Sorensen

SGX Series

RS232, Ethernet and IEEE 488.2

Programming Manual

About AMETEK

AMETEK Programmable Power, Inc., a Division of AMETEK, Inc., is a global leader in the design and manufacture of precision, programmable power supplies for R&D, test and measurement, process control, power bus simulation and power conditioning applications across diverse industrial segments. From bench top supplies to rack-mounted industrial power subsystems, AMETEK Programmable Power is the proud manufacturer of Elgar, Sorensen, California Instruments and Power Ten brand power supplies.

AMETEK, Inc. is a leading global manufacturer of electronic instruments and electromechanical devices with annualized sales of $4 billion. The Company has over 15,000 colleagues working at more than 150 manufacturing facilities and more than 0 sales and service centers in the United States and around the world.

Trademarks

AMETEK is a registered trademark of AMETEK, Inc. Sorensen is a trademark owned by AMETEK, Inc. Other trademarks, registered trademarks, and product names are the property of their respective owners and are used herein for identification purposes only.

Notice of Copyright

Exclusion for Documentation

UNLESS SPECIFICALLY AGREED TO IN WRITING, AMETEK PROGRAMMABLE POWER, INC. (“AMETEK”):

(a) MAKES NO WARRANTY AS TO THE ACCURACY, SUFFICIENCY OR SUITABILITY OF ANY TECHNICAL OR OTHER INFORMATION PROVIDED IN ITS MANUALS OR OTHER DOCUMENTATION.

(b) ASSUMES NO RESPONSIBILITY OR LIABILITY FOR LOSSES, DAMAGES, COSTS OR EXPENSES, WHETHER SPECIAL, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL, WHICH MIGHT ARISE OUT OF THE USE OF SUCH INFORMATION. THE USE OF ANY SUCH INFORMATION WILL BE ENTIRELY AT THE USER’S RISK, AND

(c) REMINDS YOU THAT IF THIS MANUAL IS IN ANY LANGUAGE OTHER THAN ENGLISH, ALTHOUGH STEPS HAVE BEEN TAKEN TO MAINTAIN THE ACCURACY OF THE TRANSLATION, THE ACCURACY CANNOT BE GUARANTEED. APPROVED AMETEK CONTENT IS CONTAINED WITH THE ENGLISH LANGUAGE VERSION, WHICH IS POSTED AT WWW.PROGRAMMABLEPOWER.COM.

Date and Revision

June 2018, A

Part Number

M551601-01

Contact Information

Telephone:	800 733 5427 (toll free in North America) 858 450 0085 (direct)
Fax:	858 458 0267
Email:	sales.ppd@ametek.com service.ppd@ametek.com
Web:	www.programmablepower.com

This page intentionally left blank.

Important Safety Instructions

Before applying power to the system, verify that your product is configured properly for your particular application.

⚠ **WARNING**	**Hazardous voltages may be present when covers are removed. Qualified personnel must use extreme caution when servicing this equipment. Circuit boards, test points, and output voltages also may be floating above (below) chassis ground.**
⚠ **WARNING**	**The equipment used contains ESD sensitive parts. When installing equipment, follow ESD Safety Procedures. Electrostatic discharges might cause damage to the equipment.**

Only *qualified personnel* who deal with attendant hazards in power supplies, are allowed to perform installation and servicing.

Ensure that the AC power line ground is connected properly to the Power Rack input connector or chassis. Similarly, other power ground lines including those to application and maintenance equipment *must* be grounded properly for both personnel and equipment safety.

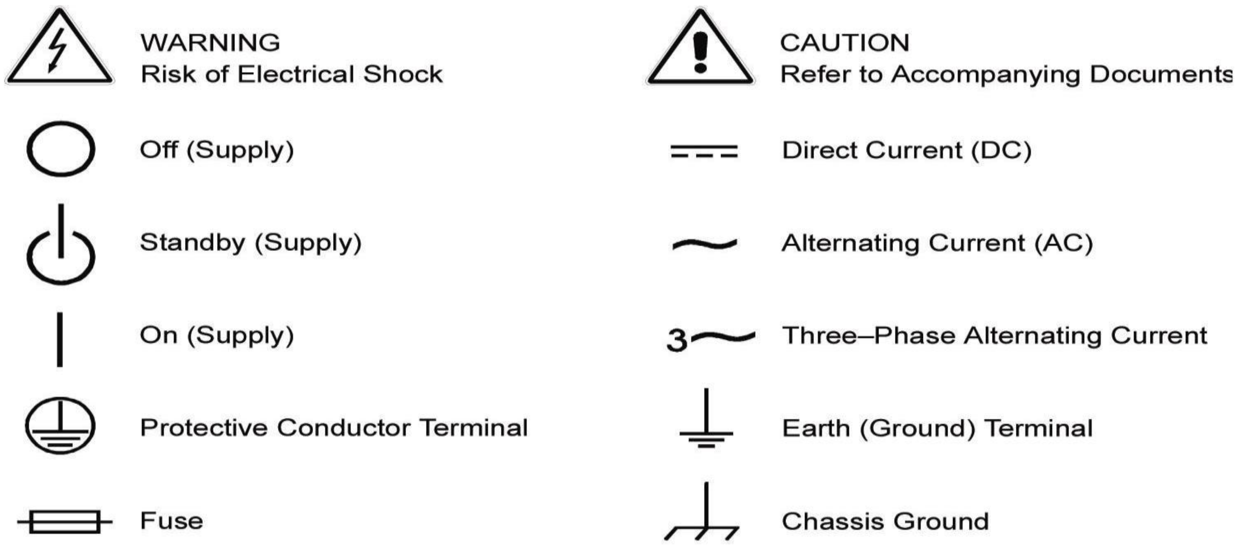
Always ensure that facility AC input power is de-energized prior to connecting or disconnecting any cable.

In normal operation, the operator does not have access to hazardous voltages within the chassis. However, depending on the user’s application configuration, **HIGH VOLTAGES HAZARDOUS TO HUMAN SAFETY** may be normally generated on the output terminals. The customer/user must ensure that the output power lines are labeled properly as to the safety hazards and that any inadvertent contact with hazardous voltages is eliminated.

Guard against risks of electrical shock during open cover checks by not touching any portion of the electrical circuits. Even when power is off, capacitors may retain an electrical charge. Use safety glasses during open cover checks to avoid personal injury by any sudden component failure.

Neither AMETEK Programmable Power Inc., San Diego, California, USA, nor any of the subsidiary sales organizations can accept any responsibility for personnel, material or inconsequential injury, loss or damage that results from improper use of the equipment and accessories.

SAFETY SYMBOLS



This page intentionally left blank.

Product Family: SGX Series Power Supply

Warranty Period: Five Years

WARRANTY TERMS

AMETEK Programmable Power, Inc. ("AMETEK"), provides this written warranty covering the Product stated above, and if the Buyer discovers and notifies AMETEK in writing of any defect in material or workmanship within the applicable warranty period stated above, then AMETEK may, at its option: repair or replace the Product; or issue a credit note for the defective Product; or provide the Buyer with replacement parts for the Product.

The Buyer will, at its expense, return the defective Product or parts thereof to AMETEK in accordance with the return procedure specified below. AMETEK will, at its expense, deliver the repaired or replaced Product or parts to the Buyer. Any warranty of AMETEK will not apply if the Buyer is in default under the Purchase Order Agreement or where the Product or any part thereof:

is damaged by misuse, accident, negligence or failure to maintain the same as specified or required by AMETEK;

is damaged by modifications, alterations or attachments thereto which are not authorized by AMETEK;

is installed or operated contrary to the instructions of AMETEK;

is opened, modified or disassembled in any way without AMETEK's consent; or

is used in combination with items, articles or materials not authorized by AMETEK.

The Buyer may not assert any claim that the Products are not in conformity with any warranty until the Buyer has made all payments to AMETEK provided for in the Purchase Order Agreement.

PRODUCT RETURN PROCEDURE

1. Request a Return Material Authorization (RMA) number from the repair facility (**must be done in the country in which it was purchased**):

   In the USA, contact the AMETEK Repair Department prior to the return of the product to AMETEK for repair:

   Telephone: 800-733-5427, ext. 2295 or ext. 2463 (toll free North America)
   858-450-0085, ext. 2295 or ext. 2463 (direct)

   Outside the United States, contact the nearest Authorized Service Center (ASC). A full listing can be found either through your local distributor or our website, www.programmablepower.com, by clicking Support and going to the Service Centers tab.

2. When requesting an RMA, have the following information ready:

   Model number

   Serial number

   Description of the problem

NOTE: Unauthorized returns will not be accepted and will be returned at the shipper's expense.

NOTE: A returned product found upon inspection by AMETEK, to be in specification is subject to an evaluation fee and applicable freight charges.

This page intentionally left blank.

CONTENTS

1 OVERVIEW 1-1

1.1 Introduction..1-1
1.2 RS232 Interface...1-1
1.3 IEEE 488.2 GPIB Interface 1-2
1.4 Ethernet Interface 1-2

2 FEATURES, FUNCTIONS AND SPECIFICATIONS 2-1

2.1 Introduction..2-1
2.2 Features of RS232 interface 2-1
2.3 Features of IEEE 488.2 GPIB interface 2-1
2.4 Features of Ethernet interface 2-2
2.5 Programmable Functions..2-2
2.6 Readback Functions..2-2
2.7 Specifications 2-3
2.7.1 Programming Resolution 2-3
2.7.2 Programming Accuracy 2-3
2.7.3 Readback Resolution...2-3
2.7.4 Readback Accuracy...2-3
2.8 Minimum System Requirements 2-4
2.8.1 PC Connection 2-4
2.8.2 Internet or LAN Connection 2-4

3 REMOTE/LOCAL SELECTION 3-1

3.1 Introduction..3-1

4 RS232 CONFIGURATIONS AND REMOTE PROGRAMMING..... 4-1

4.1 Rear Panel 4-1
4.2 Remote Programming Via RS232...4-2
4.2.1 RS232 Connector Pinout 4-2
4.2.2 RS232 Setup Procedure 4-3

5 IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING 5-1

5.1 Rear Panel 5-1
5.2 Remote Programming Via IEEE 488.2 GPIB 5-2
5.2.1 Address Selection..5-2
5.2.2 Power-On GPIB Service Request (PON SRQ) Selection 5-2
5.2.3 Shield Ground..5-2
5.2.4 IEEE 488.2 GPIB Setup Procedure 5-2

6 ETHERNET CONFIGURATION AND REMOTE PROGRAMMING 6-1

6.1 Introduction 6-1

6.2 Ethernet/LAN Configuration 6-1

6.2.1 Ethernet Configuration Factory Defaults 6-2

6.3 Rear Panel 6-2

6.4 Ethernet Setup Procedure 6-3

6.4.1 Network Setup Using DHCP Server 6-3

6.4.2 Network Setup Using Auto-IP (Direct Connection between SGX and PC using Cross cable) 6-6

6.4.3 Network Setup using Static IP 6-7

6.5 Programming/Communication Via Ethernet 6-8

6.5.1 Raw Socket Interface 6-8

6.5.2 VXI-11 Protocol 6-8

6.5.3 Web Server 6-8

6.6 Ethernet Web Pages, Overview 6-9

6.6.1 Home 6-10

6.6.2 IP Configuration 6-11

6.6.3 Interactive Control 6-13

6.6.4 LXI Identification 6-14

7 EXTERNAL USER CONTROL SIGNAL CONNECTOR 7-1

7.1 Introduction 7-1

7.2 External User Control Signal Connector details 7-2

7.3 Wiring Diagram 7-4

8 RS232/ETHERNET/ IEEE 488.2 GPIB AND SCPI COMMAND OPERATION 8-1

8.1 Introduction 8-1

8.2 Register Definitions 8-1

8.2.1 Protection Condition and Protection Event Status Register 8-1

8.2.2 Standard Event Status Register (ESR) 8-3

8.2.3 SCPI Status Byte 8-4

8.2.4 Error/Event Queue 8-5

8.2.5 Serial Poll Operation 8-6

8.3 SCPI Conformance Information 8-7

8.3.1 Parameter Definitions 8-7

8.3.2 Units 8-8

8.3.3 Conventions 8-8

8.3.4 Queries 8-8

8.4 IEEE 488.2 Common Command Subsystem 8-8

8.5 SOURCE SCPI Command Subsystem 8-10

8.5.1 SOURCE SCPI Command Summary 8-10

8.5.2 SOURCE SCPI Command Reference 8-11

8.5.3 RAMP FUNCTION..8-14
8.6 MEASURE SCPI Command Subsystem...8-15
8.6.1 MEASURE SCPI Command Summary 8-15
8.6.2 MEASURE SCPI Command Reference 8-15
8.7 OUTPUT SCPI Command Subsystem...8-16
8.7.1 OUTPUT SCPI Command Summary 8-16
8.7.2 OUTPUT SCPI Command Reference 8-16
8.8 STATUS SCPI Command Subsystem 8-17
8.8.1 STATUS SCPI Command Summary..8-17
8.8.2 STATUS SCPI Command Reference...8-18
8.9 SYSTEM SCPI Command Subsystem...8-19
8.9.1 SYSTEM SCPI Command Summary 8-19
8.9.2 SYSTEM SCPI Command Reference 8-20
8.10 HTRIGGER SCPI Command Subsystem 8-22
8.10.1 HTRIGGER SCPI Command Summary 8-22
8.10.2 HTRIGGER SCPI Command Reference.......................................8-22
8.11 TRIGGER SCPI Command Subsystem 8-23
8.11.1 TRIGGER SCPI Command Summary...8-23
8.11.2 TRIGGER SCPI Command Reference 8-23
8.12 CALIBRATION SCPI Command Subsystem...8-24
8.12.1 CALIBRATION SCPI Command Summary 8-24
8.12.2 CALIBRATION SCPI Command Reference 8-26
8.13 SGX-Unique Commands 8-29
8.13.1 Restrictions on Sequence Programming:......................................8-29
8.13.2 SGX SOURCE SCPI Command Subsystem..................................8-29
8.13.3 SGX PROGRAM SCPI Command Subsystem...............................8-31
8.13.4 SGX MEASURE SCPI Command Subsystem 8-36
8.13.5 SGX HTRIGGER SCPI Command Subsystem 8-37
8.14 Examples of Using the SCPI Commands 8-37
8.14.1 VI Mode Example 8-37
8.14.2 OVP Setup Example...8-38
8.14.3 Trigger Example 8-39
8.14.4 Hardware Trigger Example 8-39
8.14.5 Ramp V Example..8-40
8.14.6 Ramp I Example 8-40
8.14.7 Ramp V Example 2...8-40
8.14.8 Power On INIT Example 8-41
8.14.9 Sequence Creation and Execution Examples 8-41

