CONTENTS **PAGE**

GPI/GPT SERIES PROGRAMMABLE EST

PROGRAMMER MANUAL

1. INTRODUCTION

In the modern automatic measurement system, communication between equipments and computers is essential. The measured procedures can be varied with users' testing programs, therefore, the programmable power supply can be operated remotely from an instrument controller or computer across the RS232 interface (standard) or GPIB (optional).

Interface selection and setup

The GPIB address and RS232 baud rate can be changed in normal operation condition. Press [UTILITY] key and knob switch or arrow keys on the front panel, in which the last transmitting interface settings will be displayed. Select interface and press [EDIT/SAVE], then select the baud rate (or GPIB address) and press [FIELD] to confirm the setting. Finally, press [EDIT/SAVE] to store the setup.

2. CONNECTING THE PROGRAMMABLE EST VIA GPIB INTERFACE

The GPIB interface capabilities:

The GPIB interface of the programmable EST corresponds to the standard of IEEE488.1-1987, IEEE488.2-1992 and SCPI-1994. The GPIB interface functions are listed as follows:

SH1(Source Handshake) : The EST can transmit multilane messages across the GPIB.

AH1(Acceptor Handshake) : The EST can receive multilane messages across the GPIB.

T6(Talker) : Talker interface function includes basic talker, serial poll, and unaddress if MLA capabilities, without talk only mode function.

L4 (Listener) : The EST becomes a listener when the controller sends its listen address with the ATN (attention) line asserted. The power supply does not have listen only capability.

SR1 (Service Request) : The EST asserts the SRQ (Service request) line to notify the controller when it requires service.

RL1 (Remote/Local) : The EST responds to both the GTL(Go to Local) and LLO(Local Lock Out) interface messages.

PP0 (Parallel Poll) : The EST has no Parallel Poll interface function.

DC1 (Device Clear) : The EST has Device clear capability to return the device to power on status.

DT0 (Device Trigger) : The EST has no Device Trigger interface function.
C0 (Controller) : The EST can not control other devices.

Notes for GPIB installation

When the programmable EST is set up with a GPIB system, please check the following things:

- *Only a maximum of 15 devices can be connected to a single GPIB bus.*
- *Do not use more than 20m of cable to connect devices to a bus.*
- *Connect one device for every 2m of cable used.*
- *Each device on the bus needs a unique device address. No two devices can share the same device address.*
- *Turn on at least two-thirds of the devices on the GPIB system while using the system.*
- *Do not use loop or parallel structure for the topology of GPIB system.*

Computer's Connection

A personal computer with a GPIB card is the essential facilities in order to operate the programmable EST via GPIB interface.

The connections between power supply and computer are following:

I. Connect one end of a GPIB cable to the computer.

II. Connect the other end of the GPIB cable to the GPIB port on the programmable EST.

III. Turn on the programmable EST.

IV. Turn on the computer.

The GPIB connection testing

If you want to test whether the GPIB connection is working or not, you can send a GPIB command from computer. For instance, the query command

```
*idn?
```

should return the Manufacturer, model number, serial number and firmware version in the following format:

```
GW.Inc,GPI-745,0,FW1.00
```

If you do not receive a proper response from the EST, please check if the power is on, the GPIB address is correct, and all cable connections are active.

3. CONNECTING THE PROGRAMMABLE EST VIA RS232 INTERFACE

The RS232 interface capabilities:

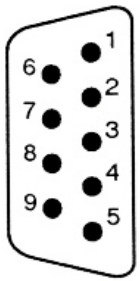
The RS232 interface provides a point-to-point connection between two items of equipment such as a computer and the EST. There are some parameters you need to set on the both sides. Once you have set these parameters, you can control the power supply through the RS232 interface.

- Baud rate: You can set rates of 1200, 2400, 4800 or 9600 baud.
- Parity bit: none.
- Data bit: 8 bits.
- Stop bit: 1 stop bit.
- Data flow control: none.

Notes for RS232 installation

The EST is a DTE device with a 9-pin D-type shell RS232 connector located on the rear panel. Figure 1 shows the equipment of 9-pin connector (male) with its pin number assignments. Figure 2 shows the wiring configuration for DB9 to DB9. When the programmable EST is set up with a RS232 interface, please check the following points:

- *Do not connect the output line of one DTE device to the output line of the other.*
- *Many devices require a constant high signal on one or more input pins.*
- *Ensure that the signal ground of the equipment is connected to the signal ground of the external device.*
- *Ensure that the chassis ground of the equipment is connected to the chassis ground of the external device.*
- *Do not use more than 15m of cable to connect devices to a PC.*
- *Ensure the same baud rate is used on the device as the one used on PC terminal.*
- *Ensure the connector for the both side of cable and the internal connected line are met the demand of the instrument.*



1. No connection
2. Receive Data(RxD) (input)
3. Transmit Data(TxD) (output)
4. No connection
5. Signal Ground(GND)
6. No connection
7. No connection
8. No connection
9. No connection

Figure 1 Pin assignments of the RS232 connector on the rear panel for DB-9-D

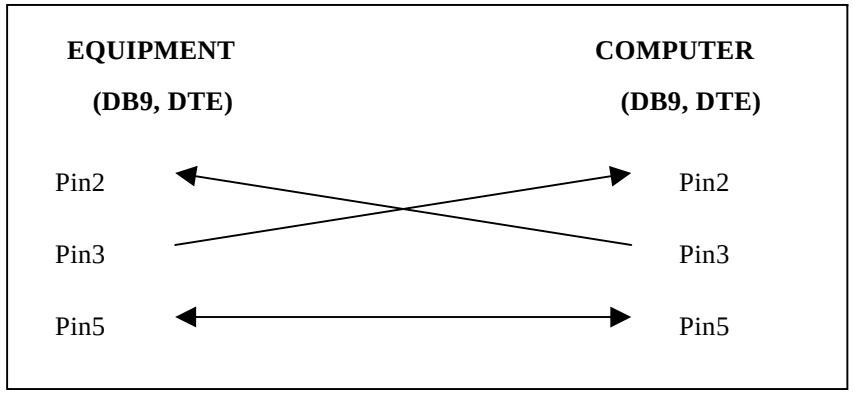


Figure 2 Wiring configuration for DB9 to DB9

Computer's Connection

A personal computer with a COM port is the essential facilities in order to operate the programmable EST via RS232 interface.

