to 2000A/s
SOUR:CURR:SLEW:FALL 2000    //Set the current downslope to 2000A/s
SOUR:VOLT 10                //Set the output voltage to 10V
SOUR:CURR 10                //Set the output current to 10A
OUTP ON                     //Enable output
SOUR:VOLT 20                //Change the output voltage to 20V
```

2.63. Read and retrieve the echo parameters

```
MEAS:VOLT?                  //Read and retrieve the display voltage
MEAS:CURR?                  //Read and retrieve the display current
MEAS:POW?                   //Read and retrieve echo power
MEAS:VOLT?;CURR?;POW?       //Read voltage, current and power at a time
```

2.64. Edit the SEQ file

```
OUTP OFF                    //Editing the sequence file is allowed only when the output is closed
SEQ:EDIT:NUMB 1             //Edit the sequence 1 file
SEQ:EDIT:COUN 3             //Set the file length to 3 steps
SEQ:EDIT:CYCL 1             //Set the running number to 1
SEQ:EDIT:LINK 0             //Set the link sequence to 0, that is, no link file
SEQ:EDIT:STEP 1             //Edit step 1
SEQ:EDIT:VOLT 12.0          //Set the output voltage to 12V
SEQ:EDIT:CURR 1.0           //Set the output current to 1A
```

SEQ:EDIT:DEL 1.0 //Set the single step delay to 1 s

SEQ:EDIT:STEP 2 //Edit step 2

SEQ:EDIT:VOLT 24.0 //Set the output voltage to 24V

SEQ:EDIT:CURR 2.0 //Set the output current to 2A

SEQ:EDIT:DEL 2.0 //Set the single-step delay to 2s

SEQ:EDIT:STEP 3 //Edit step 3

SEQ:EDIT:VOLT 36.0 //Set the output voltage to 36V

SEQ:EDIT:CURR 3.0 //Set the output current to 3A

SEQ:EDIT:DEL 3.0 //Set the single-step delay to 3s

SEQ:EDIT:SAVE //Save the current edited sequence file

2.65. Run the SEQ file

OUTP OFF //Switching test functions is allowed only when output is closed

OUTP:FUNC SEQ //Switch to the SEQ output function

SEQ:RUN:NUMB 1 //Select the number 1 sequence file

OUTP ON //Enable output

2.66. Battery charging CHG output operation

OUTP OFF //Switching test functions is only allowed when the output is off

OUTP:FUNC SEQ //Switch to the CHG output function

CHG:VOLT50.0 //Set the charging output voltage to 50.0V

CHG:CURR20.0 //Set the charging output current to 20.0A

CHG:POW 1000 //Set the charging output power to 1000W

CHG:STOP:VOLT 45.0 //Set the charging termination voltage to 45.0V

CHG:STOP:CURR 1.0 //Set the charging stop current to 1.0A

CHG:STOP:CAP 2500 //Set the charge termination power to 2500Ah

CHG:STOP:TIME 1500 //Set the charging termination time to 1500Sec

2.67. Battery discharge DISC output operation

OUTP OFF //Switching test functions is allowed only when output is closed

OUTP:FUNC DISC //Switch to DISC output function

DISC:CURR10.0 //Set the discharge current to 10A

DISC:POW 500 //Set the discharge output power to 500W

DISC:STOP:VOLT 15.0 //Set the discharge termination voltage to 15V

DISC:STOP:CAP 1500 //Set the discharge stopping power to 1500Ah

DISC:STOP:TIME 100 //Set the discharge termination time to 100Sec

2.68. Slow down the RAMP output operations

OUTP OFF //Switching test functions is allowed only when output is closed

OUTP:FUNC RAMP //Switch to the output function of slow rise and slow fall

RAMP:VOLT:BEG 10 //Set the starting voltage to 10V

RAMP:VOLT:MIDD 60 //Set the intermediate point voltage to 60V

RAMP:VOLT:END 20 //Set the end point current to 20V

RAMP:CURR 100.0 //Set the maximum output current to 100A

RAMP:POW 10000 //Set the maximum output power to 10000W

RAMP:SLEW:BTOM 1 //Set the voltage slope from the starting point to the middle point as 1V/s

RAMP:SLEW:MTOE 2 //Set the voltage slope from the middle point to the end point to 2V/s

OUTP ON //Enable output