9 CALIBRATION.. 9-1

9.1 Introduction...9-1
9.2 Setup for Calibration..9-2
9.3 Voltage Programming Calibration 9-3

9.4 Voltage Program Gain/Offset and Measurement Readback Calibration 9-5
9.5 Overvoltage Protection Programming Calibration 9-6
9.6 Current Programming Calibration 9-7
9.7 Current Programing gain/offset and Measurement Readback Calibration 9-9
9.8 Analog Program Adjustment 9-10

10 SCPI STATUS IMPLEMENTATION 10-1

10.1 SCPI Status Byte Implementation 10-1

LIST OF TABLES

Table 1. Remote/Local Switch....3-2
Table 2. Remote Mode Power-on Conditions....3-3
Table 3. External User Control Signal Connector Pinout – Ethernet only....7-3
Table 4. Example of Internal Relay Contact using SCPI Command....7-5
Table 5. Protection Condition and Event Status Registers....8-2
Table 6. Standard Event Status Register....8-3
Table 7. SCPI Status Byte....8-4
Table 8 SCPI Error Codes....8-5
Table 9. System Fault Registers....8-21

LIST OF FIGURES

Figure 3-1. Rear Panel - Remote/Local Configuration Switch 3-1
Figure 3-2. SGX Remote/Local Configuration Switch.. 3-1
Figure 3-3. Local/Remote Screen 3-2
Figure 4-1. Rear Panel – RS232 Interface 4-1
Figure 4-2. RS232 Rear Panel RJ-type 6P6C Connector Pinout................................ 4-2
Figure 4-3. RS232 Communications Cable Pinout 4-3
Figure 4-4. RS232 Screen (Configure)... 4-4
Figure 5-1. Rear Panel – IEEE 488.2 GPIB Interface 5-1
Figure 5-2. GPIB Screen... 5-2
Figure 6-1. Rear Panel - Ethernet Interface 6-2
Figure 6-2. LAN Screen (Configure)... 6-3
Figure 6-3. LAN Screen 6-4
Figure 6-4. LAN Screen (Settings) 6-5
Figure 6-5. RS232 Screen (Settings) 6-5
Figure 6-6. LAN Screen (Configure Static IP).. 6-7
Figure 6-7. SGX Banner and Tab... 6-9
Figure 6-8. Login Window 6-9
Figure 6-9. SGX Power Supply's Home Page 6-10
Figure 6-10. SGX IP Configuration Page 6-11
Figure 6-11. Interactive Control Page 6-13
Figure 6-12. LXI Identification Page 6-14
Figure 7-1. Rear Panel - External User Control Signal Connector.............................. 7-1
Figure 7-2. External User Connector Pinout (10-pin Molex, rear panel view) 7-2
Figure 7-3. Example of Open-Collector, TTL Input, and Relay Output Circuits........... 7-4
Figure 8-1. Protection Event Status and Protection Event Status Enable Register...... 8-2
Figure 8-2. Standard Event Status and Standard Event Status Enable Register......... 8-3
Figure 8-3. SCPI Status Byte and Service Request Enable Register 8-5
Figure 8-4. Power Supply Output for Example 1 8-43
Figure 8-5. Power Supply Output for Example 2 8-44
Figure 8-6. Power Supply Output for Example 3 8-45
Figure 8-7. End-of-Sequence Pause for Example 4 8-47
Figure 8-8. Power Supply Output for Example 5 8-50
Figure 8-9. Power Supply Output for Example 6 8-51
Figure 8-10. Power Supply Output for Example 7 8-53
Figure 10-1. SCPI Status Implementation Diagram.. 10-3

This page intentionally left blank.

1
OVERVIEW

1.1 INTRODUCTION

This manual provides instructions for remote programming control and monitoring from a computer, for your SGX series high power DC power supply. For easy navigation to the applicable instructions, this manual separates RS232, IEEE 488.2 GPIB and Ethernet setup instructions. The instructions then converge where they are common to all three interface options. See Section 1.2, Section 1.3 and Section 1.4 for orientation. The SGX series provides default interface of RS232 and Ethernet. IEEE 488.2 GPIB is an optional interface. Use this programming manual in conjunction with your SGX Operation manual.

1.2 RS232 INTERFACE

If you are using the RS232 interface, go to:

- 2 for Features, Functions and Specifications
- 3 for Remote/Local Selection
- 4 for RS232 Configuration and Remote Programming
- 7 for External User Control Signal Connector
- 8 for SCPI Commands and Definitions
- 9 for Calibration Procedures
- 10 for SCPI Status Implementation

1.3 IEEE 488.2 GPIB INTERFACE

If you are using the IEEE 488.2 GPIB interface, go to:

- 2 for Features, Functions and Specifications,
- 3 for Remote/Local Selection
- 5 for IEEE 488.2 GPIB Configuration and Remote Programming
- 7 for External User Control Signal Connector
- 8 for SCPI Commands and Definitions
- 9 for Calibration Procedures
- 10 for SCPI Status Implementation

1.4 ETHERNET INTERFACE

If you are using an Ethernet interface, go to:

- 2 for Features, Functions and Specifications
- 3 for Remote/Local Selection
- 6 for Ethernet Configuration and Remote Programming
- 7 for External User Control Signal Connector
- 8 for SCPI Commands and Definitions
- 9 for Calibration Procedures
- 10 for SCPI Status Implementation

2
FEATURES, FUNCTIONS AND SPECIFICATIONS

2.1 INTRODUCTION

This section introduces the features, functions and specifications for RS232, IEEE 488.2 GPIB and Ethernet. Programmable, readback functions and specifications are applicable to all the communication interfaces.

2.2 FEATURES OF RS232 INTERFACE

- Programming and readback of voltage and current
- Programmable overvoltage protection with reset
- SCPI compliant command set
- User selectable Constant-Voltage/Constant-Current or Foldback mode, with reset
- Voltage Ramp and Current Ramp functions
- Field-upgradable firmware via RS232
- Soft calibration
- Rear panel RS232 control interface
- Rear panel External User Control Signal Interface (Common to all interfaces)
- Remote/Local Configuration Switch (Common to all interfaces)

2.3 FEATURES OF IEEE 488.2 GPIB INTERFACE

- Programming and readback of voltage and current
- Programmable overvoltage protection with reset
- IEEE 488.2 and SCPI compliant command set
- User selectable Constant-Voltage/Constant-Current or Foldback mode, with reset
- Voltage Ramp and Current Ramp functions
- Soft calibration
- Rear panel GPIB IEEE 488.2 control interface

- Rear panel External User Control Signal Interface (Common to all interfaces)
- Remote/Local Configuration Switch (Common to all interfaces)

2.4 FEATURES OF ETHERNET INTERFACE

- Ethernet/LAN connectivity, 10/100base-T compatible
- Fully LXI™ (LAN eXtensions for Instrumentation) class C compliant
- Built-in Web Server for direct control using Web Browser
- Programming and readback of voltage and current
- Programmable overvoltage protection with reset
- SCPI compliant command set
- User selectable Constant-Voltage/Constant-Current or Foldback mode, with reset
- Voltage Ramp and Current Ramp functions
- Full calibration through software control
- Rear panel Ethernet control interface
- Rear panel External User Control Signal Interface (Common to all interfaces)
- Remote/Local Configuration Switch (Common to all interfaces)

2.5 PROGRAMMABLE FUNCTIONS

The below functions are common to all communication interfaces (RS232, IEEE 488.2 GPIB and Ethernet).

- Output voltage and current
- Soft limits for voltage and current
- Overvoltage protection
- Output enable/disable
- Maskable fault interrupt
- Hold and trigger
- External relay control.
- Full calibration

2.6 READBACK FUNCTIONS

The below functions are common to all communication interfaces (RS232, IEEE 488.2 GPIB and Ethernet).

- Actual measured voltage and current
- Voltage and current settings
- Soft voltage and current limits
- Overvoltage protection setting
- Status and Accumulated Status registers

- Programming error codes
- Fault codes
- Manufacturer, power supply model, and firmware version identification

2.7 SPECIFICATIONS

Specifications are subject to change without notice. Refer to your SGX power supply operation manual for effects of line regulation, load regulation, and temperature on accuracy specifications. Specifications are common to all interfaces (RS232, IEEE 488.2 GPIB and Ethernet).

2.7.1 Programming Resolution

Voltage: 0.002% of full scale

Current: 0.002% of full scale

Overvoltage Protection: 0.002% of full scale (full scale is 110% of max output voltage.)

2.7.2 Programming Accuracy

Voltage: ± (0.1% of maximum output voltage)

Current: ± (0.4% of maximum output current) *

Overvoltage Protection: ± (1.0% of max output voltage)

***** After 30 minutes operation with fixed line, load, and temperature.

2.7.3 Readback Resolution

Voltage: ± 0.002% of full scale

Current: ± 0.002% of full scale

2.7.4 Readback Accuracy

Voltage: ± (0.15% of full scale output voltage)

Current: ± (0.4% of full scale output current) *****

***** After 30 minutes operation with fixed line, load, and temperature.

2.8 MINIMUM SYSTEM REQUIREMENTS

The minimum software and equipment requirements to operate your SGX Power Supply depend on whether it is connected directly to your PC or connected to the Internet or to a Local Area Network (LAN).

2.8.1 PC Connection

To operate your SGX Power Supply with Ethernet option connected directly to a PC (no Internet or LAN connection), you will need:

- Pentium-based laptop or desktop computer running Microsoft Windows 7 (or better)
- Ethernet based Network Interface Card (NIC) or built-in port capable of 10/100 MBit operation
- CAT 5 cable Ethernet crossover cable
- Web Browser

2.8.2 Internet or LAN Connection

To operate your SGX Power Supply connected to the Internet or a LAN you will need:

- Pentium-based laptop or desktop computer running Microsoft Windows 7 (or better)
- Ethernet based Network Interface Card (NIC) or built-in port capable of 10/100 MBit operation
- Appropriate Ethernet modem for Internet connection, or
- Switch or hub (Linksys brand strongly recommended) for LAN connection
- Standard CAT 5 Ethernet interconnect cable
- Web Browser

3
REMOTE/LOCAL SELECTION

3.1 INTRODUCTION

The Remote/Local Configuration Switch for RS232, IEEE 488.2 GPIB and Ethernet is shown in Figure 3-1.

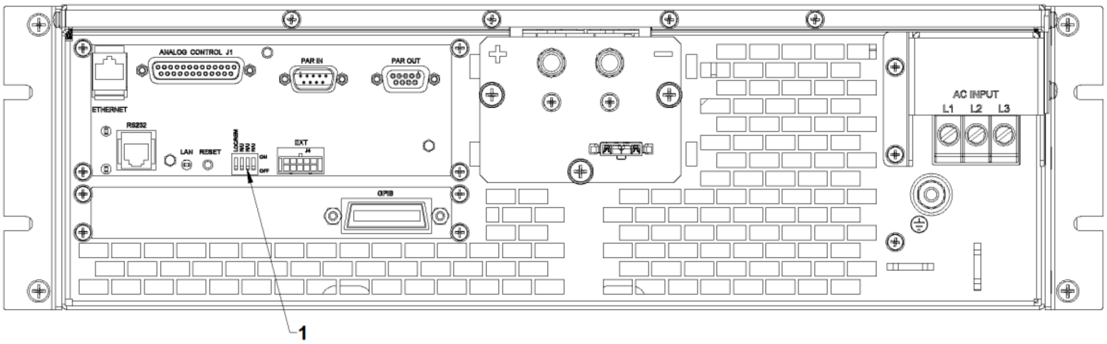


Figure 3-1. Rear Panel - Remote/Local Configuration Switch

1 – Remote/Local Configuration Switch

The first dip-switch in the SGX 4-pin Configuration Switch is used for Remote/Local selection, see Figure 3-2.