The connections between EST and computer are as follows:

I. Connect one end of a RS232 cable to the computer.

II. Connect the other end of the cable to the RS232 port on the programmable EST.

III. Turn on the programmable EST.

IV. Turn on the computer.

The RS232 connection testing

If you want to test whether the RS232 connection is working or not, you can send a command from computer. For instance, using a terminal program send the query command

```
*idn?
```

should return the Manufacturer, model number, serial number and firmware version in the following format:

```
GW.Inc,GPI-745,0,FW1.00
```

If you do not receive a proper response from the EST, please check if the power is on, the RS232 baud rate are the same on both sides, and all cable connections are active.

4. INPUT AND OUTPUT QUEUE

The design of 128 bytes input queue and 128 bytes output queue for storing the pending commands or return messages is to prevent the transmitted commands of remote control and return messages from missing. As the maximum stored capacity for Error/Event Queue is 20 groups of messages, it should be noted that input data exceeding the capacity by using these buffers will cause data missing.

5. COMMANDS AND SYNTAX

The GPIB commands of the programmable EST are compatible with IEEE-488.2 and SCPI standards

SCPI

SCPI (Standard Commands for Programmable Instruments) is a standard that created by an international consortium of the major test and measurement equipment manufacturers. The IEEE-488.2 syntax has been adopted by SCPI to provide common commands for the identical functions of different programmable instruments.

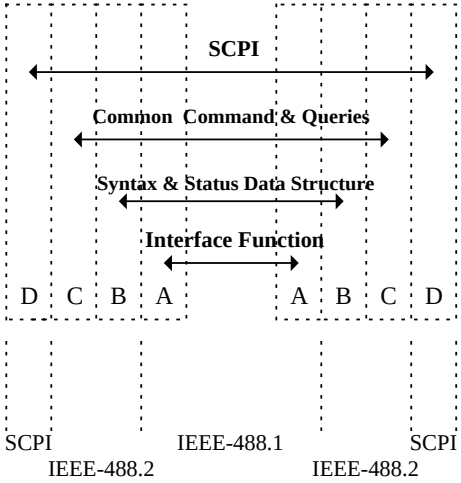


Figure 3 the relationship between IEEE-488.1, IEEE-488.2, and SCPI

As shown in the figure 3, the IEEE-488.1 standard locates at layer A, the layer A belongs to the protocol of interface function on the GPIB bus. The source handshake (SH), acceptor handshake (AH) and talker are included to this layer (10 interface functions totally).
At layer B, the syntax and data structure could be the essence of entire IEEE-488.2 standard. The syntax defines the function of message communication, which contain the <PROGRAM MESSAGE> (or simply “commands”) and <RESPONSE MESSAGE>. The two kinds of messages represent the syntax formation of device command and return value. The data structure is the constitution of status reporting, which IEEE-488.2 standard have been defined.

The common commands and queries are included to layer C. Commands and queries can be divided into two parts: mandatory and optional. Commands modify control settings or tell the instrument to perform a specific action. Queries cause the instrument to send data or status information back to the computer. A question mark at the end of a

command identifies it as a query.

Layer D is interrelated with device information. Different devices have different functions. SCPI command sets belong to this layer.

Command Syntax

If you want to transfer any instructions to an instrument, and comply with SCPI, there are three basic elements must be included.

- Command header
- Parameter (if required)
- Message terminator or separator

Command Header

The command header has a hierarchical structure that can be represented by a command tree (Figure 4).
The top level of the tree is the root level. A root node is located at the root level. A root node and one or more lower-level nodes form a header path to the last node called the leaf node.

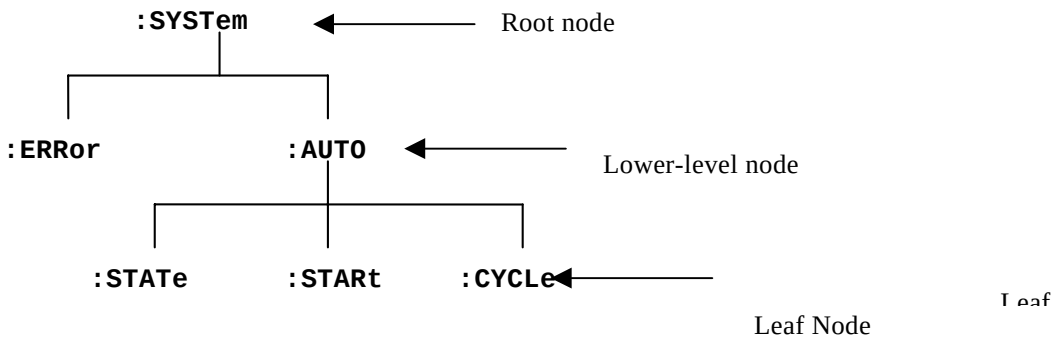


Figure 4: Tree hierarchy

The command header is configured by header path and leaf node. Figure 5 shows the command header for the leaf node indicated in Figure 4.

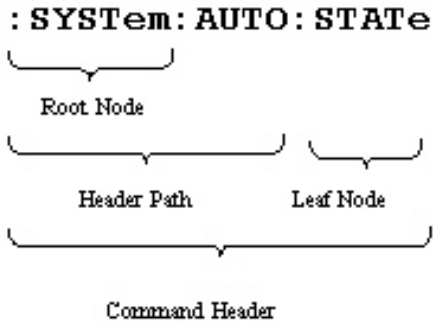


Figure 5 Command Header

Parameter

If the commands have parameters, the values have to be included. In this manual, when we expressed the syntax of the command, the < > symbols are used for enclosing the parameter type. For instance, the syntax of the command in Figure 6 includes the Boolean parameter type.

NOTE: Do not include the <, >, or | symbols when entering the actual value for a parameter.

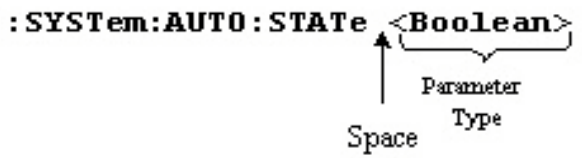


Figure 6 Command Header with Parameter

Table 1 defines the Boolean and other parameter types for the programmable EST.

Parameter Type	Description	Example
Boolean	Boolean numbers or values	0, 1
NR1	Integers	0, 1, 18
NR2	Decimal numbers	1.5, 3.141, 8.4
NR3	Floating point numbers	4.5E-1, 8.25E+1
String	Alphanumeric characters	“No error”

Table 1: Parameter Types for Syntax Descriptions

Message Terminator and Message Separator

I. GPIB message terminators

In accordance with IEEE 488.2 standard, any of the following message terminators are acceptable:

- LF^END Line feed code (hexadecimal 0A) with END message
- LF Line feed code
- <dab>^END Last data byte with END message

These terminators are compatible with most application programs. A semicolon separates one command from another when the commands appear on the same line.

