BENCHTOP INSTRUMENT

Single Output Programmable DC Power Supply Programming Manual

Contents

1. Single-output Programmable DC Power Supply Programming Manual

This manual provides users with guidelines in writing commands for remote control of the single-output programmable DC power supply. It is considered that readers of this manual have carefully read the User Operation Manual of this product and are familiar with operation methods this product.

The power supply communicates with PC via RS232 or USB interface. For details of communication methods, please refer to different chapters of this manual.

After the power supply is reliably connected to PC, user can control the power supply via programming commands. All commands are sent to the power supply from the PC as an ASCII string and will be recognized by the PC to achieve remote operation, control and extended development.

SCPI Commands Introduction
Command System
Command Quick Reference A-Z

2. SCPI Commands Introduction

SCPI (Standard Commands for Programmable Instrument) is standard commands for programmable instruments based on IEEE 488.2 common commands. SCPI commands include two parts: IEEE 488.2 Common Commands and Control Commands defined for SCPI instruments.

Common Commands are defined by IEEE 488.2, which the instrument must support. Syntax and semantics of common commands must follow the application of IEEE 488.2. Common commands work independently of measurement and are generally used for controlling reset, self-test and status operations. For more details, please refer to chapter IEEE 488.2 Common Commands.

Control Commands defined for SCPI instruments are used to measure, read data, switch ON/OFF a function or a mode and so on. This chapter includes the following contents:

Syntax

Symbol Description

Parameter Type

Command Abbreviation

2.1 Syntax

The SCPI commands is to be seen as a tree, originating at the root keyword and dispering into different branches depending on the function required. Each sub-system contains a root keyword and one or more sub-kerwords. A command usually starts with a colon ":" (it can omitted). Two keywords are separated by a colon ":". Parameters are permitted to follow a keyqord. A command ending with a question mark "?" means for query. Command and parameter are separated by a blank space.

For example:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]  {<current>|MINimum|MAXimum}
[: SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?  [MINimum|MAXimum]
```

From the above commands we can see that: SOURce is root keyword of the command. CURRent, LEVel, IMMediate and AMPLitude are keywords of the 2nd, 3rd, 4th and 5th levels respectively. The command starts with a colon ":"; keywords of each levels are separated by a colon ":"; <current> means the parameter is assignable; question mark "?" means for query.
Commands [:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] and parameter <current> are separated by a blank space.

A comma "," is generally used for separating different parameters contained in the same command.

2.2 Symbol Description

The following four symbols are not contents of the SCPI commands. However, they are usually used in instances of the parameters from an Explanation command.

Braces {}

Parameters enclosed in braces are optional. Only one of them should be chosen. For example:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]  {<current>|MINimum|MAXimum}
```

In this command, {<current>|MINimum|MAXimum} means that "MINimum", "MAXimum", self-defined value <current> are parameters, and only one of them can be chosen.

Vertical Bar |

Vertical bar is used to separate two or more optional parameters. Before sending commands, only one of the parameters can be chosen. For example:

```
[:]SYSTem:BEEPer:STATe  {ON|OFF}
```

In this command, users can choose a parameter command from "ON" or "OFF".

Triangle Brackets <>

Parameters enclosed in triangle brackets must be replaced by an effective value. For example:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]  {<current>|MINimum|MAXimum}
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]  5
```

In the above command, “5” is the defined value for current.

Square Brackets [ ]

Contents (keywords) enclosed in square brackets are optional. They would be executed regardless of the square brackets are omitted or not. For example:

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

Sending the follow six commands, the power supply will give same response:

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

2.3 Parameter Type

Parameters introduced in this manual are categorized into 4 types: Boolean, Keyword, Consecutive Integer, and Consecutive Real Number.

Boolean

Parameter should be "ON" or "OFF". For example:

```
[:]SYSTem:BEEPer:STATe  {ON|OFF}
```

Keyword

Parameter value is values as listed in the command. For example:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]   {<current>|MINimum|MAXimum}
```

Parameter is "MINimum"or "MAXimum".

Consecutive Integer

Unless specified, parameters can be any integer within the valid range.

Remarks: Parameters cannot be set with numbers behind decimal points. Otherwise, there will be abnormal response.

For example:

```
*RCL {1|2|3...}
```

Parameter can be any integer number within 0 to 99.

Consecutive Real Number

According to the request of accuracy (normally the default accuracy is to take 6 digits of effective values behind decimal points), parameters can be any value within the valid range. For example:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]   {<current>|MINimum|MAXimum}
```

Parameter {<current> can be any real number within 0 to 10.

2.4 Command Abbreviation

All commands are case insensitive. Commands can be written in all uppercase or lowercase. If commands need to be written in abbreviation, all capital letters specified in a command must be completely entered. For example:

```
[:]SYSTem:BEEPer:STATe?
```

The above command can be abbreviated as:

```
SYST:BEEP:STAT?
SysT:Beep:STAT?
syst:beep:state?
```

3. Command System

This chapter will introduce commands system in the order of command's function.
Commands supported by this product are mainly categorized into the following types:

IEEE 488.2 Common Commands
APPLy Commands
CALibration Commands
HOTKey Commands
MEASure Commands
OUTPut Commands
SOURce Commands
STATus Commands
SYSTem Commands

Remarks: In this command system, all commands related to setup commands such as voltage, current, it is allowed to be followed with parameter's unit. Different parameter's units and omitted units are listed as below:

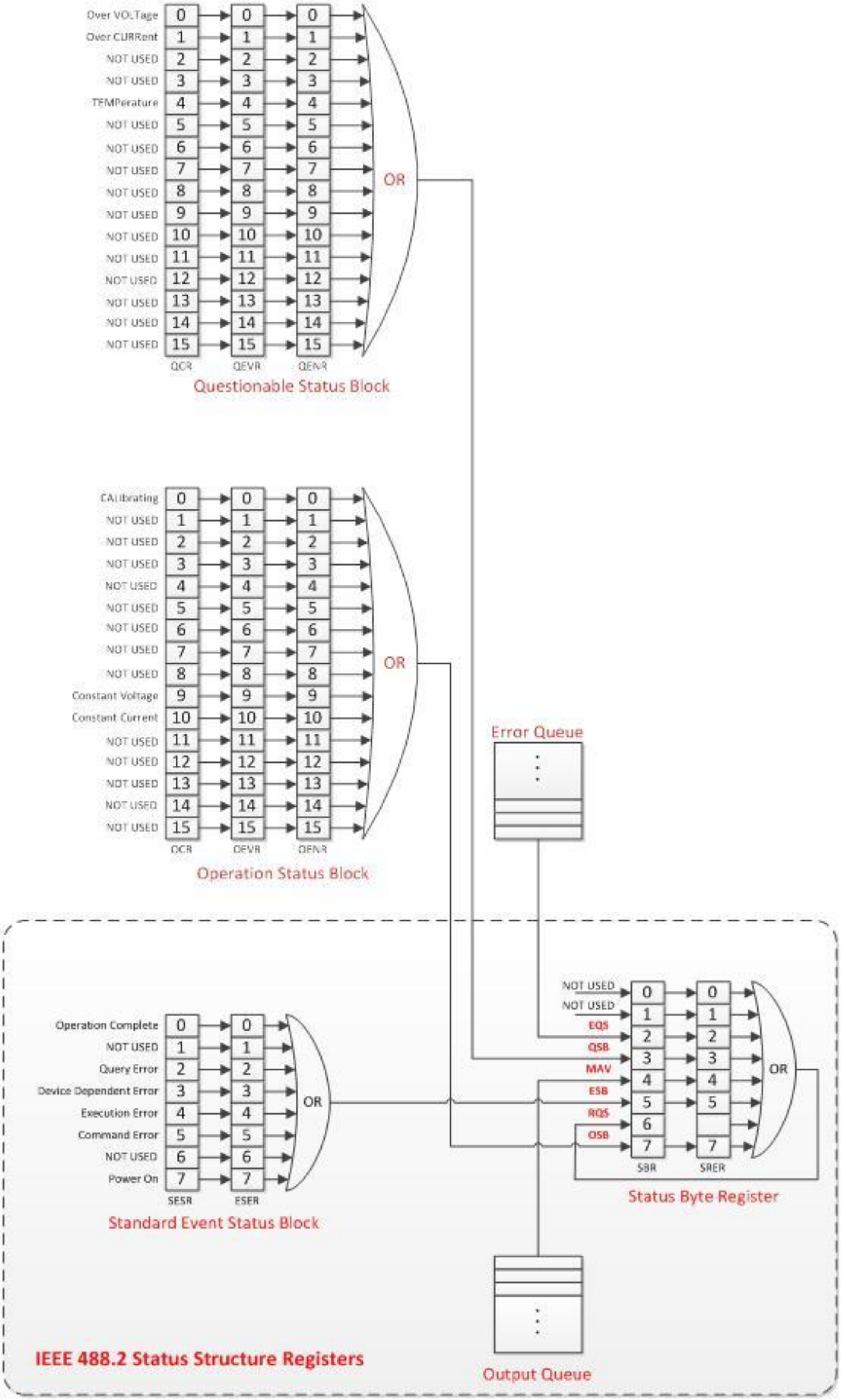
Parameter	Supported Unit	Omitted Unit
Voltage	V	V
Current	A	A
Cycle numbers	CYC	CYC
Time	s	s

3.1 IEEE 488.2 Common Commands

IEEE 488.2 standard defines a common command set for querying or executing some basic operations. These commands usually start with a “*” and hold a keyword that is 3 characters long.

***CLS**
***ESE**
***ESR?**
***IDN?**
***OPC**
***PSC**
***RCL**
***RST**
***SAV**
***SRE**
***STB?**

In the IEEE 488.2 Common Commands and STATus Commands, some of the commands are generally used for setup, query and operation status registers. The principle of the register is shown as below figure, which contains Questionable Status Register groups, Operation Status Register groups, Standard Event Status Register (SESR), Status Byte Register (SBR), Output Buffer, and Error Queue. Some of the commands from IEEE 488.2 and SYSTem sub-system can operate Standard Event Status Register, Status Byte Register, Output Buffer, and Error Queue. STATus sub-system commands are used to set up and query Questionable Status Register groups as well as Operation Status Register groups.


Over VOLTage
Over CURRent
NOT USED
NOT USED
TEMPerature
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
QCR
QEVR
QENR
OR
Questionable Status Block
CALibrating
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
Constant Voltage
Constant Current
NOT USED
NOT USED
NOT USED
NOT USED
NOT USED
OCR
OEVR
OENR
OR
Operation Status Block
Error Queue
Operation Complete
NOT USED
Query Error
Device Dependent Error
Execution Error
Command Error
NOT USED
Power On
SESR
ESER
OR
Standard Event Status Block
NOT USED
NOT USED
EQS
QSB
MAV
ESB
RQS
OSB
SBR
SRER
OR
Status Byte Register
Output Queue
IEEE 488.2 Status Structure Registers

3.1.1 *CLS

Syntax:

*CLS

Function:

This command clears the event registers in all register groups. This command also clears the Error queue.

Example:

```
*CLS
```

3.1.2 *ESE

Syntax:

*ESE <value>
*ESE?

Function:

Set the bits in the Event Status Enable Register (ESER).
Query the bits in the Event Status Enable Register (ESER)

Parameters:

Name	Type	Range	Default
<value>	Consecutive Integer	0 to 255	0

Remarks:

Different bits of the Event Status Enable Register (ESER) are defined as in the following table. Bit 1 and Bit 6 are unused and are always regarded as 0. Parameter <value> is a number from 0 (00000000, in decimal value as 0) to 255 (11111111, in decimal value as 255) of the binary bits of the ESER.

Bit	Weights	Name	Enable
7	128	PON	Power On
6	Not Used	--	Not Used
5	32	CME	Command Error
4	16	EXE	Execution Error
3	8	DOE	Device-Specific Error
2	4	QYE	Query Error
1	Not Used	--	Not Used
0	1	OPC	Operation Complete

Examples:

```
*ESE 128
```

Returns: Enable bit 7 (in decimal value as 128) of the ESER.

```
*ESE?
```

Returns:“128”

3.1.3 *ESR

Syntax:

*ESR?

Function:

Return the contents of the Standard Event Status Register (SESR).

Remarks:

Different bits of the Standard Event Status Register SESR) are defined as in the following table. Bit 1 and Bit 6 are unused and are always regarded as 0. Parameter <value> is a number from 0 (00000000, in decimal value as 0) to 255 (11111111, in decimal value as 255) of the binary bits of the SESR.

Bit	Weights	Name	Enable
7	128	PON	Power On
6	Not Used	--	Not Used
5	32	CME	Command Error
4	16	EXE	Execution Error
3	8	DOE	Device-Specific Error
2	4	QYE	Query Error
1	Not Used	--	Not Used
0	1	OPC	Operation Complete

Examples:

*ESR?

Returns: “128” (Bit 7 is already setup)

3.1.4 *IDN

Syntax:

*IDN?

Function:

Query the instrument ID and return a string (unique identification code of the instrument).

Examples:

*IDN?

Returns: 00000002030400

Remarks:

Different models have different returned value.

3.1.5 *OPC

Syntax:

*OPC
*OPC?

Function:

The command form (*OPC) sets the Operation Complete bit (bit 0) in the Standard Event Status Register (SESR) in bit 1 when all pending operations are finished.

The query form (*OPC?) returns the pending operation is finished or not.

Examples:

*OPC

*OPC?

Returns: If pending operations are finished, returns “1”
If pending operations are not finished, returns “0”.

3.1.6 *PSC

Syntax:

*PSC <value>

*PSC?

Function:

Set the Power-on-Status-Clean-Flag.

Query the Power-on-Status-Clean-Flag.

Parameters:

Parameter value	Power-on-Status-Clean-Flag
0	When power on, the registers will not be removed.
1	When power on, the registers will be removed.

Examples:

*PSC 0

When power on, the registers will not be removed.

*PSC?

Returns: 0

3.1.7 *RCL

Syntax:

*RCL {0|1|2...|99}

Function:

Recall the saved instrument status.

Examples:

*RCL 3

Recall the setting data saved in memroy location 3.

3.1.8 *RST

Syntax:

*RST

Function:

Restore the instrument to the defaults.

Examples:

```
*RST
```

3.1.9 *SAV

Syntax:

*SAV {0|1|2...|99}

Function:

Save the current system status into the non-volatile memory with a specified location.

Examples:

```
*SAV 3
```

Save the current system status into memory location 3.

3.1.10 *SRE

Syntax:

*SRE <value>
*SRE?

Function:

Set the bits in the Service Request Enable Register (SRER).
Query the bits in the Service Request Enable Register (SRER).

Parameters:

Name	Type	Range	Default value
<value>	Consecutive Integer	0 to 255	0

Remarks:

Different bits of the SRER are defined as in below table. Bit 0 and bit 1 are not used and are always regarded as 0. No need to set bit 6. Parameter <value> is a number from 0 (00000000, in decimal value as 0) to 255 (11111111, in decimal value as 255) of the binary bits of the SRER.

Bit	Weights	Name	Enable
7	128	OPE	Standard Operation Summary
6	64	RQS	Request Service
5	32	ESB	Standard Event Summary
4	16	MAV	Message Available Summary
3	8	QUES	Questionable Data Summary
2	4	EQ	Error Queue
1	Not Used	--	Not Used
0	Not Used	--	Not Used

Examples:

*SRE 16

Enable bit 4 (in decimal number as 16) of the SRER.

*SRE?

Returns: 16

3.1.11 *STB

Syntax:

*STB?

Function:

The query returns the contents of the Status Byte Register (SBR).

Remarks:

Different bits of the SBR are defined as in below table. Bit 0 and bit 1 are not used and are always regarded as 0.

Bit	Weights	Name	Enable
7	128	OPE	Standard Operation Summary
6	64	RQS	Request Service
5	32	ESB	Standard Event Summary
4	16	MAV	Message Available Summary
3	8	QUES	Questionable Data Summary
2	4	EQ	Error Queue
1	Not Used	--	Not Used
0	Not Used	--	Not Used

Examples:

*STB?

Returns: 4 (bit 2 is set up)

3.2 APPLy Commands

The APPLy Commands are used to quickly set or query voltageand current value of the power supply.

3.2.1 [:]APPLy

Syntax:

```
[:]APPLy {<voltage>|DEFault|MINimum|MAXimum}[,{<current>|DEFault|MINimum|MAXimum}]
[:]APPLy?
```

Function:

Set the output voltage and current value.
Query the output voltage and current value.

Examples:

```
:APPL 5,1
```

Set the output voltage at 5V and output current at 1A.