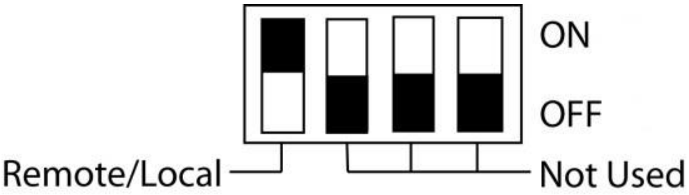


Figure 3-2. SGX Remote/Local Configuration Switch

Set the rear panel Remote/Local switch to select remote or local operation.

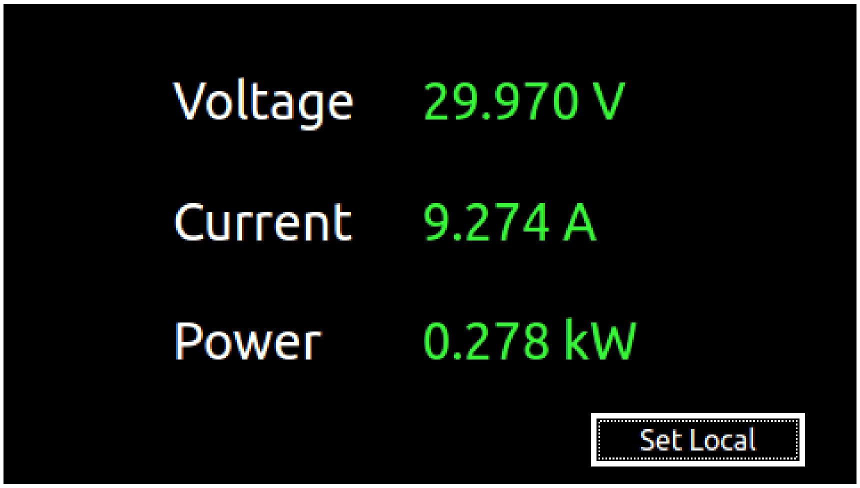
Table 1. Remote/Local Switch

Switch Position	**Description**
ON	Remote operation selected. *****
OFF	Local operation selected, and front panel control is enabled. **NOTE:** Unit will switch to remote operation upon issuing the first non-query command.

***** In the ON position, the SGX Power Supply initializes to the remote state at power-on. In this state, control of the power supply is possible only through communication interfaces (RS232, Ethernet and IEEE 488.2 GPIB).

To revert to front panel control, do one of the following:

- use the special command **SYST:LOCAL** <on/off>.
- Press Set Local button on the Local/Remote screen on the Front Panel (see Figure 3-3, for details refer to SGX Operation Manual, M551600-01).



Voltage 29.970 V

Current 9.274 A

Power 0.278 kW

Figure 3-3. Local/Remote Screen

Powering up in remote mode will result in the operating conditions described in Table 2.

Table 2. Remote Mode Power-on Conditions

Condition	Default
Voltage	0 Volts (initial from factory power–on voltage); otherwise, last value saved by SCPI command. ****** See CAL:INIT:VOLT to change.
Current	0 Amps (initial from factory power–on current); otherwise, last value saved by SCPI command. ****** See CAL:INIT:CURR to change.
Soft Voltage Limit	Model maximum voltage *****
Soft Current Limit	Model maximum current *****
OVP Trip Voltage	Model maximum voltage +10% (initial from factory power–on OVP); otherwise, last value saved by SCPI command. ****** See CAL:INIT:VOLT:PROT to change.
Delay	*** 0.5 seconds
Foldback Protection	*** OFF
Output	ON ****** See CAL:MOD:POWERON
Hold	*** OFF
Unmask	*** NONE
Service Request Capability	*** OFF

***** User-programmable temporary limit (reverts to power-on defaults after power cycle or Reset command is issued)

****** User-selectable

******* Non-configurable

This page intentionally left blank.

4
RS232
CONFIGURATIONS
AND REMOTE PROGRAMMING

4.1 REAR PANEL

This section provides illustrations of the SGX power supply's rear panel layout, which differs among the SGX models. For example, see Figure 4-1. Regardless of the layout, the component functions are common across all models, and those that are pertinent to the RS232 options are described here.

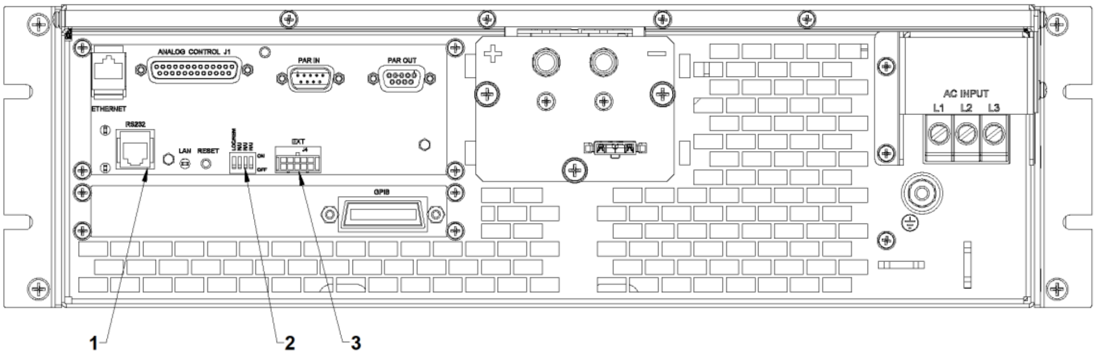


Figure 4-1. Rear Panel – RS232 Interface

1 – RS232 (RJ-type 6P6C) Connector.

2 – Remote/Local Selection Switch (4-pin) - for correct settings, refer to 3.

3 – External User Control Signal Connector, refer to 7.

4.2 REMOTE PROGRAMMING VIA RS232

4.2.1 RS232 Connector Pinout

The RS232 interface operates at a default baud of 19.2K. The baud is selectable from 9600 to 115200.

The RS232 interface is accessible through the rear panel RJ-type 6P6C (6-pin) connector (see Figure 4-2), labeled RS232 on the power supply's rear panel (see Figure 4-1).

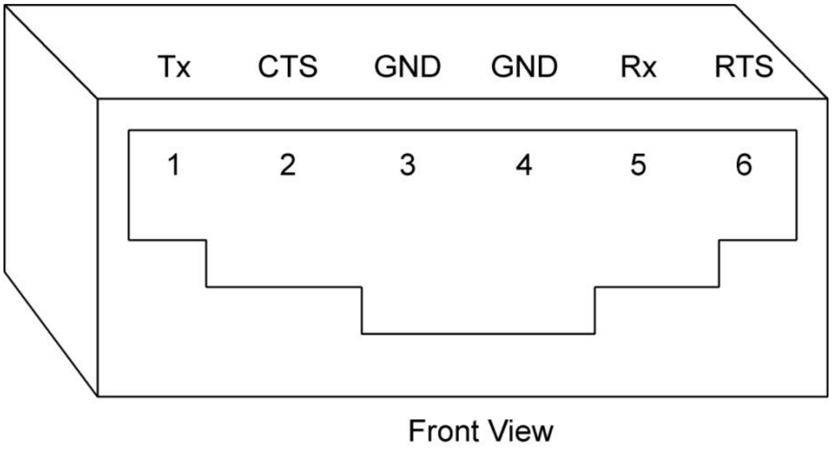


Figure 4-2. RS232 Rear Panel RJ-type 6P6C Connector Pinout

4.2.2 RS232 Setup Procedure

This section provides a quick reference for the configuration requirements for RS232.

Parameter	Setting	Notes
Baud Rate	Selectable from 9600 to 11520	The baud rate is selectable through the front panel
Data Bits	8	Not Selectable
Stop Bits	1	Not Selectable
Parity	None	Not Selectable
Incoming Termination Character	• CR (Carriage Return): HEX, 0x0d (DEC, 13), • LF (Line Feed): HEX, 0x0a (DEC, 10), • CR LF (Carriage Return and Line Feed): HEX, 0x0d 0x0a (DEC, 13 10) • LF CR (Line Feed and Carriage Return): HEX, 0x0a 0x0d (DEC, 10 13)	
Outgoing Termination Character(s)	CR LF (Carriage Return and Line Feed): HEX, 0x0d 0x0a (DEC, 13 10)	

1. Build an RS232 communications cable as per the pinout description illustrated in Figure 4-3 (with crossover of signals Rx/Tx and CTS/RTS):

RJ-type 6P6C Plug Terminal	D-Subminiature 9-Pin Connector Female Socket
1	2
2	7
3	No Connection
4	5
5	3
6	8

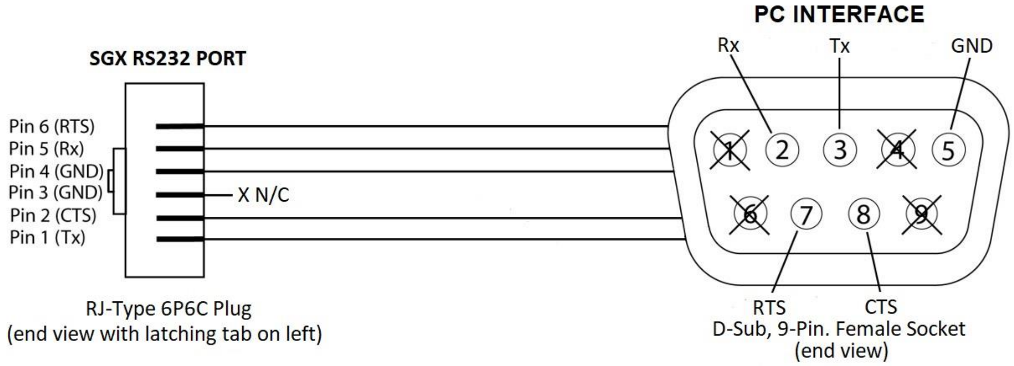


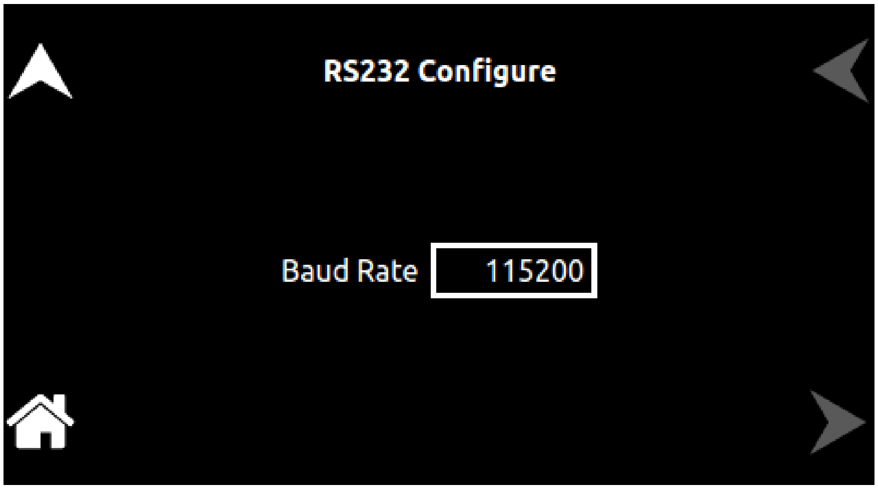
Figure 4-3. RS232 Communications Cable Pinout

2. Set the rear panel Remote/Local switch to Remote (ON).

3. Connect power to the unit and turn on the unit.

4. The baud rate is selectable through the front panel, see Figure 4-4. Change the baud rate for RS232 to 19200.

   To navigate to RS232 Configure screen, go to Home → Control Interface → RS232 → RS232 Configure. For details refer to SGX Operation Manual, M551600-01.



RS232 Configure

Baud Rate 115200

Figure 4-4. RS232 Screen (Configure)

5. Use one of the available programs for serial communication, such as MS HyperTerminal™, and set the RS232 baud rate to 19200, 8 data bits, no parity, 1 stop bit, and no flow control.

   If you choose to use MS HyperTerminal™:

   a. After inputting the above parameters, in the HyperTerminal™ window click the disconnect icon and then the properties icon.
   b. In the properties window select the Settings tab.
   c. In the Settings window click the ASCII Setup button.
   d. In the ASCII Setup window in the ASCII Sending section, check "Echo typed characters locally" and in the ASCII Receiving section, check "Append line feeds to incoming line ends." Leave all other check boxes in their default state.