II. RS232 message terminators

As there is no signal of end message on RS232 bus, therefore, use LF as message terminator. When a series of commands are sent to the instrument, it must add a LF to be a judgment for message terminator. As for query command, the return message of the instrument is also added a LF for PC to judge message terminator.

Entering Commands

The standards that govern the command set for the programmable EST allow for a certain amount of flexibility when you enter commands. For instance, you can abbreviate many commands or combine commands into one message that you send to the programmable EST. This flexibility, called friendly listening, saves programming time and makes the command set easier to remember and use.

Command Characters

The programmable ESTs are not sensitive to the case of command characters. You can enter commands in either uppercase or lowercase.

You can execute any command with white space characters. You must, however, use at least one space between the parameter and the command header

Abbreviating Commands

Most commands have a long form and a short form. The listing for each command in this section shows the abbreviations in uppercase. For instance, you can enter the query : ACWStanding: VOLTage 1.000 simply as : ACWStanding: VOLTage 1.000

Because the programmable EST hypothesis that a command starts from the root, you have the option of beginning the initial command header with a colon (:).

Combining Commands

You can use a semicolon (;) to combine commands. But continuously query command will cause message missing. For example
:GRO:STEP: ACWS:VOLT:;CMAX?

If the command that follows the semicolon has a different header path from the root level, you must use a colon to force a return to the root level.

:ACC:CMIN 1.00;:ACC:CMAX 2.00

If the command that follows the semicolon has the same header path, you may omit the colon and the path and state only the new leaf node. For example:

:ACC:CMIN 1.00;:ACC:CMAX 2.00

is equal to

:ACC:CMIN 1.00;:CMAX 2.00

You can combine commands and queries into the same message. Note, for example, the following combination:

:ACC:CMIN 1.00;CMIN?

Synopsis of Commands

The tables in this section summarize the command of the programmable power supply. These tables divide the commands into three functional classifications:

- General Setting Commands
- Status Commands
- Miscellaneous Commands

The tables also provide a brief explanation of each command.

- **General Setting Commands**

Table 2 lists the general setting commands that control and query the settings of the power supply.

Table 2: General Setting Commands

Commands	Explanation
:GROup <NR1>	Set the value of group
:GROup?	Query the value of group
:STEP <NR1>	Set the value of step
:STEP?	Query the value of step
:FUNCtion:TEST:MODe <NR1>	Set the value of test mode
:FUNCtion:TEST:MODe?	Query the value of test mode
:FUNCtion:TEST:STATe <NR1>	Set the value of test state
:FUNCtion:TEST:STATe?	Query the value of test state
:MEASure<x>?	Query the value of measurement
:UTILity: FREQuency <50 or 60>	Set the value of AC frequency
:UTILity: FREQuency?	Query the value of AC frequency
:UTILity:ARC:CURRent <NR2>	Set the value of ARC current
:UTILity:ARC:CURRent?	Query the value of ARC current
:UTILity:ARC:MODe <NR1>	Set the ARC test mode
:UTILity:ARC:MODe?	Query the ARC test mode
:UTILity:TEST:MODe <NR1>	Set the test mode from step 1 or form present step
:UTILity:TEST:MODe?	Query the test mode
:UTILity:IRTest:MODe <NR1>	Set the value of IR test mode
:UTILity:IRTest:MODe?	Query the value of IR test mode

:UTILity:GBZero:STATe?	Query the GB zero check
:UTILity:GBZero:TEST	Start the GB zero check
:ACContinue:VOLTage <NR2>	Set the CAC test voltage
:ACContinue:VOLTage?	Query the CAC test voltage
:ACContinue:CMINimum <NR2>	Set the CAC test minimum current
:ACContinue:CMINimum?	Query the CAC test minimum current
:ACContinue:CMAXimum <NR2>	Set the CAC test maximum current
:ACContinue:CMAXimum?	Query the CAC test maximum current
:ACContinue:RTIMe <NR2>	Set the CAC ramp time
:ACContinue:RTIMe?	Query the CAC ramp time
:DCContinue:VOLTage <NR2>	Set the CDC test voltage
:DCContinue:VOLTage?	Query the CDC test voltage
:DCContinue:CMINimum <NR2>	Set the CDC test minimum current
:DCContinue:CMINimum?	Query the CDC test minimum current
:DCContinue:CMAXimum <NR2>	Set the CDC test maximum current
:DCContinue:CMAXimum?	Query the CDC test maximum current
:DCContinue:RTIMe <NR2>	Set the CDC ramp time
:DCContinue:RTIMe?	Query the CDC ramp time
:ACWStanding:VOLTage <NR2>	Set the ACW test voltage
:ACWStanding:VOLTage?	Query the ACW test voltage
:ACWStanding:CMINimum <NR2>	Set the ACW test minimum current
:ACWStanding:CMINimum?	Query the ACW test minimum current
:ACWStanding:CMAXimum <NR2>	Set the ACW test maximum current
:ACWStanding:CMAXimum?	Query the ACW test maximum current
:ACWStanding:RTIMe <NR2>	Set the ACW ramp time

:ACWStanding:RTIMe?	Query the ACW ramp time
:ACWStanding:TTIMe <NR2>	Set the ACW test time
:ACWStanding:TTIMe?	Query the ACW test time
:DCWStanding:VOLTage <NR2>	Set the DCW test voltage
:DCWStanding:VOLTage?	Query the DCW test voltage
:DCWStanding:CMINimum <NR2>	Set the DCW test minimum current
:DCWStanding:CMINimum?	Query the DCW test minimum current
:DCWStanding:CMAXimum <NR2>	Set the DCW test maximum current
:DCWStanding:CMAXimum?	Query the DCW test maximum current
:DCWStanding:RTIMe <NR2>	Set the DCW ramp time
:DCWStanding:RTIMe?	Query the DCW ramp time
:DCWStanding:TTIMe <NR2>	Set the ACW test time
:DCWStanding:TTIMe?	Query the ACW test time
:IRESistance:VOLTage <NR2>	Set the IR test voltage
:IRESistance:VOLTage?	Query the IR test voltage
:IRESistance:RMINimum <NR2>	Set the IR test minimum resistance
:IRESistance:RMINimum?	Query the IR test resistance
:IRESistance:RMAXimum <NR2>	Set the IR test minimum resistance
:IRESistance:RMAXimum?	Query the IR test resistance
:IRESistance:RMAXimum <NR2>	Set the IR test maximum resistance
:IRESistance:RMAXimum?	Query the IR test resistance
:IRESistance:TTIMe <NR2>	Set the IR test time
:IRESistance:TTIMe?	Query the IR test time

:GBONd:CURRent <NR2>	Set the GB test current
:GBONd:CURRent?	Query the GB test current
:GBONd:RMINimum <NR2>	Set the GB test minimum resistance
:GBONd:RMINimum?	Query the GB test resistance
:GBONd:RMAXimum <NR2>	Set the GB test minimum resistance
:GBONd:RMAXimum?	Query the GB test resistance
:GBONd:TTIMe <NR2>	Set the GB test time
:GBONd:TTIMe?	Query the GB test time
:CONTinuity:RMINimum <NR2>	Set the CONT test minimum resistance
:CONTinuity:RMINimum?	Query the CONT test resistance
:CONTinuity:RMAXimum <NR2>	Set the CONT test minimum resistance
:CONTinuity:RMAXimum?	Query the CONT test resistance
:CONTinuity:TTIMe <NR2>	Set the CONT test time
:CONTinuity:TTIMe?	Query the CONT test time

- **Status Commands**

Table 3 lists the status commands that set and query the various registers and queues that make up the status and event structure of the programmable power supply.