```
APPL MAX,MAX
```

Set the output voltage and current at MAX value.

```
:APPL?
```

Retruns the present output voltage and current.

Related commands:

None

3.3 CALibration Commands

The CALibration Commands are used for calibration of the power supply. The CALibration Commands includes the following:

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:CODE
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

3.3.1 [:]CALibration:COUNt?

Syntax:

[:]CALibration:COUNt?

Function:

The query returns the number of times that the power supply has been calibrated.

Examples:

```
:CAL:COUN?
```

Retruns:1(it means that the power supply has been calibrated for 1 time.)

Related commands:

[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:CODE
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

3.3.2 [:]CALibration:CURRent[:DATA]

Syntax:

[:]CALibration:CURRent[:DATA] <value>

Function:

Set the current value of the current calibration point.

Before using this command, it needs to set the calibration point of the current calibration, using command “CALibration:CURRent:LEVel”.

Examples:

:CAL:CURR <value>

Set current value of the calibrationpoint as <value>.

Related commands:

[:]CALibration:COUNt?

[:]CALibration:CURRent:LEVel

[:]CALibration:SECure:CODE

[:]CALibration:SECure:STATe

[:]CALibration:VOLTage[:DATA]

[:]CALibration:VOLTage:LEVel

3.3.3 [:]CALibration:CURRent:LEVel

Syntax:

[:]CALibration:CURRent:LEVel <value>

Function:

Set calibration point of the current calibration.

Examples:

:CAL:CURR:LEV <value>

Set calibration point of the current calibration as <value>.

Related commands:

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:SECure:CODE
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

3.3.4 [:]CALibration:SECure:CODE

Syntax:

[:]CALibration:SECure:CODE <password>

Function:

Set new calibration password (security code). The password must be 4 characters long.

Examples:

:CAL:SEC:CODE <password>

Set new calibration password.

Related commands:

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

3.3.5 [:]CALibration:SECure:STATe

Syntax:

```
[:]CALibration:SECure:STATe {ON | OFF},<password>
[:]CALibration:SECure:STATe?
```

Function:

Turn on or off calibration protection. Calibration of the power supply can only be executed after the calibration protection is turned off.
The query returns calibration protection state.

Examples:

```
:CAL:SEC:STAT ON,<password>
```

Turn on calibration.

```
:CAL:SEC:STAT?
```

Returns:"ON" or "OFF".

Related commands:

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:CODE
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

3.3.6 [:]CALibration:VOLTage[:DATA]

Syntax:

[:]CALibration:VOLTage[:DATA] <value>

Function:

Set voltage value of the voltage calibration point.

Before using this command, it needs to set the voltage calibration point, using command "CALibration:VOLTage:LEVel".

Examples:

:CAL:VOLT <value>

Set the voltage of the voltage calibration point as <value>.

Related commands:

[:]CALibration:COUNt?

[:]CALibration:CURRent[:DATA]

[:]CALibration:CURRent:LEVel

[:]CALibration:SECure:CODE

[:]CALibration:SECure:STATe

[:]CALibration:VOLTage:LEVel

3.3.7 [:]CALibration:VOLTage:LEVel

Syntax:

[:]CALibration:VOLTage:LEVel <value>

Function:

Set the calibration point of voltage calibration.

Examples:

:CAL:VOLT:LEV <value>

Set the voltage calibration point as <value>.

Related commands:

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:CODE
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]

3.4 HOTKey Commands

The HOTKey Commands are used for setting and querying of the hotkey status of the power supply.

3.4.1 [:]HOTKey

Syntax:

[:]HOTKey {ON | OFF}
[:]HOTKey?

Function:

Turn on or off hotkey function.
The query returns hotkey function status.

Examples:

:HOTK ON
Turn on hotkey funciton.

:HOTK?
The query returns hotkey function status.
Return: “ON” or “OFF”.

Related commands;

None

3.5 MEASure Commands

The MEASure Commands are used for query the present values of output voltage, current and power on the power supply’s output terminals. The MEASure Commands includes the following:

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

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

Syntax:

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

Function:

The query returns the value of output current on output terminals.

Examples:

:MEAS:CURR?

Return the value of output current on output terminals.

Related commands:

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

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

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

Syntax:

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

Function:

The query returns the value of output power on output terminals.

Examples:

```
:MEAS:POW?
```

Return the value of output power on output terminals.

Related commands:

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

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

Syntax:

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

Function:

The query returns the value of output voltage on output terminals.

Examples:

:MEAS?

Return the value of output voltage on output terminals.

Related commands:

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

3.6 OUTPut Commands

The OUTPut Commands are used to configure and query the output settings of the power supply.

3.6.1 [:]OUTPut

Syntax:

[:]OUTPut[:STATe] {ON | OFF}
[:]OUTPut[:STATe]?

Function:

Turn on or off the present output.
The query returns “ON” or “OFF”.

Examples:

:OUTP ON
Turn on the output.

:OUTP?
Retruns:“ON” or “OFF”.

Related commands:

None

3.7 SOURce Commands

The SOURce Commands are used to directly set the values of output voltage, output current, output voltage step, output current step, and protection threshold, set the status of OCP and OVP switch and to query related status.

The SOURce Commands includes the following:

```
[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]
```

3.7.1 [:SOURce:]CURRent

The [:SOURce:]CURRent commands are used to set the values of output current, output current step, output current step, OCP threshold, set the status of OCP switch and to query related status.

The [:SOURce:]CURRent commands includes the following:
[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]

3.7.1.1 [:SOURce:]CURRent:PROTection:CLEar

Syntax:

```
[:SOURce:]CURRent:PROTection:CLEar
```

Function:

Clear the present OCP.

Examples:

```
:CURR:PROT:CLE
```

Clear the present OCP.

Related commands:

```
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
```

3.7.1.2 [:SOURce:]CURRent:PROTection:STATe

Syntax:

```
[:SOURce:]CURRent:PROTection:STATe {OFF|ON}
[:SOURce:]CURRent:PROTection:STATe?
```

Function:

Disable or enable the present OCP.

The query returns the present OCP.

Examples:

```
:CURR:PROT:STAT OFF
```

Disable the present OCP.

```
:CURR:PROT:STAT?
```

Query the present OCP.

Return “ON” (enabled) or “OFF” (disabled).

Related commands:

```
[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
```

3.7.1.3 [:SOURce:]CURRent:PROTection:TRIPped?

Syntax:

[:SOURce:]CURRent:PROTection:TRIPped?

Function:

The query returns that the present current protection circuit is in operation or not.

Examples:

:CURR:PROT:TRIP?

Return “ON” (in operation) or “OFF” (not in operation).

Related commands:

[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]

3.7.1.4 [:SOURce:]CURRent:PROTection[:LEVel]

Syntax:

[:SOURce:]CURRent:PROTection[:LEVel] {<current>|DEFault|MINimum|MAXimum}
[:SOURce:]CURRent:PROTection[:LEVel]? {DEFault|MINimum|MAXimum}

Function:

Set the OCP threshold.
The query returns the OCP threshold.

Examples:

```
:CURR:PROT 6
```

Set the OCP threshold at 6A.

```
:CURR:PROT MAX
```

Set the OCP threshold at maximum value.

```
:CURR:PROT?
```

Return the OCP threshold.

```
:CURR:PROT? MAX
```

Return the OCPmaximum value.

Related commands:

[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]

3.7.1.5 [:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]

Syntax:

[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement] {<numeric value>|DEFault|MINimum|MAXimum}
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]? {DEFault|MINimum|MAXimum}

Function:

Set the current step increment.
The query returns the current step increment.

Examples:

:CURR:STEP 1
Set the currentstep increment at 1A.

:CURR:STEP DEF
Set the currentstep increment at default value.

:CURR:STEP?
Return the present current stepincrement.

:CURR:STEP? DEF
Return the present default value of the current stepincrement.

Related commands:

[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]

3.7.1.6 [:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]

Syntax:

```
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]  {<current>|DEFault|MINimum|MAXimum}
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?  {DEFault|MINimum|MAXimum}
```

Function:

Set the output current.
The query returns the present output current value.

Examples:

```
:CURR  1
```

Set the output current at 1A.

```
:CURR  MAX
```

Set the output current at maximum value.

```
:CURR?
```

Return the present output current value.

```
:CURR?  MAX
```

Return the maximum value of the present output current.

Related commands:

```
[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
```

3.7.2 [:SOURce:]VOLTage

The [:SOURce:]VOLTage commands are used to set the values of output voltage, output voltage step, output voltage step, OVP threshold, set the status of OVP switch and to query related status.

The [:SOURce:]VOLTage commands includes the following:
[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]

3.7.2.1 [:SOURce:]VOLTage:PROTection:CLEar

Syntax:

[:SOURce:]VOLTage:PROTection:CLEar

Function:

Clear the OVP mode.

Examples:

:VOLT:PROT:CLE

Clear the OVP mode.

Related commands:

[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]

3.7.2.2 [:SOURce:]VOLTage:PROTection:STATe

Syntax:

[:SOURce:]VOLTage:PROTection:STATe {OFF|ON}
[:SOURce:]VOLTage:PROTection:STATe?

Function:

Enable or disable the present OVP.
The query returns that the present OVP is enabled or disabled.

Examples:

:VOLT:PROT:STAT OFF
Disable the present OVP.

:VOLT:PROT:STAT?
If return "OFF", it means that the OVP is turned off.

Related commands:

[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]

3.7.2.3 [:SOURce:]VOLTage:PROTection:TRIPped?

Syntax:

[:SOURce:]VOLTage:PROTection:TRIPped?

Function:

The query returns that the present OVP is in operation or not.

Examples:

```
:VOLT:PROT:TRIP?
```

Query the present OVP is in operation or not.

Return “ON” (in operation) or “OFF” (not in operation).

Related commands:

[:SOURce:]VOLTage:PROTection:CLEar

[:SOURce:]VOLTage:PROTection:STATe

[:SOURce:]VOLTage:PROTection[:LEVel]

[:SOURce:]VOLTage:RANGe

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

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

3.7.2.4 [:SOURce:]VOLTage:PROTection[:LEVel]

Syntax:

```
[:SOURce:]VOLTage:PROTection[:LEVel]  {<voltage>|DEFault|MINimum|MAXimum}
[:SOURce:]VOLTage:PROTection[:LEVel]?  {DEFault|MINimum|MAXimum}
```

Function:

Set the OVP threshold.

The query returns the present OVP threshold.

Examples:

```
:VOLT:PROT  31
```

Set the OVP threshold at 31V.