6. Establish communication.

7. Test the communication interface by issuing the *IDN? Command. This returns the supply's model and serial numbers, and software version(s). This command does not affect the output of the supply.

5
IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING

5.1 REAR PANEL

This section provides illustrations of the SGX power supply's rear panel layout, which differs among the SGX models. For example, see Figure 5-1. Regardless of the layout, the component functions are common across all models, and those that are pertinent to the IEEE 488.2 GPIB options are described here.

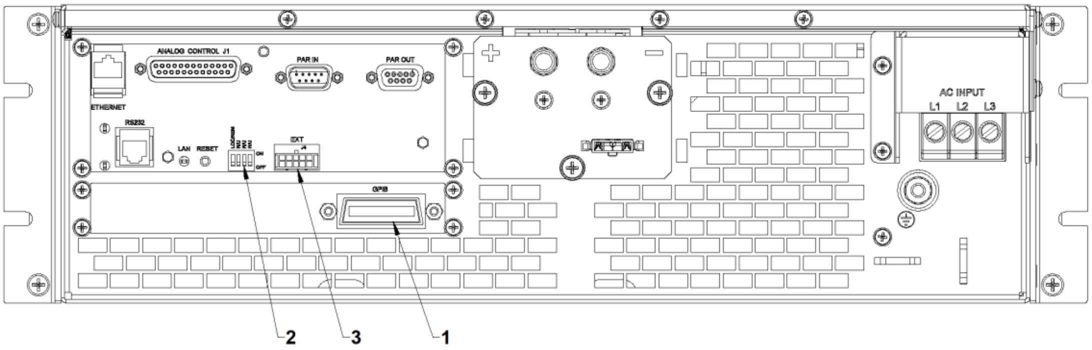


Figure 5-1. Rear Panel – IEEE 488.2 GPIB Interface

1 – IEEE 488.2 GPIB Connector

2 – Remote/Local Selection Switch (4-pin) - for correct settings, refer to 3

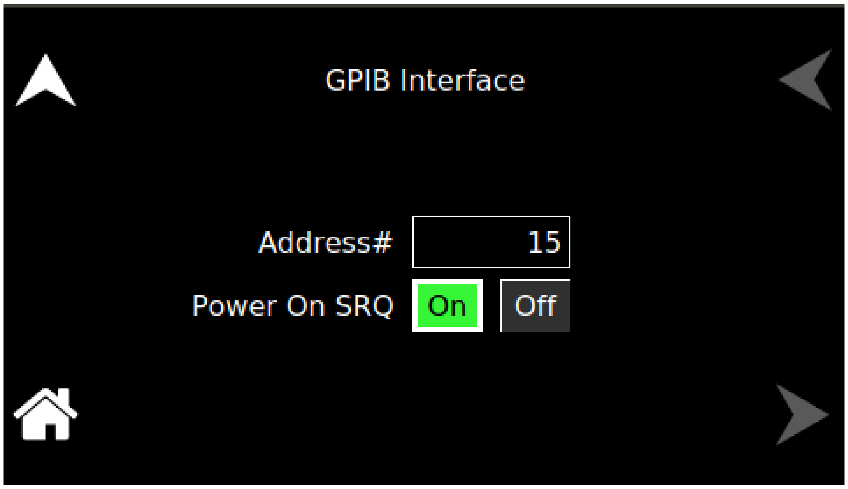
3 – External User Control Signal Connector, refer to 7

5.2 REMOTE PROGRAMMING VIA IEEE 488.2 GPIB

5.2.1 Address Selection

GPIB address for unit can be (1-30). SCPI reserves channel 0 as the global channel to address all channels.

SGX GPIB address can be configured by using the front panel menu, see Figure 5-2. To navigate to GPIB screen, go to Home → Control Interface → GPIB. For details refer to SGX Operation Manual, M551600-01.



GPIB Interface

Address# 15

Power On SRQ [✓ On] Off

Figure 5-2. GPIB Screen

5.2.2 Power-On GPIB Service Request (PON SRQ) Selection

PON SRQ can be enable using the front panel menu, see Figure 5-2. When PON SRQ is enabled, a GPIB service request will be sent by the power supply to the computer controller on Power ON.

5.2.3 Shield Ground

Connects GPIB cable shield to chassis ground.

5.2.4 IEEE 488.2 GPIB Setup Procedure

1. Set the rear panel Local/Remote switch to Remote (ON).
2. Set the GPIB address via the front panel menu, see Figure 5-2.
3. Connect GPIB cable from the controlling computer to the SGX power supply. NOTE: If operating in an inherently noisy environment, e.g., high RF or other radiated emissions, a double-shielded GPIB cable is recommended.
4. Connect power to the unit and turn on the unit.
5. Using a GPIB communication software, test the communication interface by issuing the *IDN? Command. This returns the supply's model and serial numbers, and software version(s). This command does not affect the output of the supply.

6 ETHERNET CONFIGURATION AND REMOTE PROGRAMMING

6.1 INTRODUCTION

This section covers the Remote Programming Ethernet Interface for the SGX series power supplies. This configuration enables you to operate your SGX power supply from a computer via Ethernet IEEE-802.3 communication protocol, with SCPI-compatible language, allowing full remote programming control and monitoring of your power supply.

An important point is that this Ethernet option is LXI™ (1.5 LXI™ device specification 2016) class C compliant. LXI™ is an instrumentation platform based on industry-standard Ethernet technology designed to provide ease of integration by modularity, flexibility and performance.

6.2 ETHERNET/LAN CONFIGURATION

- Ethernet Standard: IEEE-802.3 compliant
- Technology: 10/100Base-T
- Protocol: TCP/IP, IPV4
- ICMP (ping server): Always Enabled
- mDNS/DNS-SD: Always Enabled
- IP Address Assignment: Via DHCP or Static IP
- VXI-11 Discovery: Supported

6.2.1 Ethernet Configuration Factory Defaults

PARAMETER	DEFAULT
Host Name	SGx<base model>-<last four digits of serial number>
Description	SGX Power Supply <base model>
IP Address	* DHCP-acquired, If DHCP server absent, assigned via Auto-IP
Subnet Mask	* DHCP-acquired, If DHCP server absent, assigned via Auto-IP
Gateway	* DHCP-acquired, If DHCP server absent, assigned via Auto-IP
DNS Server	0.0.0.0
Listening Port	9221
User ID	admin
Password	password

***** The Ethernet interface provides the opportunity to assign an IP address via Auto-IP. If DHCP server fails to assign an IP address and Auto-IP setting is ON, the unit gets an IP address in the range of 169.254.X.X.

6.3 REAR PANEL

This section provides illustrations of the SGX power supply's rear panel layout, which differs among the SGX models. For example, see Figure 6-1. Regardless of the layout, the component functions are common across all models and those that are pertinent to the Ethernet option are described here.

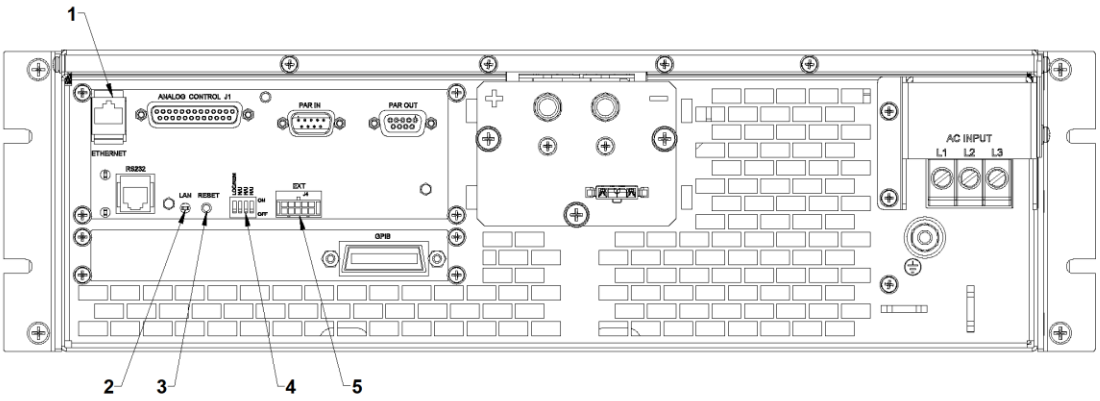


Figure 6-1. Rear Panel - Ethernet Interface

1 – Ethernet (RJ-45) connector. Adjacent to the RJ-45 connector are two green LEDs. If one of the LEDs is lit, the link is connected either to a hub switch or to another host. If both are lit, the connection speed is 100MB.

2 – LAN LED: - When solid-lit, indicates Network Connectivity; When blinking, indicates Instrument ID. If the LED is off, there is no Ethernet connection found by the power supply.

3 – Reset switch:- (must be depressed until LAN LED starts blinking, which could take five or more seconds), it returns the Ethernet configuration parameters to factory default settings, refer to Section 6.2.1.

4 – Remote/Local Selection Switch (4-pin) (Dip Switch) – for correct settings, refer to section 3.

5 – External User Control Signal Connector (Molex Connector), refer to section 7.

6.4 ETHERNET SETUP PROCEDURE

There are three ways to setup the Ethernet network in the SGX power supply.

- Network setup using DHCP Server
- Network Setup using Auto-IP (Direct Connection between SGX and PC using Cross cable)
- Network Setup using Static IP

The network setups are described in the subsections that follow. Use the Setup procedure that applies to your system and application to configure the Ethernet.

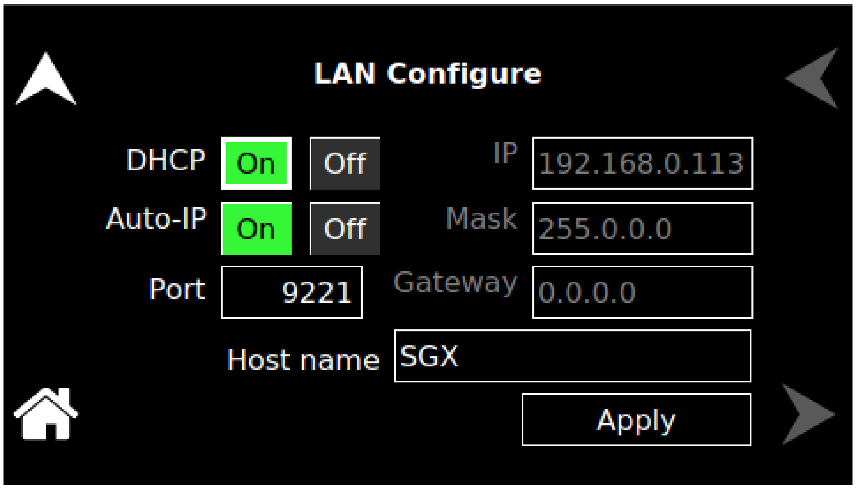
NOTE: When connecting your SGX power supply to a network, it is strongly recommended to use Linksys® hubs or switches, which have undergone extensive compatibility testing with the Ethernet interface.

6.4.1 Network Setup Using DHCP Server

For this network setup to work, DHCP mode must be enabled. DHCP mode can be enabled using the front panel or the serial interface.

6.4.1.1 DHCP MODE SELECTION USING FRONT PANEL

Navigate to Home → Control Interface → LAN → LAN Configure. Make both DHCP and Auto-IP as ON to use the DHCP mode of operation, See Figure 6-2. For details refer to SGX Operation Manual, M551600-01.


LAN Configure

DHCP: On (selected) | Off

Auto-IP: On (selected) | Off

Port: 9221

IP: 192.168.0.113

Mask: 255.0.0.0

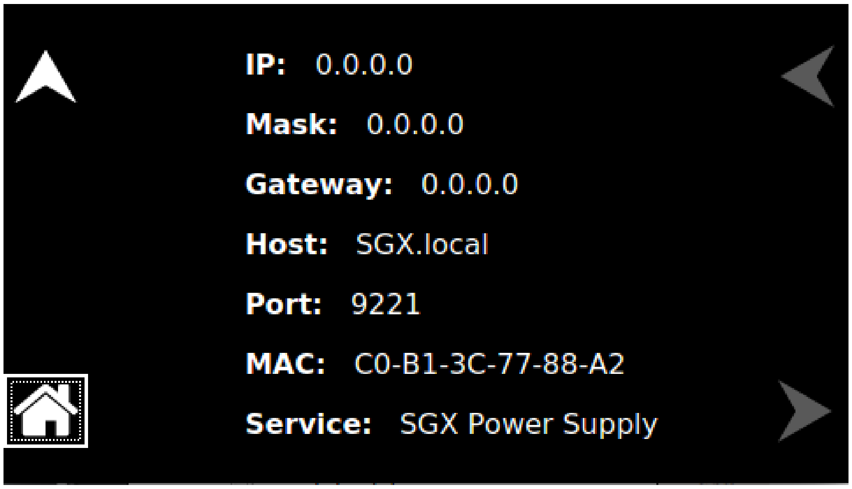
Gateway: 0.0.0.0

Host name: SGX

Figure 6-2. LAN Screen (Configure)

NOTE: If DHCP server is not available and Auto-IP is enabled, the unit can assign itself an IP address in the Auto-IP (dynamic link local addressing) range.