Table 3: Status Commands

*CLS	Clears the status data structures.
*ESE <NR1>	Sets the Event Status Enable Register (ESER).
*ESE?	Returns contents of Event Status Enable Register (ESER).

*ESR?	Returns and clear the contents of Standard Event Status Register (SESR).
*SRE <NR1>	Sets contents of Service Request Enable Register(SRER).
*SRE?	Returns contents of Service Request Enable Register (SRER).
*STB?	Reads Status Byte Register (SBR).
:STATus:OPERation:CONDition ?	Returns the contents of the OPERation condition register. Returns NR1.
:STATus:OPERation:ENABle <NR1>	Sets the contents of the enable mask for the OPERation event register.
:STATus:OPERation:ENABle ?	Returns the contents of the enable mask for the OPERation event register. Returns NR1.
:STATus:OPERation:EVENt ?	Query the contents of the OPERation Event register.
:STATus:PRESet	Presets the OPERation and QUEStionable status registers.
:STATus:QUEStionable: CONDition ?	Returns the contents of the OPERation condition register. Returns NR1.
:STATus:QUEStionable:ENABle <NR1>	Sets the contents of the enable mask for the QUEStionable enable register.
:STATus:QUEStionable:ENABle?	Query the contents of the Questionable Enable register.

:STATus:QUEStionable:EVENt ?	Query the contents of the QUEStionable Event register.

- **Miscellaneous Commands**

Table 4 lists the miscellaneous commands that control general housekeeping functions of the programmable power supply.

Table 4: Miscellaneous Commands

*IDN?	Returns instrument identification.
*OPC	Reports when operation is complete by setting the Operation Complete bit in SESR.
*OPC?	Reports when operation is complete. Same as *OPC except returns a 1 to the output queue and dose not set the SESR bit.
*RST	Resets the protection levels and states, resets the current and voltage levels to zero, sets the output off, sets both Group section and Step position to 0.
*TST?	Initiates internal self-test and reports results.
*WAI	Wait to continue. This command forces sequential operation of commands. This command is required by IEEE-488.1-1987. The power supply, however, forces sequential operation of commands by design.
:SYSTem: ERRor?	Read the next item from the error/event queue.
:SYSTem: VERSion?	Return the SCPI version level.

6. DETAILS OF COMMAND REFERENCE

Each command in this chapter will give a detailed description. The examples of each command will be provided and what query form might return.

*CLS (no query form)

Function:
Clear all event status data register. This includes the Output Queue, Operation Event Status Register, Questionable Event Status Register, and Standard Event Status Register.

Syntax:

`*CLS`

Examples:

`*CLS` clears all event registers.

*ESE

Function:
Set or return the bits in the Event Status Enable Register (ESER). The ESER enables the Standard Event Status Register (SESR) to be summarized on bit 5 (ESB) of the Status Byte Register (SBR).

Syntax:

`*ESE <NR1>`

`*ESE?`

`<NR1>` is in the range from 0 through 255.

Returns:

`<NR1>` is a number from 0 to 255 that indicates the decimal value of the binary bits of the ESER.

Examples:

`*ESE 65` sets the ESER to binary `0100 0001`.

If the ESER contains the binary value `1000 0010`, `the *ESE?` will return the value of 130.

*ESR? (query only)

Function:
Return and clear the contents of the Standard Event Status Register (SESR).

Syntax:

`*ESR?`

Returns:

`<NR1>` is a number from 0 to 255 that indicates the decimal value of the binary bits of the ESER.

Examples:

If the ESER contains the binary value `1100 0110`, `the *ESR?` will return the value of 198.

*IDN? (query only)

Function:
Return the unique identification code of the EST.

Syntax:

`*IDN?`

Returns:

`<string>` includes Manufacturer, model number, serial number and firmware version.

Examples:

*IDN? Returns GW, GPI-745, 0, FW:1.00

*OPC

Function:
The command form (`*OPC`) sets the operation complete bit (bit 0) in the Standard Event Status Register (SESR) when all pending operations are finished.

The query form (`*OPC?`) tells the programmable EST to place an ASCII 1 in the Output Queue when the power supply completes all pending operations.

Syntax:

`*OPC`
`*OPC?`

Returns:

`1`

*RST (no query form)

Function:
Set all control settings of EST to their default values but does not purge stored setting. The equivalent panel control will be set as below:

Front Panel Control	Default Setting
TEST	OFF
Group and Step	0:0

Syntax:

*RST

*SRE

Function:
Set the contents of the Service Request Enable Register (SRER). The query form returns the contents of the SRER. Bit 6 of the SRER is always zero. The bits on the SRER correspond to the bits on the SBR.

Syntax:

*SRE <NR1>
*SRE?

Returns:

<NR1> is in the range from 0 through 255.

Examples

*SRE 7 sets bits of the SRER to 0000 0111.
If the *SRE? returns 3, the SRER is set to 0000 0011.

*STB? (query only)

Function:

The query of the Status Byte register (SBR) with *STB? will return a decimal number representing the bits that are set (true) in the status register.

Syntax:

*STB?

Returns:

<NR1> is in the range from 0 through 255.

Examples:

*STB? returns 81, if SBR contains the binary value 0101 0001.

*TST? (query only)

Function:
Self-test and test the RAM, ROM.

Syntax:

*TST?

Returns:

0|-300

Examples:

*TST? returns 0, if the test is successful.
*TST? returns –300, if the test is unsuccessful.

*WAI (no query form)

Function:

WAI prevents the programming instrument from executing further commands or queries until all pending operations are finished.

Syntax:

*WAI

:GROup

Function:

Set or query the value of group.

Syntax:

:GROup <NR1>

:GROup?

<NR1>is in the range from 0 through 10.

* 10: COM mode

Return:

<NR1>

Examples:

:GROup 3 set the group 3.