```
:VOLT:PROT  MAX
```

Set the OVP threshold at maximum value.

```
:VOLT:PROT?
```

Return the present OVP threshold.

```
:VOLT:PROT?  MAX
```

Return the OVP maximum threshold.

Related commands:

```
[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]
```

3.7.2.5 [:SOURce:]VOLTage:RANGe

Syntax:

[:SOURce:]VOLTage:RANGe {LOW | HIGH}
[:SOURce:]VOLTage:RANGe?

Function:

Set the output voltage range.
The query returns the present output voltage range.

Remarks:

This command is used to choose an output voltage range (only applied to dual-range models.)
“HIGH”, it means that high voltage range is selected.
“LOW” , it means that low voltage range is selected.

Examples:

:VOLT:RANG LOW
Set the output voltage range at low range.

Related commands:

[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]

3.7.2.6 [:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]

Syntax:

```
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]  {<numeric value>|DEFault}
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]?  {DEFault}
```

Function:

Set the output voltage step increment.
The query returns the output voltage step increment.

Examples:

```
:VOLT:STEP 1
```

Set the output voltage increment at 1V.

```
:VOLT:STEP DEF
```

Set the output voltage step increment at default value.

```
:VOLT:STEP?
```

Retrun the present output voltage step increment.

```
:VOLT:STEP? DEF
```

Return the present default value of the output voltage increment.

Related commands:

```
[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]
```

3.7.2.7 [:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]

Syntax:

[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] {<voltage>|DEFault|MINimum|MAXimum}
[:SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? {DEFault|MINimum|MAXimum}

Function:

Set the output voltage.
The query returns the present output voltage.

Examples:

```
:VOLT 5
```

Set the output voltage at 5V.

```
:VOLT MAX
```

Set the output voltage at maximum value.

```
:VOLT?
```

Return the present output voltage.

```
:VOLT? MAX
```

Return the present maximum output voltage.

Related commands:

[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe
[:SOURce:]VOLTage:PROTection:TRIPped?
[:SOURce:]VOLTage:PROTection[:LEVel]
[:SOURce:]VOLTage:RANGe
[:SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]

3.8 STATus Commands

The STATus Commands are used to set and query the contents of the Questionable Status Register group and Operation Status Register group.

The STATus Commands includes the following:
[:]STATus:OPERation:CONDition?
[:]STATus:OPERation:ENABle
[:]STATus:OPERation[:EVENt]?
[:]STATus:PRESet
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable:ENABle
[:]STATus:QUEStionable[:EVENt]?

3.8.1 [:]STATus:OPERation:CONDition?

Syntax:

[:]STATus:OPERation:CONDition?

Function:

This command returns the contents of the Operation Condition Register (OCR).

Examples:

:STAT:OPER:COND?

Return the status of the Operation Condition Register (OCR).

Related commands:

[:]STATus:OPERation:ENABle
[:]STATus:OPERation[:EVENt]?
[:]STATus:PRESet
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable:ENABle
[:]STATus:QUEStionable[:EVENt]?

3.8.2 [:]STATus:OPERation:ENABle

Syntax:

[:]STATus:OPERation:ENABle <enable_value>
[:]STATus:OPERation:ENABle?

Function:

This command sets the contents of the Operation Enable Register (OENR).
This command queries the contents of the Operation Enable Register (OENR).

Examples:

:STAT:OPER:ENAB?
Return the status of the Operation Enable Register (OENR).

Related commands:

[:]STATus:OPERation:CONDition?
[:]STATus:OPERation[:EVENt]?
[:]STATus:PRESet
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable:ENABle
[:]STATus:QUEStionable[:EVENt]?

3.8.3 [:]STATus:OPERation[:EVENt]?

Syntax:

[:]STATus:OPERation[:EVENt]?

Function:

The query returns the status of the Operation Event Register (OEVR).

Examples:

:STAT:OPER?

Return the status of the Operation Event Register (OEVR).

Related commands:

[:]STATus:OPERation:CONDition?
[:]STATus:OPERation:ENABle
[:]STATus:PRESet
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable:ENABle
[:]STATus:QUEStionable[:EVENt]?

3.8.4 [:]STATus:PRESet

Syntax:

[:]STATus:PRESet

Function:

This command is used to clear the Operation Enable Register (OENR) and the Questionable Enable Register (QENR).

Examples:

:STAT:PRES

Related commands:

[:]STATus:OPERation:CONDition?
[:]STATus:OPERation:ENABle
[:]STATus:OPERation[:EVENt]?
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable:ENABle
[:]STATus:QUEStionable[:EVENt]?

3.8.5 [:]STATus:QUEStionable:CONDition?

Syntax:

[:]STATus:QUEStionable:CONDition?

Function:

This command is used to query the status of the Questionable Condition Register (QCR).

Examples:

:STAT:QUES:COND?

Return the status of the Questionable Condition Register (QCR).

Related commands:

[:]STATus:OPERation:CONDition?

[:]STATus:OPERation:ENABle

[:]STATus:OPERation[:EVENt]?

[:]STATus:PRESet

[:]STATus:QUEStionable:ENABle

[:]STATus:QUEStionable[:EVENt]?

3.8.6 [:]STATus:QUEStionable:ENABle

Syntax:

[:]STATus:QUEStionable:ENABle <enable value>
[:]STATus:QUEStionable:ENABle?

Function:

Set the status of the Questionable Enable Register (QENR).
The query returns the status of the Questionable Enable Register (QENR).

Examples:

:STAT:QUES:ENAB?
Return the status of the Questionable Enable Register (QENR).

Related commands:

[:]STATus:OPERation:CONDition?
[:]STATus:OPERation:ENABle
[:]STATus:OPERation[:EVENt]?
[:]STATus:PRESet
[:]STATus:QUEStionable:CONDition?
[:]STATus:QUEStionable[:EVENt]?

3.8.7 [:]STATus:QUEStionable[:EVENt]?

Syntax:

[:]STATus:QUEStionable[:EVENt]?

Function:

This command is used to query the status of the Questionable Event Register (QEVR).

Examples:

:STAT:QUES?

Return the status of the Questionable Event Register (QEVR).

Related commands:

[:]STATus:OPERation:CONDition?

[:]STATus:OPERation:ENABle

[:]STATus:OPERation[:EVENt]?

[:]STATus:PRESet

[:]STATus:QUEStionable:CONDition?

[:]STATus:QUEStionable:ENABle

3.9 SYSTem Commands

The SYSTem Commands are used for controlling beeper, interface setup, error query, remote control and functions related to timing output.

The SYSTem Commands includes the following:

```
[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?
```

3.9.1 [:]SYSTem:AUTO:CYCLe

Syntax:

```
[:]SYSTem:AUTO:CYCLe  {0|1|…|99999}
[:]SYSTem:AUTO:CYCLe?
```

Function:

In timing mode, this command is used to set the number of times of execution. Setting at 0 means for infinite cycling.

In timing mode, this command is used to query the number of times of execution.

Examples:

```
:SYST:AUTO:CYCL  1
```

1 sets auto cycle on to repeat the setting 1 time and it stops after 1 time of execution.

```
:SYST:AUTO:CYCL?
```

Return an integer number within 0 to 99999.

Related commands:

[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.2 [:]SYSTem:AUTO:DELay

Syntax:

[:]SYSTem:AUTO:DELay {1|…|99999}
[:]SYSTem:AUTO:DELay?

Function:

In timming mode, set the delay time under the current responding memory status and then execute the next memory. (The time unit is "s", mminimum resolution is 1s).
In timming mode, this query returns the delay time under the current responding memory status.

Examples:

```
:SYST:AUTO:DEL 1
```

Set auto delay time at 1 second. After 1 second delay, execute the next memory.

```
:SYST:AUTO:DEL?
```

Return an integer number within 1 to 99999. Return 1, means delay 1 seconds at the current memory and then execute the next memroy.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.3 [:]SYSTem:AUTO:STARt

Syntax:

```
[:]SYSTem:AUTO:STARt  {0|1|…|99}
[:]SYSTem:AUTO:STARt?
```

Function:

In timing mode, set the start memory section for auto execution continuoysly.

In timing mode, the query returns the start memory section.

Examples:

```
:SYST:AUTO:STAR  0
```

Set the start memory section from memory 1 for auto execution continuoysly.

```
:SYST:AUTO:STAR?
```

Return an integer number within 0 to 99.

Related commands:

```
[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?
```

3.9.4 [:]SYSTem:AUTO:[STATe]

Syntax:

[:]SYSTem:AUTO:[STATe] {ON | OFF}
[:]SYSTem:AUTO:[STATe]?

Function:

Turn on or off the timing output function.
The query returns the timing output status.

Examples:

```
:SYST:AUTO ON
```

Turn on the timing output function.

```
:SYST:AUTO?
```

Return the timing output status: “ON” (output is on) or “OFF” (output is off).

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.5 [:]SYSTem:AUTO:STOP

Syntax:

```
[:]SYSTem:AUTO:STOP {0|1|…|99}
[:]SYSTem:AUTO:STOP?
```

Function:

In timing mode, set the stop memory section for the auto execution.

In timing mode, return the stop memory section.

Examples:

```
:SYST:AUTO:STOP 10
```

Set the stop memory section in memory 10. The auto execution stops when the execution of memory 10 is done.

```
:SYST:AUTO:STOP?
```

Return an integer number within 0 to 99.

Related commands:

```
[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?
```

3.9.6 [:]SYSTem:BEEPer:STATe

Syntax:

[:]SYSTem:BEEPer:STATe {ON | OFF}
[:]SYSTem:BEEPer:STATe?

Function:

Set the present beeper state.
The query returns the present beeper state.

Examples:

:SYST:BEEP:STAT ON
Turn on the beeper.

:SYST:BEEP:STAT?
Return “ON” (beeper is on) or “OFF (beeper is off)”.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.7 [:]SYSTem:BEEPer[:IMMediate]

Syntax:

[:]SYSTem:BEEPer[:IMMediate]

Function:

To issue a single beep immediately.

Examples:

:SYST:BEEP

The beeper beeps for a single time.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.8 [:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD

Syntax:

```
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
```

Function:

Set baud rate of the power supply’s interface.

Remarks:

A baud rate codes is corresponding to a baud rate as shown in below table:

Code	0	1	2	3	4	5	6	7	8	9
Baud rate	1200	2400	4800	9600	14400	19200	28800	38400	57600	115200

Examples:

```
:SYST:COMM:SER:BAUD  3
```

Set baud rate as 9600.

Related commands:

```
[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?
```

3.9.9 [:]SYSTem:ERRor:COUNt?

Syntax:

[:]SYSTem:ERRor:COUNt?

Function:

The query returns the number of erros records to be read from the power supply.

Remarks:

The maximum recorded error records is 20 sets.

Examples:

:SYST:ERR:COUN?

Return 3, means that there are 3 error records to read from the power supply.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.10 [:]SYSTem:ERRor[:NEXT]?

Syntax:

[:]SYSTem:ERRor[:NEXT]?

Function:

To read the next error message from the power supply.

Remarks:

The possible returned messages are listed as below:

0,"No error"

-100,"Command error"

-108,"Parameter not allowed"

-109,"Missing parameter"

-112,"Program mnemonic too long"

-123,"Exponent too large"

-200,"Execution error"

-220,"Parameter error"

-221,"Settings conflict"

-222,"Data out of range"

-224,"Illegal parameter value"

-350,"Queue overflow"

-702,"Cal secured"

-703,"Invalid secure code"

-704,"Secured code too long"

Examples:

```
:SYST:ERR?
```

Return 0, means there is no error.

Related command:

[:]SYSTem:AUTO:CYCLe

[:]SYSTem:AUTO:DELay

[:]SYSTem:AUTO:STARt

[:]SYSTem:AUTO[:STATe]

[:]SYSTem:AUTO:STOP

[:]SYSTem:BEEPer:STATe

[:]SYSTem:BEEPer[:IMMediate]

[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD

[:]SYSTem:ERRor:COUNt?

[:]SYSTem:LOCal

[:]SYSTem:MEMory?

[:]SYSTem:REMote

[:]SYSTem:RWLock

[:]SYSTem:VERSion?

3.9.11 [:]SYSTem:LOCal

Syntax:

[:]SYSTem:LOCal

Function:

Set the power supply to the local mode.

Examples:

:SYST:LOC

Set the power supply to the local mode. All control knobs onthe front panel are enabled in local mode.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.12 [:]SYSTem:MEMory?

Syntax:

[:]SYSTem:MEMory?

Function:

The query returns the memory section that is curretly recalled.

Examples:

:SYST:MEM?

Return an integer number within 0 to 99.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:REMote
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.13 [:]SYSTem:REMote

Syntax:

[:]SYSTem:REMote

Function:

Set the power supply to remote control mode. In remote control mode, all knobs onthe front panel are disabled,except “Local” and “Output” keys. During remote control, press “Local” key to return the power supply to local mode.

Examples:

:SYST:REM

Set the power supply to remote control mode.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:RWLock
[:]SYSTem:VERSion?

3.9.14 [:]SYSTem:RWLock

Syntax:

[:]SYSTem:RWLock

Function:

Set the power supply to remote control mode. In remote control mode, all knobs onthe front panel are disabled,except “Output” key.

Examples:

:SYST:RWL

Set the power supply to remote control mode.

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:VERSion?

3.9.15 [:]SYSTem:VERSion?

Syntax:

[:]SYSTem:VERSion?

Function:

Return the SCPI version of the power supply.

Examples:

:SYST:VERS?

Return 1999.0

Related commands:

[:]SYSTem:AUTO:CYCLe
[:]SYSTem:AUTO:DELay
[:]SYSTem:AUTO:STARt
[:]SYSTem:AUTO[:STATe]
[:]SYSTem:AUTO:STOP
[:]SYSTem:BEEPer:STATe
[:]SYSTem:BEEPer[:IMMediate]
[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD
[:]SYSTem:ERRor:COUNt?
[:]SYSTem:ERRor[:NEXT]?
[:]SYSTem:LOCal
[:]SYSTem:MEMory?
[:]SYSTem:REMote
[:]SYSTem:RWLock

4. Command Quick Reference A-Z

*CLS
*ESE
*ESR?
*IDN?
*OPC
*PSC
*RCL
*RST
*SAV
*SRE
*STB?

A

[:]APPLy

C

[:]CALibration:COUNt?
[:]CALibration:CURRent[:DATA]
[:]CALibration:CURRent:LEVel
[:]CALibration:SECure:CODE
[:]CALibration:SECure:STATe
[:]CALibration:VOLTage[:DATA]
[:]CALibration:VOLTage:LEVel

H

[:]HOTKey

M

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

O

[:]OUTPut[:STATe]

S

[:SOURce:]CURRent:PROTection:CLEar
[:SOURce:]CURRent:PROTection:STATe
[:SOURce:]CURRent:PROTection:TRIPped?
[:SOURce:]CURRent:PROTection[:LEVel]
[:SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]
[:SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
[:SOURce:]VOLTage:PROTection:CLEar
[:SOURce:]VOLTage:PROTection:STATe

[:SOURce:]VOLTage:PROTection:TRIPped?

[:SOURce:]VOLTage:PROTection[:LEVel]

[:SOURce:]VOLTage:RANGe

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

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

[:]STATus:OPERation:CONDition?

[:]STATus:OPERation:ENABle

[:]STATus:OPERation[:EVENt]?

[:]STATus:PRESet

[:]STATus:QUEStionable:CONDition?

[:]STATus:QUEStionable:ENABle

[:]STATus:QUEStionable[:EVENt]?

[:]SYSTem:AUTO:CYCLe

[:]SYSTem:AUTO:DELay

[:]SYSTem:AUTO:STARt

[:]SYSTem:AUTO[:STATe]

[:]SYSTem:AUTO:STOP

[:]SYSTem:BEEPer:STATe

[:]SYSTem:BEEPer[:IMMediate]

[:]SYSTem:COMMunicate:SERial[:RECeive]:BAUD

[:]SYSTem:ERRor:COUNt?

[:]SYSTem:ERRor[:NEXT]?

[:]SYSTem:LOCal

[:]SYSTem:MEMory?

[:]SYSTem:REMote

[:]SYSTem:RWLock

[:]SYSTem:VERSion?