NOTE: In DHCP mode of operation, if Auto-IP is Off and DHCP server is not available, the IP address will default to 0.0.0.0. See Figure 6-3. This configuration is not usable for network connection.



IP: 0.0.0.0

Mask: 0.0.0.0

Gateway: 0.0.0.0

Host: SGX.local

Port: 9221

MAC: C0-B1-3C-77-88-A2

Service: SGX Power Supply

Figure 6-3. LAN Screen

6.4.1.2 DHCP MODE SELECTION USING SERIAL INTERFACE

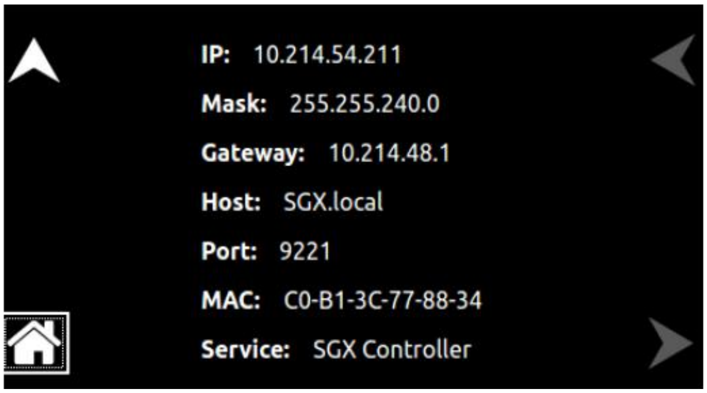
1. Connect using a computer serial communications program such as MS HyperTerminalTM and establish communication as described in section 4.2.2.
2. Turn ON DHCP mode using the SCPI command, SYST:NET:DHCPMODE 1.
3. Turn ON AUTO-IP mode using the SCPI command, SYST:NET:AUTOIP 1.
4. Type SYST:NET:APPLY <enter> to apply the Network settings.
5. After configuring the settings, verify with the queries, SYST:NET:DHCPMODE? and SYST:NET:AUTOIP?.

6.4.1.3 IP ADDRESS IDENTIFICATION FOR DHCP MODE OF OPERATION

1. Start with the power supply in the power-off state.
2. Connect a RJ-45 network cable from the power supply to the network with the DHCP server.
3. Power on the power supply and allow the power supply to perform its initialization.
4. Identify the IP address assigned to the power supply by accessing the DHCP server, by any of four ways:
   a. Asking your network administrator.
   b. Discovering it with a VXI-11 compliant discover program.

NOTE: The power supply is VXI-11 compliant, so even without access to the DHCP server, it is still possible to discover the IP address assigned to the power supply with programs such as National Instrument's NI-VISA.

c. Using front panel and navigating to Home → Control Interface → LAN → LAN Settings. For details refer to SGX Operation Manual, M551600-01, see Figure 6-4.

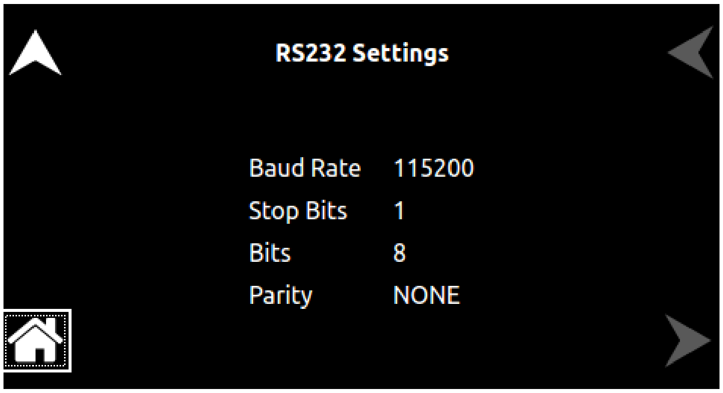


IP: 10.214.54.211
Mask: 255.255.240.0
Gateway: 10.214.48.1
Host: SGX.local
Port: 9221
MAC: C0-B1-3C-77-88-34
Service: SGX Controller

Figure 6-4. LAN Screen (Settings)

d. Using a computer serial communications program such as HyperTerminal^{TM}, set for the same baud as SGX power supply (see Figure 6-5) no parity, 8 data bits, 1 stop bit. Use SCPI command **SYST:NET:IP?** ‹Enter› to get the IP address.

To find RS232 Settings of the power supply, navigate to Home → Control Interface → RS232 → RS232 Settings. For details refer to SGX Operation Manual, M551600-01.



RS232 Settings

Baud Rate 115200
Stop Bits 1
Bits 8
Parity NONE

Figure 6-5. RS232 Screen (Settings)

5. The SGX Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page of the power supply. See Figure 6-9.

6.4.2 Network Setup Using Auto-IP (Direct Connection between SGX and PC using Cross cable)

This setup requires that DHCP and Auto-IP is ON (see Figure 6-2). DHCP and Auto-IP can be enabled using the front panel or the serial interface as described in the Section 6.4.1.1 and Section 6.4.1.2 respectively.

Since the setup is not connected to the DHCP server, SGX Power Supply will assign itself an IP address in the IP address range from 169.254.0.1 to 169.254.255.254 with a subnet mask of 255.255.0.0.

6.4.2.1 IP ADDRESS IDENTIFICATION FOR AUTO-IP MODE OF OPERATION

1. Start with the power supply in the power-off state.
2. Connect a crossover cable from the SGX power supply directly to your PC.
3. If the PC is already configured to obtain an IP address automatically, skip to Step 4. Otherwise:
   a. In Windows click **Start**, **Settings**, **Control Panel**.
   b. Click open **Network Connections**.
   c. In the Network Connections window, right click the icon for the network adapter used to connect to the power supply, and click **Properties**.
   d. Find the TCP/IP protocol item under the **Configuration** tab and click **Properties**. Select **Obtain an IP Address Automatically.**
   e. Click **OK** to save the change.
   f. Click **OK** again to apply the settings to the network adapter.
4. In Windows, click **Start**, and then **Run…**
5. In the Run window, type “ipconfig /release” and click **OK**.
6. Again, click **Start,** and then **Run…**
7. In the Run window, type “ipconfig /renew” and click **OK**. Your PC will assign itself an IP address in the Auto-IP range.
8. Power on the power supply and allow the power supply to perform its initialization.
9. Identify the IP address assigned to the power supply by following ways.
   a. Discovering it with a VXI-11 compliant discover program.
   b. Using front panel and navigating to Home → Control Interface → LAN → LAN Settings. For details refer to SGX Operation Manual, M551600-01, see Figure 6-4.
   c. Using a computer serial communications program such as HyperTerminal^{TM}, set for the same baud as SGX power supply (see Figure 6-5) no parity, 8 data bits, 1 stop bit. Use SCPI command **SYST:NET:IP?** ⟨Enter⟩ to get the IP address.

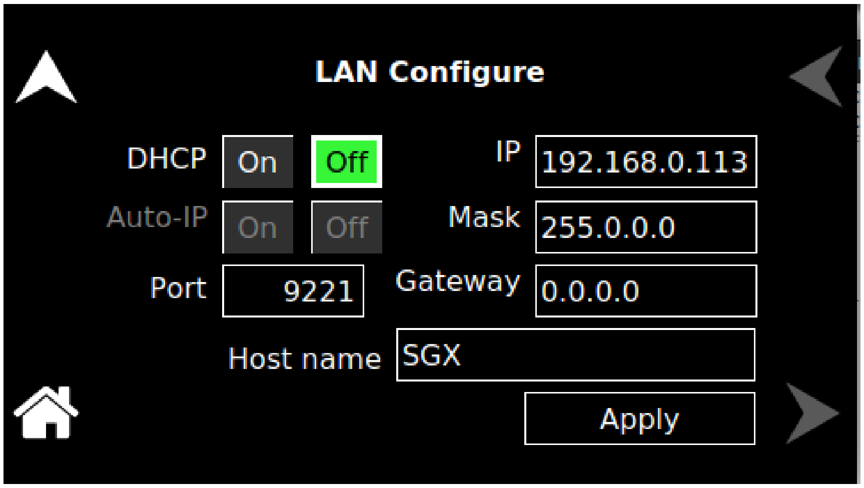
10. The SGX Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page of the power supply. See Figure 6-9.

6.4.3 Network Setup using Static IP

This setup requires that DHCP is OFF. DHCP mode can be made OFF using the front panel or using the serial interface.

6.4.3.1 STATIC IP SETUP USING FRONT PANEL

1. Navigate to Home → Control Interface → LAN → LAN Configure. Make the DHCP as OFF, See Figure 6-6. For details refer to SGX Operation Manual, M551600-01.



DHCP: On [✓ Off]
Auto-IP: On Off
Port: 9221
IP: 192.168.0.113
Mask: 255.0.0.0
Gateway: 0.0.0.0
Host name: SGX

Figure 6-6. LAN Screen (Configure Static IP)

2. Enter the required IP address, Subnet Mask and Gateway (See Figure 6-6).
3. Press "Apply" button for the settings to take effect.
4. The SGX Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page of the power supply. See Figure 6-9.

6.4.3.2 STATIC IP SETUP USING THE SERIAL INTERFACE

1. Connect using a computer serial communications program such as HyperTerminal™, set for the same baud as SGX power supply (see Figure 6-5) no parity, 8 data bits, 1 stop bit. Refer section 4.2.2 for more details.
2. Turn OFF DHCP mode using the SCPI command, SYST:NET:DHCPMODE 0.
3. Set the IP address by typing **SYST:NET:IP "xxx.xxx.xxx.xxx"** <enter> (where xxx.xxx.xxx.xxx is the new IP address). For example, to set 192.168.0.200 as the IP address, type SYST:NET:IP "192.168.0.200" <enter>.

   NOTE: The format requires a single space after SYST:NET:IP and double quotes around the IP address numbers.

4. Set the subnet mask with SYST:NET:MASK xxx.xxx.xxx.xxx <enter>.
5. Set the gateway with SYST:NET:GATE xxx.xxx.xxx.xxx <enter>.
6. Type SYST:NET:APPLY <enter> to apply the Static IP configuration.
7. After configuring all settings, verify with the queries, SYST:NET:IP? <enter>.
8. SYST:NET:GATE? <enter> and SYST:NET:MASK? <enter>.
9. The SGX Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page of the power supply. See Figure 6-9.

6.5 PROGRAMMING/COMMUNICATION VIA ETHERNET

With the Ethernet option, there are three basic methods to communicate with the power supply from a PC:

- Raw socket interface, sending delimited strings
- Application program that utilizes VXI-11 Discovery protocol
- Web browser

6.5.1 Raw Socket Interface

The essential components of communicating via a raw socket interface are the socket number, IP address and command delimiter. The default values are: socket = 9221, IP address = 192.168.0.200 (when DHCP is disabled), and delimiter = line feed <CRLF>. We can set the static IP address, Subnet Mask and Gateway using web browser (refer to Section 6.6.2 **IP Configuration**) or the RS232C interface (refer to Section 8.9 **System SCPI command**) or Front Panel (see Figure 6-6).

For convenience and to comply with the proposed LXI™ standard, the VISA resource name is available on the home page of the power supply's Web server.

6.5.2 VXI-11 Protocol

With programs such as National Instrument's NI-VISA, the VXI-11 protocol allows the power supply to be easily configured in a test system.

6.5.3 Web Server

To communicate with the power supply via the built-in Web server, open a Web browser and type the IP address of the power supply in the "Address" field. Tap the ENTER key to launch the power supply's Ethernet Web page interface.

6.6 ETHERNET WEB PAGES, OVERVIEW

The layout of each of the Web pages includes the banner showing the Model (e.g SGX 100/150), Manufacturer (AMETEK Programmable Power), AMETEK Logo and the Device name (e.g LXI-SGX-100/150-9999A99999). Below the SGX banner are four tabs, each linked to its corresponding page. See Figure 6-7.