:GROup? return 3 if the group setting is 3.

:STEP

Function:

Set or query the value of step

Syntax:

: STEP <NR1>

:SETP?

<NR1> is in the range from 1 through 16.

*0: COM mode

Return:

<NR1>

Examples:

:SETP3 set the step 3

:SETP? return 3 if the STEP setting is 3.

:FUNCtion:TEST:MODe

Function:

Set or query the test function (0:EMPTY 1: ACW 2: DCW 3: IR 4: Cnt (GB for 745A) 5: CAC 6: CDC 7: HOLD). Only Group 0 has CAC and CDC function, but Group 0 does not have HOLD function.

Syntax:

:FUNCtion:TEST:MODe <NR1>

:FUNCtion:TEST:MODe?

<NR1> is in the range from 0 through 7.

Return:

<NR1>

Examples:

:FUNCtion:TEST:MODe 3 set the test function to IR.

:FUNCtion:TEST:MODe? return 3 if the test function setting is IR.

:FUNCtion:TEST:STATe

Function:

Set or query the test state (0. OFF 1. ON 2. CONTINUE). During testing, when proceed HOLD function, the instrument will stop testing until a command of "FUNC:TEST:STAT 2" has been received.

Syntax:

:FUNCtion:TEST:STATe <NR1>

:FUNCtion:TEST:STATe?

<NR1> is in the range from 0 through 1.

Return:

<NR1>

Examples:

:FUNCtion:TEST:STATe 1 start test.

:FUNCtion:TEST:STATe ? return 1 if the test state is 1.

:MEASure

Function:

Return measurement value

Syntax:

:MEASure<X>?

<X> is in the range from 1 through 16 for STEP.

: MEASure? Return measurement value of COM mode.

Return:

Status, **Mode**, **V/I**, **Meas**, **Time**

(2 bytes) (1 byte) (7 bytes) (8 bytes) (7 bytes)

Status: 00:Emtpy
10:Pass
20:Fail
21:Interlock
22:Low Voltage
23:Hi voltage
24:ARC
25:Measurement over limit maximum
26:Measurement below limit minimum
30:Stop
32:Hight Voltage
33:Heatsink is too hot

Mode: 0: Empty
1:ACW
2:DCW
3:IR
4:Cnt (GB for 745A)
5:CAC
6:CDC
7:HOLD

V/I:

Test Function	Unit (2 bytes)
ACW	KV
DCW	KV
IR	V
Cnt (GB)	A
CAC	KV
CDC	KV
HOLD	PAUSE

Meas:

Test Function	Unit (4 bytes)
ACW	mA
DCW	mA
IR	MOhm
Cnt (GB)	mOhm
CAC	mA
CDC	mA
HOLD	

Time:

Time (1 byte)	Value (5 bytes)	Unit (1 bytes)
Ramp	005.0	sec
Test	005.0	sec

Examples:

User the command **:MEAS3?** after measurement

Display for PC 10,1,0.100kV,01.00mA, T005.0S

10=Pass

1=ACW

0.100kV=Mesuremett voltage 100V

01.00mA=Mesurement current 1mA

T005.0S=Test time 5 sec

:UTILity:AC:FREQuency

Function:

Set or query the ACW CAC frequency (0:50Hz, 1:60Hz)

Syntax:

:UTILity: FREQuency <Boolean>

:UTILity: FREQuency?

50 (50Hz) or 60 (60Hz)

Return:

50 | 60

Examples:

:UTILity: FREQuency 50 set the ACW CAC frequency to 50Hz

:UTILity: FREQuency? return 50 if the ACW CAC frequency setting is 50Hz.

:UTILity:ARC:CURRent

Function:

Set or query the ARC current

Syntax:

:UTILity:ARC:CURRent <NR2>

:UTILity:ARC:CURRent?

<NR2>Please refer to the specification

Return:

<NR2>

Examples:

:UTILity:ARC:CURRent 2.00 set the ARC current to 2.00mA.

:UTILity:ARC:CURRent? return 02.00 if the ARC current setting is 2.00mA.

:UTILity:ARC:MODe

Function:

Set or query the ARC test state

(0:disable, 1:Enable & Stop, 2:Enable & Continue)

Syntax:

:FUNCtion:TEST:STATe <NR1>

:FUNCtion:TEST:STATe?

<NR1> is in the range from 0 through 2.

Return:

<NR1>

Examples:

:UTILity:ARC:MODe 0 set the AEC test state to disable.

:UTILity:ARC:MODe? return 0 if the ARC mode setting is disable.

:UTILity:TEST:MODe

Function:

Set or query the test mode (0: From Step 1, 1: From the present step)

Syntax:

:UTILity:TEST:MODe <Boolean>

:UTILity:TEST:MODe?

<Boolean>can be 0 or 1

Return:

0 | 1

Examples:

:UTILity:TEST:MODe 0 set the test mode from step 1.

:UTILity:TEST:MODe? return 0 if the mode setting is From Step 1.

:UTILity:IRTest:MODe

Function:

Set or query the IR test mode (0: Stop on Fail, 1: Stop on Pass, 2: TIMER)

Syntax:

:UTILity:IRTest:MODe <NR1>

:UTILity:IRTest:MODe?

<NR1> is in the range from 0 through 2.

Return:

<NR1>

Examples:

:UTILity:IRTest:MODe 0 set the IR test mode stop on fail.

:UTILity:IRTest:MODe? return 0 if the IR test mode setting is STOP on Fail.

:UTILity:GBZero:TEST

Function:

Start the GB zero check.

Syntax:

:UTILity:GBZero:TEST

Return:

Examples:

:UTILity:GBZero:TEST Start the GB zero check.

:UTILity:GBZero:STATe

Function:

The query from return 1 if the GB zero check is OK and 0 if the GB zero check is fail.

Syntax:

:UTILity:GBZero:STATe?

Return:

0|1

Examples:

:UTILity:GBZero:STATe? Return 1 if the GB zero check is OK.

:ACContinue:VOLTage

Function:

Set or query the CAC test voltage.

Syntax:

:ACContinue:VOLTage <NR2>

:ACContinue:VOLTage?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACContinue:VOLTage 1.000 set the CAC test voltage to 1.000kV.

:ACContinue:VOLTage? return 1.000 if the CAC test voltage setting is 1.000kV.

:ACContinue:CMINimum

Function:

Set or query the CAC test minimum current.

Syntax:

:ACContinue:CMINimum <NR2>

:ACContinue:CMINimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACContinue:CMINimum 1.00 set the CAC test minimum current to 1.00mA.

:ACContinue:CMINimum? return 01.00 if the CAC test minimum current setting is 1.00mA.

:ACContinue:CMAXimum

Function:

Set or query the CAC test maximum current.

Syntax:

:ACContinue:CMAXimum <NR2>

:ACContinue:CMAXimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACContinue:CMAXimum 1.00 set the CAC test maximum current to 1.00mA.

:ACContinue:CMAXimum? return 01.00 if the CAC maximum current setting is 1.00mA

:ACContinue:RTIMe

Function:

Set or query the CAC ramp time.

Syntax:

:ACContinue:RTIMe <NR2>

:ACContinue:RTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACContinue:RTIMe 5.0 set the CAC ramp time to 5 sec.

:ACContinue:RTIMe? return 005.0 if the CAC ramp time setting is 5 sec.

:DCContinue:VOLTage

Function:

Set or query the CDC test voltage.

Syntax:

:DCContinue:VOLTage <NR2>

:DCContinue:VOLTage?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCContinue:VOLTage 1.000 set the CDC test voltage to 1.000kV.

:DCContinue:VOLTage? Return 1.000 if the CDC test voltage setting is 1.000kV.

:DCContinue:CMINimum

Function:

Set or query the CDC test minimum current.

Syntax:

:DCContinue:CMINimum <NR2>

:DCContinue:CMINimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCContinue:CMINimum 1.00 set the CDC test minimum current to 1.00mA.

:DCContinue:CMINimum? return 01.00 if the CDC test minimum current setting is 1.00mA.

:DCContinue:CMAXimum

Function:

Set or query the CDC test maximum current.

Syntax:

:DCContinue:CMAXimum <NR2>

:DCContinue:CMAXimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCContinue:CMAXimum 1.00 set the CDC test maximum current to 1.00mA.

:DCContinue:CMAXimum? return 01.00 if the CDC test maximum current setting is 1.00mA.

:DCContinue:RTIMe

Function:

Set or query the CDC ramp time.

Syntax:

:DCContinue:RTIMe <NR2>

:DCContinue: RTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCContinue:RTIMe 5.0 set the CDC ramp time to 5 sec.

:DCContinue:RTIMe? return 005.0 if the CDC ramp time setting is 5 sec.

:ACWStanding:VOLTage

Function:

Set or query the ACW test voltage.

Syntax:

:ACWStanding:VOLTage <NR2>

:ACWStanding:VOLTage?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

: ACWStanding:VOLTage 1.000 set the ACW test voltage to 1.000kV.

: ACWStanding:VOLTage return 1.000 if the CAC test voltage setting is 1.000kV.

:ACWStanding:CMINimum

Function:

Set or query the ACW test minimum current.

Syntax:

:ACWStanding:CMINimum <NR2>

:ACWStanding:CMINimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACWStanding:CMINimum 1.00 set the ACW test minimum current to 1.00mA.

:ACWStanding:CMINimum? return 01.00 if the ACW test minimum current setting is 1.00mA.

:ACWstanding:CMAXimum

Function:

Set or query the ACW test maximum current.

Syntax:

:ACWStanding:CMAXimum <NR2>

:ACWStanding:CMAXimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACWStanding:CMAXimum 1.00 set the ACW test maximum current to 1.00mA.

:ACWStanding:CMAXimum? return 01.00 if the ACW test maximum current setting is 1.00mA.

:ACWStanding:RTIMe

Function:

Set or query the ACW ramp time.

Syntax:

:ACWStamding:RTIMe <NR2>

:ACWStamding:RTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACWStanding:RTIMe 5.0 set the ACW ramp time to 5 sec.

:ACWStanding:RTIMe? return 005.0 if the ACW ramp time setting is 5 sec.

:ACWStanding:TTIMe

Function:

Set or query the ACW test time.

Syntax:

:ACWStamding:TTIMe <NR2>

:ACWStamding:TTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:ACWStanding:TTIMe 5.0 set the ACW test time to 5 sec.

:ACWStanding:TTIMe? return 005.0 if the ACW test time setting is 5 sec.

:DCWStanding:VOLTage

Function:

Set or query the DCW test voltage.

Syntax:

:DCWStanding:VOLTage <NR2>

:DCWStanding:VOLTage?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

: DCWStanding:VOLTage 1.000 set the DCW test voltage to 1.000kV.

: DCWStanding:VOLTage? return 1.000 if the DCW test voltage setting is 1.000kV.

:DCWStanding:CMINimum

Function:

Set or query the DCW test minimum current.

Syntax:

:DCWStanding:CMINimum <NR2>

:DCWStanding:CMINimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCWStanding:CMINimum 1.00 set the DCW test minimum current to 1.00mA.

:DCWStanding:CMINimum? return 01.00 if the DCW test minimum current setting is 1.00mA.

:DCWstanding:CMAXimum

Function:

Set or query the DCW test maximum current

Syntax:

:DCWStanding:CMAXimum <NR2>

:DCWStanding:CMAXimum?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCWStanding:CMAXimum 1.00 set the DCW test maximum current to 1.00mA.

:DCWStanding:CMAXimum? return 01.00 if the DCW test maximum current setting is 1.00mA.

:DCWStanding:RTIMe

Function:

Set or query the DCW ramp time.

Syntax:

:DCWStamding:RTIMe <NR2>

:DCWStamding:RTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCWStanding:RTIMe 5.0 set the DCW ramp time to 5 sec.

:DCWStanding:RTIMe? return 005.0 if the DCW ramp time setting is 5 sec.

:DCWStanding:TTIMe

Function:

Set or query the DCW test time.

Syntax:

:DCWStamding:TTIMe <NR2>

:DCWStamding:TTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:DCWStanding:TTIMe 5.0 set the DCW test time to 5 sec.

:DCWStanding:TTIMe? return 005.0 if the DCW test time setting is 5 sec.

:IRESistance:VOLTage

Function:

Set or query the IR test voltage.

Syntax:

:IRESistancce:VOLTage <NR1>

:IRESistancce:VOLTage?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:IRESistancce:VOLTage 100 set the IR test voltage to 100V.

:IRESistancce:VOLTage? return 0100 if the IR test voltage setting is 100V.

:IRESistance:RMINimum

Function:

Set or query the IR test minimum resistance.

Syntax:

:IRESistance:RMINimum <NR1>

:IRESistance:RMINimum?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:IResistance:RMINimum 100 set the IR test minimum resistance to 100MΩ.

:IResistance:RMINimum? return 0100 the IR test minimum resistance setting is 100MΩ.

:IRESistance:RMAXimum

Function:

Set or query the IR test maximum resistance.

Syntax:

:IRESistance:RMAXimum <NR1>

:IRESistance:RMAXimum?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:IRESistance:RMAXimum 100 set the IR test maximum resistance to 100MΩ.