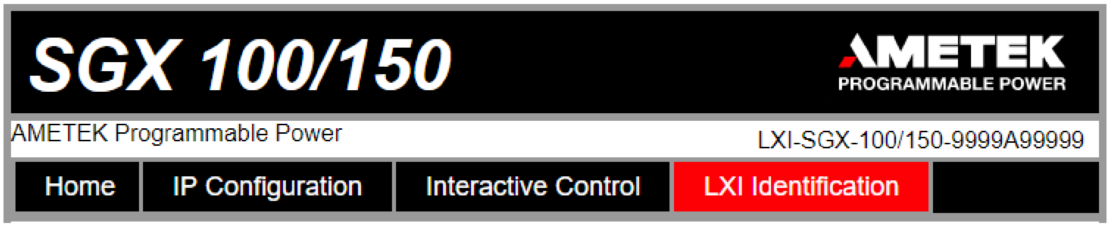
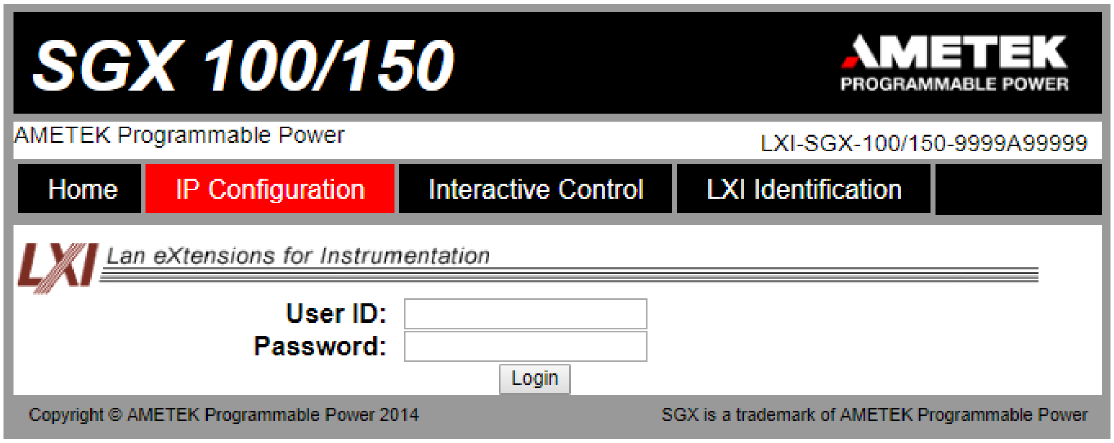


Figure 6-7. SGX Banner and Tab

When navigating to the Ethernet Web pages by clicking their tabs, you will find that the HOME page (default), Interactive Control and LXI Identification can be accessed without logging in. You must enter User ID and Password (For default values, refer to Section 6.2.1) for accessing the IP Configuration tab. See Figure 6-8.

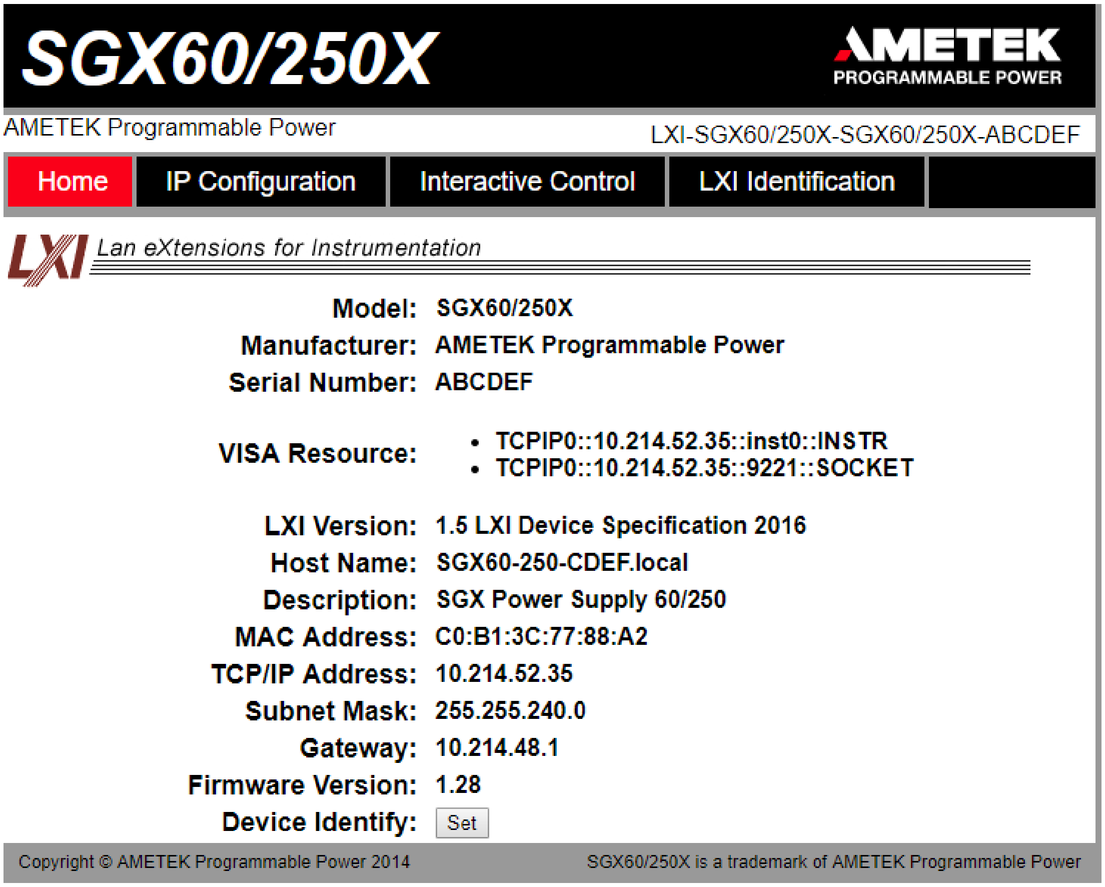


Copyright © AMETEK Programmable Power 2014 SGX is a trademark of AMETEK Programmable Power

Figure 6-8. Login Window

6.6.1 Home

This is the default, information-only page (see Figure 6-9). It displays all of the current information about the supply that you are connected to:



AMETEK Programmable Power
LXI-SGX60/250X-SGX60/250X-ABCDEF

Model: SGX60/250X
Manufacturer: AMETEK Programmable Power
Serial Number: ABCDEF
VISA Resource:
- TCPIP0::10.214.52.35::inst0::INSTR
- TCPIP0::10.214.52.35::9221::SOCKET

LXI Version: 1.5 LXI Device Specification 2016
Host Name: SGX60-250-CDEF.local
Description: SGX Power Supply 60/250
MAC Address: C0:B1:3C:77:88:A2
TCP/IP Address: 10.214.52.35
Subnet Mask: 255.255.240.0
Gateway: 10.214.48.1
Firmware Version: 1.28
Device Identify: Set

Copyright © AMETEK Programmable Power 2014
SGX60/250X is a trademark of AMETEK Programmable Power

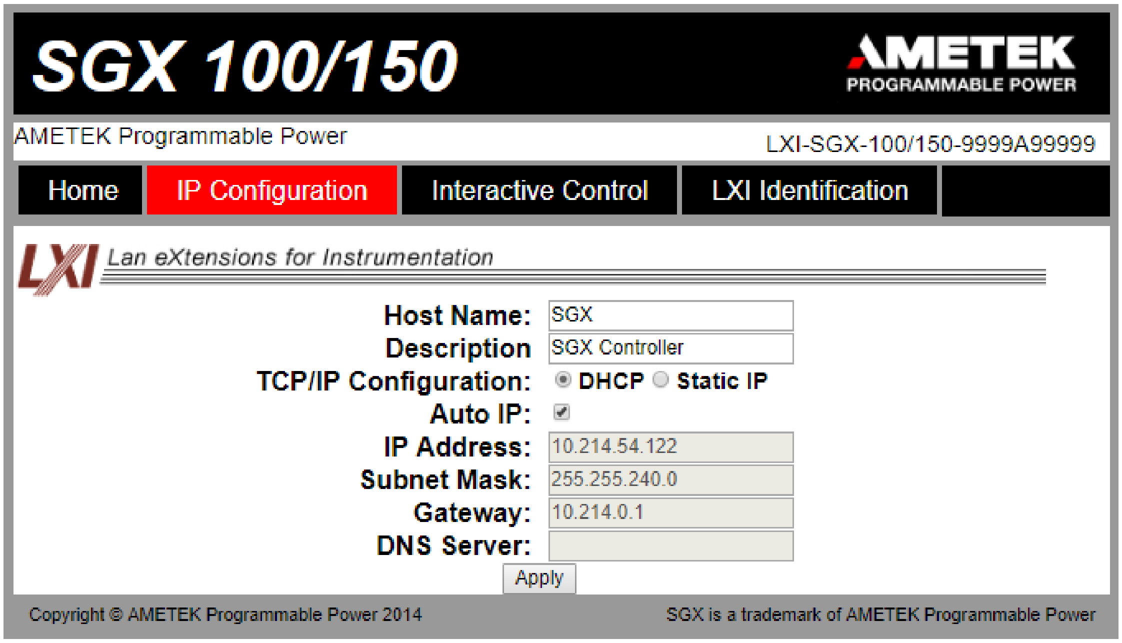
Figure 6-9. SGX Power Supply's Home Page

- The **Model** number, the **Manufacturer**, and the **Serial Number** of your SGX power supply.
- **VISA Resource** identifies the specific resource name used to communicate via VISA (Virtual Instrument Software Architecture).
- **LXI Version**: the version and instrument class of the LXI™ standard with which your power supply is compliant.
- **Host Name**: either the default or user-defined, network-unique identity (Must be limited to 15 characters or less for LXI compliance).
- **Description**: either the default or user-defined description of the power supply in use.
- **MAC Address**: the power supply Ethernet's unique hardware address.
- **TCP/IP Address**: your power supply's address actually in use at start-up; can be statically configured, DHCP acquired (default), or Auto-IP assigned (see description for IP CONFIGURATION page).
- **Subnet Mask**: network segment your power supply is on.

- **Gateway**: IP address through which the instrument communicates with systems that are not on the local subnet.
- **Firmware Version**: the version of the firmware that is currently installed.
- **Device Identify:** Blinks the LXI LED (green) on the front panel when the Set button is pressed (Set button changes to Unset button). Turns off the LXI LED when the Unset button is pressed.

6.6.2 IP Configuration

To access this web page, users need to login using the User ID and Password (For default values, refer to Section 6.2.1). You are only required to complete the information for the parameters that you wish to change, all previously entered and saved information remains by default (see Figure 6-10).



Host Name: SGX
Description SGX Controller
TCP/IP Configuration: ● DHCP ○ Static IP
Auto IP: ☑
IP Address: 10.214.54.122
Subnet Mask: 255.255.240.0
Gateway: 10.214.0.1
DNS Server:

Copyright © AMETEK Programmable Power 2014 SGX is a trademark of AMETEK Programmable Power

Figure 6-10. SGX IP Configuration Page

Host Name: the default host name is SGX. You may change this name as long as it is unique (Host Name must be limited to 15 characters for LXI compliance) so that VXI-11 Discovery and any other IP Discovery program can identify your specific device on your network.

To change: Type the new name (15 characters maximum) in the blank field provided and click **Apply** to update (or make all desired changes before clicking **Apply**).

Description: you may change the default factory setting to something more meaningful to your current setup.

To change: Type your customized description, up to 36 characters, in the blank field provided, and click **Apply** to update (or make all desired changes before clicking **Apply**).

TCP/IP Configuration: the power supply can operate in DHCP or Static IP Configuration.

You may statically assign an IP address as well as configure other Ethernet/LAN parameters (Subnet Mask and Gateway) or use DHCP for automatic assignment of an IP address.

Static IP Configuration: Click the radio button next to **Static IP** to manually configure some or all of the following the Ethernet/LAN parameters:

IP Address – input any standard IP address. (Factory setting is 192.168.0.200). Click **Apply** and enter the new IP address in LXI web browser to view the Home page of the power supply. See Figure 6-9. If you have changed the network portion of the IP address, it may be necessary to alter the network settings of your attached computer to reconnect to the power supply.

Subnet Mask – input a value that identifies which network segment your power supply is on, consisting of 4 whole numbers, each ranging from 0 through 255, separated by periods. (Factory setting is 255.0.0.0, a class-C network subnet mask). Click **Apply** to update (or make all desired changes before clicking **Apply**).

Gateway – input the IP Address of any gateway that stands between the instrument and any other network entities that communicate with the power supply. (No factory setting). Click **Apply** to update (or make all desired changes before clicking **Apply**).