:IRESistance:RMAXimum? return 0100 the IR test maximum resistance setting is 100MΩ.

:IRESistance:TTIMe

Function:

Set or query the IR test time.

Syntax:

:IRESistance:TTIMe <NR2>

:IRESistance:TTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:IRESistance:TTIMe 5.0 set the IR test time to 5 sec.

:IRESistance:TTIMe? return 005.0 if the IR test time setting is 5 sec.

:GBONd:CURRent

Function:

Set or query the GB test current.

Syntax:

:GBONd:CURRent <NR1>

:GBONd:CURRent?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:GBONd:CURRent 25.00 set the GB test current to 25A.

:GBONd:CURRent? return 25.00 if the GB test current setting is 25A.

:GBONd:RMINimum

Function:

Set or query the GB test minimum resistance.

Syntax:

:GBONd:RMINimum <NR1>

:GBONd:RMINimum?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:GBONd:RMINimum 100.0 set the GB test minimum resistance to 100mΩ.

:GBONd:RMINimum? return 100.0 if the GB test minimum resistance setting is 100mΩ.

:GBONd:RMAXimum

Function:

Set or query the GB test maximum resistance.

Syntax:

:GBONd:RMINimum <NR1>

:GBONd:RMINimum?

<NR1> Please refer to the specification

Return:

<NR1>

Examples:

:GBONd:RMINimum 100.0 set the GB test maximum resistance to 100.0mΩ.

:GBONd:RMINimum? return 100.0 if the GB test maximum resistance setting is 100mΩ.

:GBONd:TTIMe

Function:

Set or query the GB test time.

Syntax:

:GBONd:TTIMe <NR2>

:GBONd:TTIMe?

<NR2> Please refer to the specification

Return:

<NR2>

Examples:

:GBONd:TTIMe 5.0 set the IR test time to 5 sec.

:GBONd:TTIMe? return 005.0 if the GB test time setting is 5 sec.

:SYSTem:ERRor?(Querly only)

Function:

Query next error message from the Error/Event queue. The result of the query is the error number followed by the error text.

Syntax:

:SYSTem:ERRor?

Return:

<string>

Examples:

System:ERRor? Return 0, “No error”

:SYSTem:VERSion?(Querly only)

Function:

Return the SCPI version of the device.

Syntax:

:SYSTem:VERSion?

Return:

1994.0

7. STATUS AND ERROR REPORTING

A set of status registers allows the user to quickly determine the EST's internal processing status. The status register, as well as the status and event reporting system, adhere to SCPI recommendations.

Structure of System

The sketch of the status and event reporting system is showed as figure 7. Each component of the sketch represents a set of registers and queues that can read, report, or enable the occurrence of certain events within the system.

If a specific event in the EST sets a bit in a *status register*, reading which can tell you what types of events have occurred.

Each bit in the status register corresponds to a bit in an *enable register;* the enable bit must be high for the event to be reported to the Status Byte Register.

A Service Request (SRQ) is the last event to occur. The SRQ requests an interrupt on the GPIB to report events to the system controller.

Status Registers

There are two kinds of status registers are included to the programmable power supplies.

- OPERation Status Registers (CONDition, EVENt, and ENABle)
- QUEStionable Status Registers (CONDition, EVENt, and ENABle)

The lower level nodes: QUEStionable and OPERation each have three 16 bits registers: CONDition, EVENt, and ENABle. Figure 8 shows the sequential relationship between these three types of registers and the commands that relate to each register.

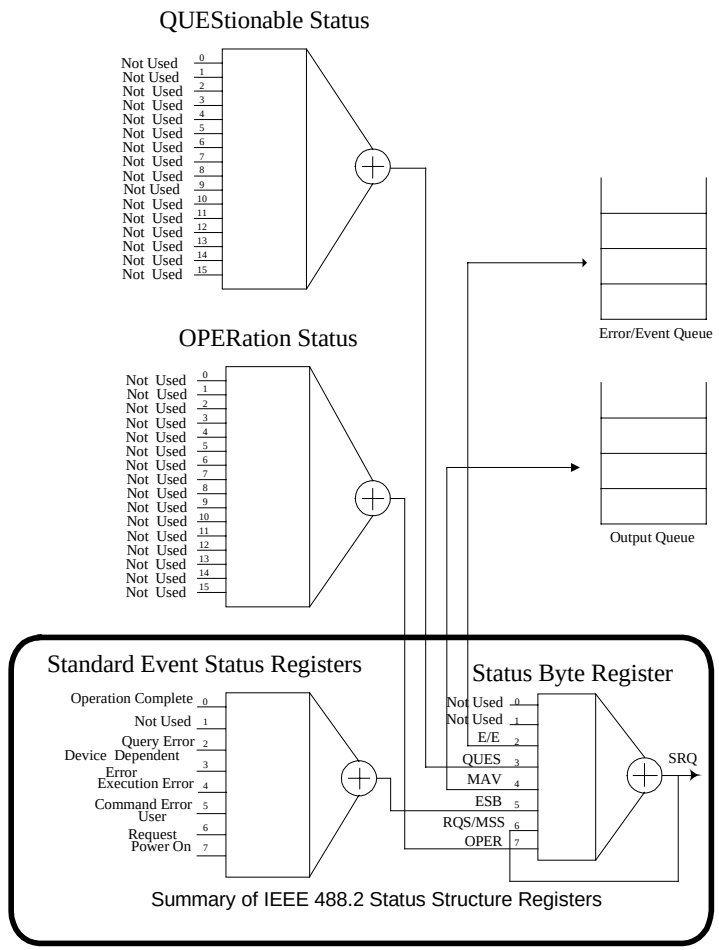


Figure 7. A graphic representation of the status registers and their connections.

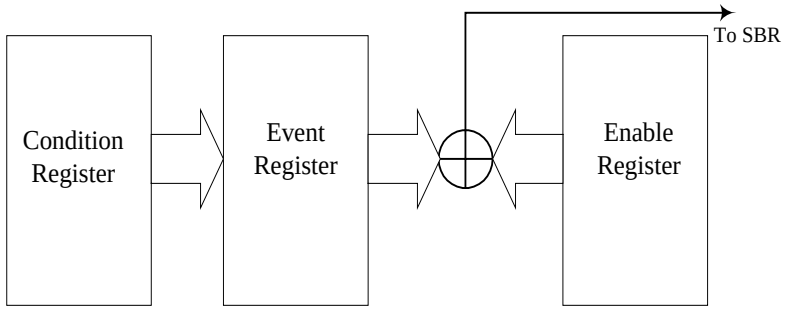


Figure 8: Status registers and related commands

The CONDition register is a read-only register which monitors the present state of the instrument. The CONDition register updates in real time and the inputs are not latched or buffered. When a condition monitored by the CONDition register becomes true, the bit for that condition also becomes true (1). When the condition is false, the bit is 0. The read-only EVENt register latches any false-to-true change in condition. Once the bit in the EVENt register is set, it is no longer affected by changes in the corresponding bit of the CONDition register. The bit remains set until the controller reads it. The command `*CLS` (Clear Status) clears the EVENt register.

QUEStionable Status Registers.

Table 4 shows the bit designations of the 16 bit QUEStionable Status Register.

Table 4: QUEStionable Status Register

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	*NU	NU	NU	NU	NU	NU	NU
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NU	NU	NU	NU	NU	NU	NU	NU

The command `STATus:QUEStionable:CONDtion?` Reads the QUEStionable CONDition register but dose not clear it.

The command `STATus:QUEStionable:EVENt?` Reads the QUEStionable EVENt Status register and clears it.

OPERation Status Registers

Table 5 shows the bit designations of the 16 bit OPERation Status Register.

Table 5: OPERation Status Register

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	NU	NU	NU	NU	NU	NU	NU
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NU	NU	NU	NU	NU	NU	NU	NU

Status Registers

* NU: not used

There are two status registers are included to the EST defined by IEEE-488.1 and IEEE-488.2 standards.

- Status Byte Register (SBR)
- Standard Event Status Register (SESR)

Status Byte Register (SBR): The SBR (Table 6) summarizes the status of all other registers and queues.

Table 6: Status Byte Register (SBR)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPER	RQS/MSS	ESB	MAV	QUES	E/E	NU	NU

The bit 0 and 1 are not used, so these bits are always zero. The bit 2 (Error and Event) indicates an error code is waiting to be read in the Error Event Queue. The bit 3 (QUES, QUEStionable) is the summary bit for the QESR (QUEStionable Event Status Register). When the bit is high it indicates that status is enabled and present in the QUES. The bit 4 (MAV, Message Available) indicates that output is available in the output queue. The bit 5 (ESB, Event Status Bit) is the summary bit for the Standard Event Status Register (SESR). When the bit is high it indicates that status is enabled and present in the SESR. The bit 6 (RQS, Request Service) is obtained from a serial poll and shows that the EST requests service from the GPIB controller. The bit 7 (OPER, OPERation) is the summary bit for the OESR (OPERation EVENt STATus Register).

Use the serial poll or the `*STB?` Query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the Standard Event Status Register (SESR), the Standard Event Status Register (SESR), and the Output Queue.

Standard Event Status Register (SESR): Table 7 shows the SESR

Table 7: Standard Event Status Register (SESR)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PON	URQ	CME	EXE	DDE	QYE	NU	OPC

The bit 0 (OPC, Operation Complete) shows that the operation is completed. This bit is active when all pending operations are completed following an `*OPC` command. The bit 1 is always zero. The bit 2 (QYE, Query Error) indicates a command or query protocol error. The bit 3 (DDE, Device Error) shows that a device error occurred. The bit 4 (EXE, Execution Error) shows that an error occurred while the EST was executing a command or query. The bit 5 (CME, Command Error) shows that an error occurred while the EST was parsing a command or query. The bit 6 (USR, User Request) indicates the LOCAL button was pushed. The bit 7 (PON, Power On) shows that the EST was powered on.

Use the *ESR? Query to read the SESR. Read the SESR and clear the bits of the registers so that the register can accumulate information about new events.

Enable Registers

The enable registers determine whether certain events are reported to the Status Byte Register and SRQ. The EST has the following enable registers.

- Event Status Enable Register (ESER)
- OPERation Enable Register
- QUEStionable Enable Register
- Service Request Enable Register (SRER)

When one of the bits of the enable registers is high and the corresponding bit in the status register is high, the enable registers will perform a logical OR function, the output that controls the set bit of the Status Byte Register is high.

Various commands set the bits in the enable registers. The following sections describe the enable registers and the commands that set them.

Event Status Enable Register (ESER): The ESER controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. The bits of the ESER correspond to the bits of the SESR.

Use the `*ESE` command to set the bits in ESER. Use the `*ESE?` query to read it.

OPERation Enable Register: Even though the OPERation Enable Register is present in the programmable EST's, the OPERation registers do not report any conditions.

QUEStionable Enable Register: The QUEStionable Enable Register controls which types of events are summarized by the QUES status bit in the SBR. Use the `STATus:QUEStionable:ENABle` command to set the bits in the QUEStionable Enable register. Use the `STATus:QUEStionable:ENABle?` query to read it.

Service Request Enable Register (SRER): The SRER controls which bits in the SBR generate a service request.

Use the `*SRE` command to set the SRER. Use the `*SRE?` query to read it.

Queues

The output queue is included to power supplies.

Output Queue: The programmable ESTs store query responses in the output queue by succeeding the IEEE 488.2 protocol. If the power supply receives a new command or query message after a message terminator, the power supply will clear and reset this queue each time. The computer must read a query response before it sends the next command (or query) or it loses response to earlier queries.

Error/Event Queues

When an error or event occurs, the output queue stores the message. The output queue stores and reports the messages on a FIFO (first in first out) state. The `SYSTem:ERRor?` query reads the next item from the output queue. If output queue overflows, the error message is –350, `"Queue overflow"`; the queue can't store or report succeeding messages till it is read or cleared.

Error Message

Table 8 lists the SCPI error messages for the programmable power supplies.

Table 8 The error messages for the power supplies

SCPI Error Code and Description	SESR Bit
0,"No error"	
-100,"Command error"	5
-200,"Execution Error"	4
-221,"Settings conflict"	4
-221,"Settings conflict; Timer setting error"	4

-221,"Settings conflict; Voltage setting error"	4
-221,"Settings conflict; Current setting error"	4
-221,"Settings conflict; Resistance setting error"	4
-221,"Settings conflict; Frequency setting error"	4
-221,"Settings conflict; Recall setting error"	4
-222,"Data out of range;"	4
-222,"Data out of range; Timer too large"	4
-222,"Data out of range; Voltage too large"	4
-222,"Data out of range; Current too large"	4
-222,"Data out of range; Resistance too large"	4
-222,"Data out of range; Timer too small"	4
-222,"Data out of range; Voltage too small"	4
-222,"Data out of range; Current too small"	4
-222,"Data out of range; Resistance too small"	4
-240,"Hardware Error"	3
-300,"Device-specific error"	3
-310,"System error"	3
-330,"Self-test failed"	3
-350,"Queue overflow"	3
-410,"Query INTERRUPTED"	2
-420,"Query UNTERMINATED"	2
-430,"Query DEADLOCKED"	2