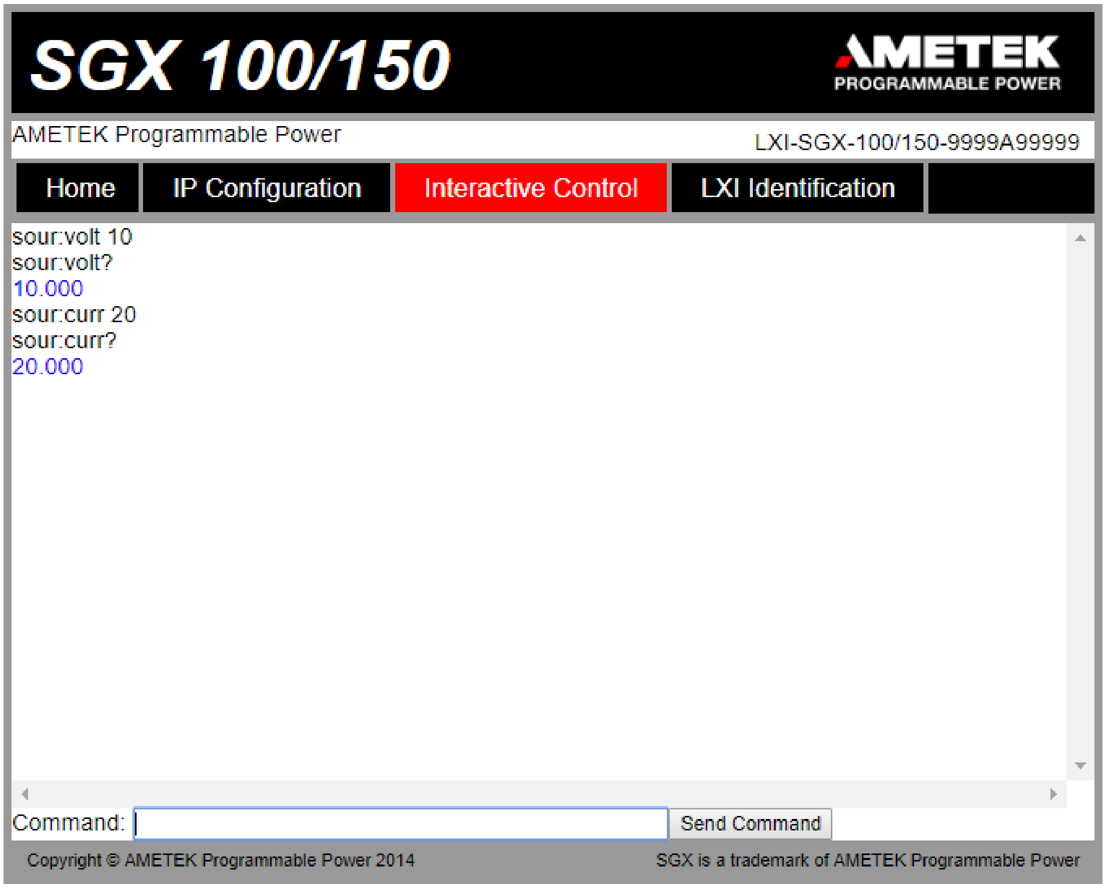
DHCP Configuration: Click the radio button next to **DHCP,** for dynamic address acquisition from the DHCP server.

Auto IP: If it is enabled, when there is no DHCP server available, the power supply will assign itself an IP address in the range from 169.254.0.1 to 169.254.255.254 with a subnet mask of 255.255.0.0.

Click in the box next to Auto IP to check (enable Auto IP); click again to uncheck (disable Auto IP) (see Figure 6-10).

6.6.3 Interactive Control

This web page allows to input a properly formatted SCPI command (refer to Section 8). Click on Send Command button to send the command to the SGX power Supply. The commands and response to the query command can be seen on the web page (see Figure 6-11).

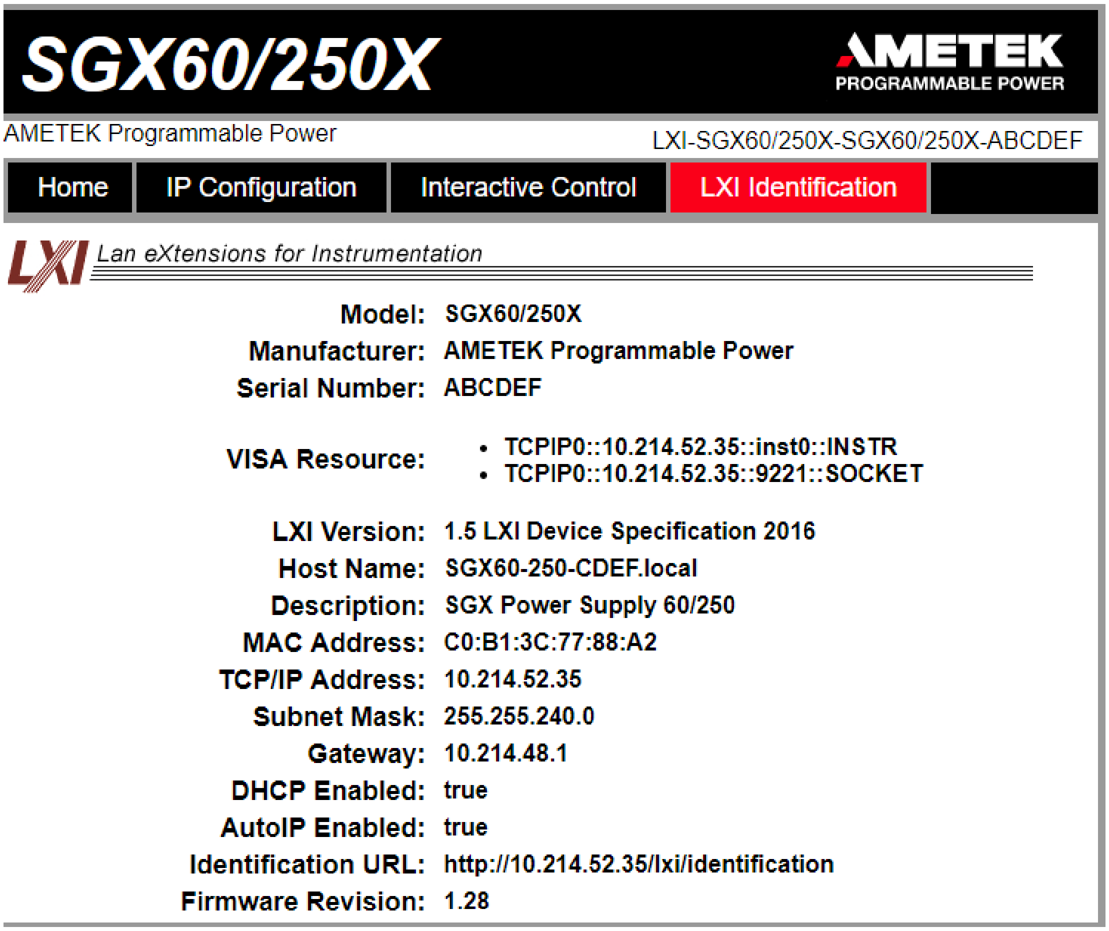

SGX 100/150
AMETEK PROGRAMMABLE POWER
AMETEK Programmable Power
LXI-SGX-100/150-9999A99999
Home
IP Configuration
Interactive Control
LXI Identification
Command:
Send Command
Copyright © AMETEK Programmable Power 2014
SGX is a trademark of AMETEK Programmable Power

```
sour:volt 10
sour:volt?
10.000
sour:curr 20
sour:curr?
20.000
```

Figure 6-11. Interactive Control Page

6.6.4 LXI Identification

The LXI Identification web page displays the LXI parameters of the SGX Power Supply (see Figure 6-12).



Model:	SGX60/250X
Manufacturer:	AMETEK Programmable Power
Serial Number:	ABCDEF
VISA Resource:	• TCPIP0::10.214.52.35::inst0::INSTR • TCPIP0::10.214.52.35::9221::SOCKET
LXI Version:	1.5 LXI Device Specification 2016
Host Name:	SGX60-250-CDEF.local
Description:	SGX Power Supply 60/250
MAC Address:	C0:B1:3C:77:88:A2
TCP/IP Address:	10.214.52.35
Subnet Mask:	255.255.240.0
Gateway:	10.214.48.1
DHCP Enabled:	true
AutoIP Enabled:	true
Identification URL:	http://10.214.52.35/lxi/identification
Firmware Revision:	1.28

Figure 6-12. LXI Identification Page

7 EXTERNAL USER CONTROL SIGNAL CONNECTOR

7.1 INTRODUCTION

This section provides illustrations of the SGX power supply's rear panel layout, which differs among the SGX models. For example, see Figure 7-1. Regardless of the layout, the component functions are common across all models and those that are pertinent to the External User Control Signal Connector is described here. The external user control signal functionality can be exercised from either of the communication interfaces (RS232, Ethernet and IEEE 488.2 GPIB).

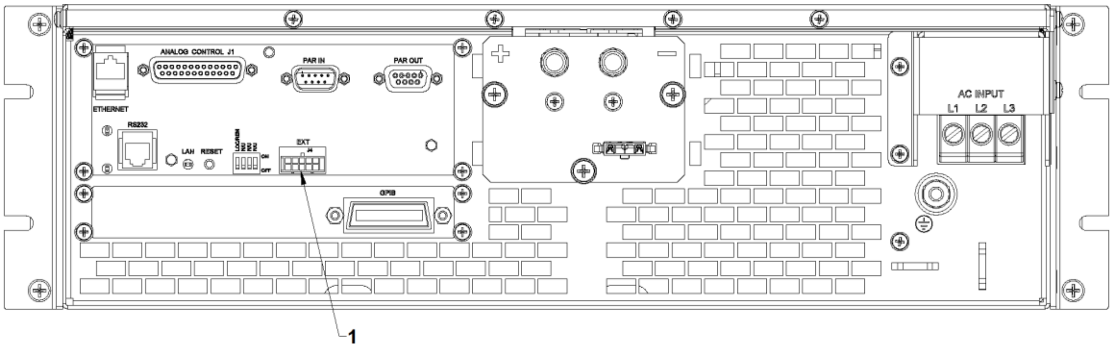


Figure 7-1. Rear Panel - External User Control Signal Connector

1 – External User Control Signal Connector

7.2 EXTERNAL USER CONTROL SIGNAL CONNECTOR DETAILS

A10-pin Molex connector (see Figure 7-2) located at the rear panel provides external auxiliary control signals to increase the user's operating control of the supply. The mating receptacle is Molex 43025-1000 with 10 female terminals. The Molex terminals accommodate wire sizes from #20 - #24.

The relay outputs, when active, connect the POLARITY, ISOLATION and SENSE pins (Pins 6, 7 and 8) of the connector to the relay COMMON pin (Pin 5). The relays are rated at 120VAC/125VDC @ 1A. Any change in output (voltage, current, etc.) initiated by the user from the RS232, GPIB, or Ethernet interface, will generate a 10ms synchronization pulse at the rear panel User Control Signal Connector of the unit (TRIGGER OUT).

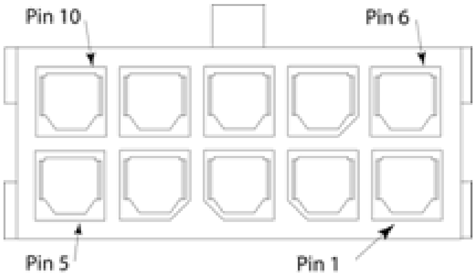


Figure 7-2. External User Connector Pinout (10-pin Molex, rear panel view)

Table 3. External User Control Signal Connector Pinout – Ethernet only

Pin	Signal Name	Functional Description	Electrical Characteristics
1	FOLDBACK	Output signal, active-low; asserted when in foldback mode; open-collector of opto-isolator transistor; emitter is connected to Pin-9. (See OUTP:PROT:FOLD command in the Output SCPI Command Subsystem, refer to Section 8.7 of this manual).	60 VDC, max., 4 mA DC, max.
2	SHUTDOWN	Input signal, TTL active-high; immediate shutdown when signal is pulled high; open-anode of opto-isolator diode with internal 1kΩ series resistor; cathode is connected to Pin-9.	12 VDC, max., - 5 VDC, max. reverse voltage
3	FAULT	Output signal, active-low; asserted when a fault is recorded in the fault register; open-collector of opto-isolator transistor; emitter is connected to Pin-9.	60 VDC, max., 4 mA DC, max.
4	TRIGGER OUT	Output signal, active-low; synchronization pulse for 10 ms when a change in the output occurs; open-collector of opto-isolator transistor; emitter is connected to Pin-9.	60 VDC, max., 7 mA DC, max.
5	COMMON	Return for all relay contacts. Could be optionally connected to Pin-9, externally.	Isolated from Pin-9
6	POLARITY	Output signal, asserted (internal relay contacts close to Pin-5, COMMON) when negative output polarity is programmed (e.g. OUTPut:POLarity INV) to program negative voltage (e.g., SOURce:VOLTage -5.0)	2 ADC, max., 30 VDC, max.
7	ISOLATION	Output signal, asserted (internal relay contacts close to Pin-5, COMMON) when the output relay is programmed ON (e.g., OUTPut:ISOlation ON).	2 ADC, max., 30 VDC, max.
8	SENSE	Output signal, asserted (internal relay contacts close to Pin-5, COMMON) when the sense relay is programmed ON (e.g., OUTput:SENse ON).	2 ADC, max., 30 VDC, max.
9	ISO COMMON	Return for all opto-isolator signals. Could be optionally connected to Pin-5, externally.	Isolated from Pin-5
10	TRIGGER IN	Input signal, TTL active-high; provides external hardware triggering of sequence functions and of voltage and current ramp functions; open-anode of opto-isolator diode with internal 1kΩ series resistor; cathode is connected to Pin-9.	12 VDC, max., - 5 VDC, max. reverse voltage

7.3 WIRING DIAGRAM

Relay control signals (POLARITY, REMOTE SENSE and ISOLATION) from the External Control Interface Connector can be used to wire external relays as shown in Figure 7-3. SCPI commands to control the external relays are provided in Table 4.

External user interface output signals (TRIGGER OUT, FAULT and FOLD BACK) should be used with the pull-up resistors as shown in Figure 7-3.

External user interface input signal TRIGGER IN is used for providing hardware trigger inputs for sequencing and ramp functions. SHUTDOWN input signal provides immediate shutdown of the power supply. Maximum electrical ratings for each signal is provided in Table 3.

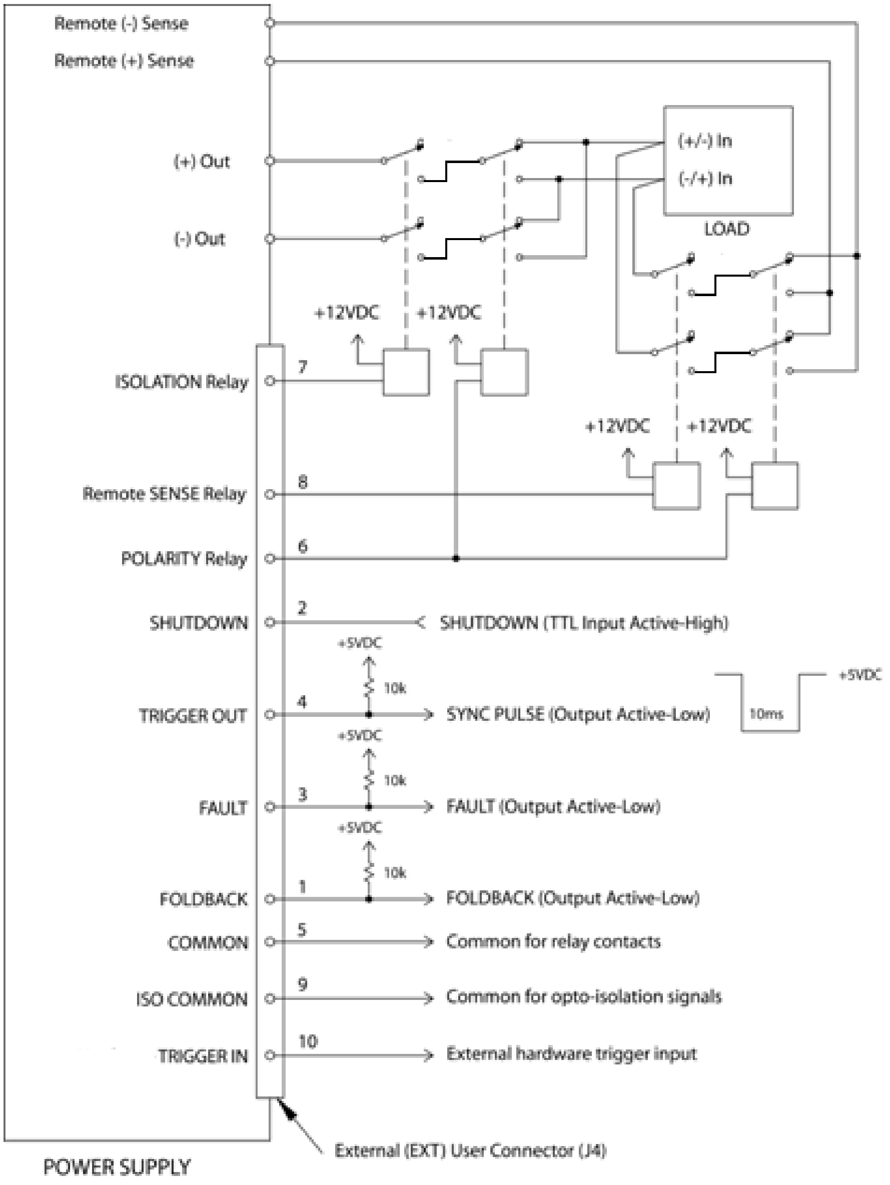


Figure 7-3. Example of Open-Collector, TTL Input, and Relay Output Circuits

Table 4. Example of Internal Relay Contact using SCPI Command

SCPI Command	Internal Relay Contact State	Pin (return to Pin-5)
OUTP:ISOL 0	ISOLATION relay = open	7
OUTP:ISOL 1	ISOLATION relay = closed	
OUTP:SENSE 0	REMOTE SENSE relay = open	8
OUTP:SENSE 1	REMOTE SENSE relay = closed	
OUTP:POL 0	POLARITY relay = open	6
OUTP:POL 1	POLARITY relay = closed	

⚠ **CAUTION**

External relays must not be hot-switched; ensure that the voltage across the relay contacts and the current through them is zero prior to changing the relay states.

This page intentionally left blank.

8
RS232/ETHERNET/ IEEE 488.2 GPIB AND SCPI COMMAND OPERATION

8.1 INTRODUCTION

This section describes the operation of the Digital Interfaces GPIB, RS232 and Ethernet by using SCPI Command sets. The command set comprises of programming, query and status commands that facilitate remote control of the power supply.

8.2 REGISTER DEFINITIONS

The SGX Power Supply supports the IEEE 488.2 GPIB, RS232, Ethernet and SCPI status reporting data structures. These structures are comprised of status registers and status register enable mask pairs. The following sections describe these pairs.

8.2.1 Protection Condition and Protection Event Status Register

These two registers have the same bit meanings, but they differ in function.

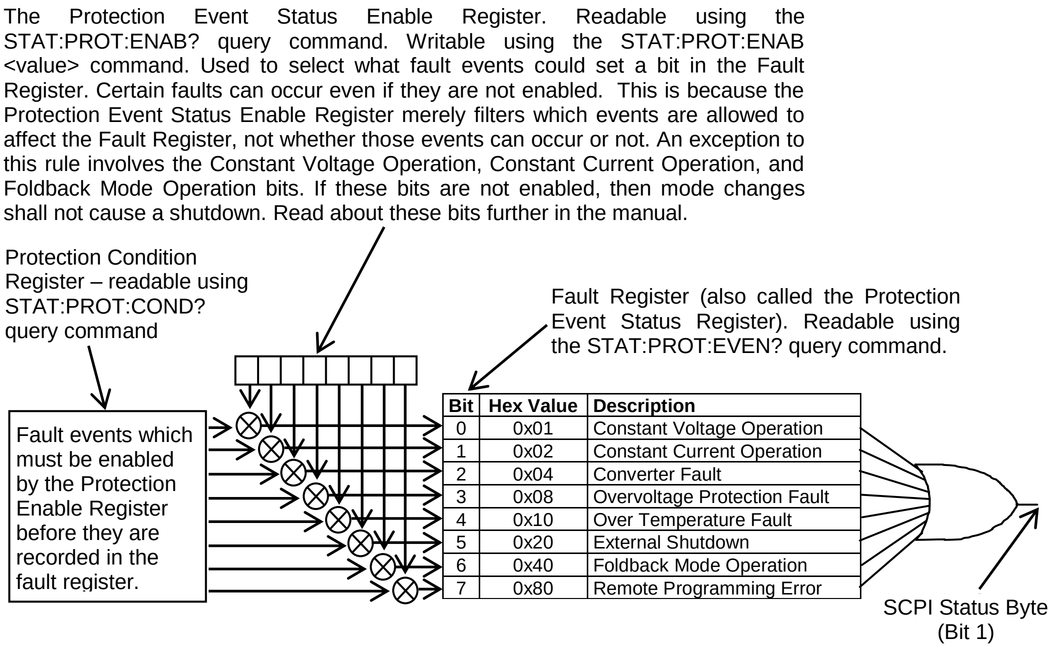
Read the **Protection Condition Register** by issuing the STAT:PROT:COND? command. This command gives the *present* status *condition* of the power hardware, so the data is not latched. It is meant to be used as a polling register.

Read the **Protection Event Status Register** by issuing the STATus:PROTection:EVENt? command. Refer Table 5 for the Protection Event Status Register details. Reading this register clears the Protection Event Status Register. Or clear the Protection Event Status Register by issuing a *CLS command or a *RST command. Bits in the Protection Event Status Register will be set only when the corresponding bit in the Protection Event Status Enable Register is set and the corresponding event occurs. The status is then latched and will remain in that state until it is read or cleared due to some command action. (Set the Enable Register with the STATus:PROTection:ENABle <mask> command, and query the Enable Register with the STATus:PROTection:ENABle? query).

To configure the Power Supply to generate service requests based on the Protection Event Status Register, program both the Protection Event Status Enable Register and the Service Request Enable Register (*SRE). See Figure 8-1 and Section 10 SCPI Status Implementation for further information.

Table 5. Protection Condition and Event Status Registers

Bit	Hex Value	Description
0	0x01	Constant voltage operation
1	0x02	Constant current operation
2	0x04	Not used
3	0x08	Overvoltage protection tripped
4	0x10	Overtemperature protection tripped
5	0x20	Supply external shutdown active
6	0x40	Foldback mode operation
7	0x80	Remote programming error



The Protection Event Status Enable Register. Readable using the STAT:PROT:ENAB? query command. Writable using the STAT:PROT:ENAB <value> command. Used to select what fault events could set a bit in the Fault Register. Certain faults can occur even if they are not enabled. This is because the Protection Event Status Enable Register merely filters which events are allowed to affect the Fault Register, not whether those events can occur or not. An exception to this rule involves the Constant Voltage Operation, Constant Current Operation, and Foldback Mode Operation bits. If these bits are not enabled, then mode changes shall not cause a shutdown. Read about these bits further in the manual.

Protection Condition Register – readable using STAT:PROT:COND? query command

Fault Register (also called the Protection Event Status Register). Readable using the STAT:PROT:EVEN? query command.

Fault events which must be enabled by the Protection Enable Register before they are recorded in the fault register.

Bit	Hex Value	Description
0	0x01	Constant Voltage Operation
1	0x02	Constant Current Operation
2	0x04	Converter Fault
3	0x08	Overvoltage Protection Fault
4	0x10	Over Temperature Fault
5	0x20	External Shutdown
6	0x40	Foldback Mode Operation
7	0x80	Remote Programming Error

SCPI Status Byte (Bit 1)

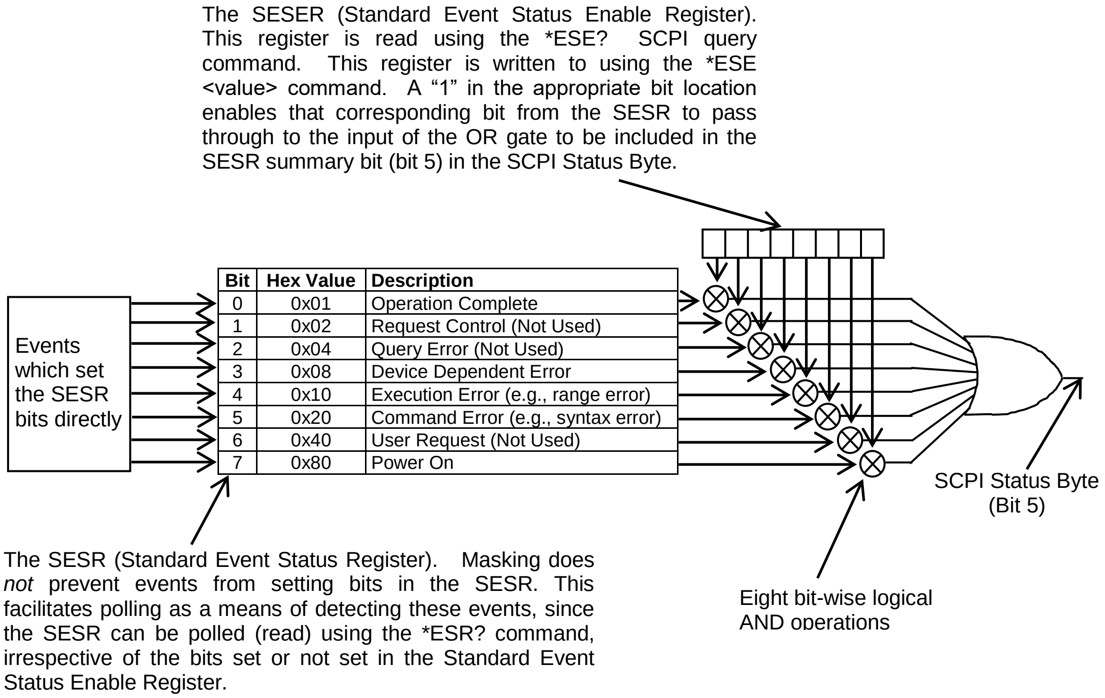
Figure 8-1. Protection Event Status and Protection Event Status Enable Register

8.2.2 Standard Event Status Register (ESR)

Read the Standard Event Status Register (ESR) by issuing the *ESR? command. See Table 6 for the Standard Event Status Register bit details. Reading this register or issuing a *CLS command will clear the ESR. Use the *ESE (Standard Event Status Enable Register) to enable corresponding ESR bits to be summarized in the summary bit of the SCPI Status byte. To configure the Power Supply to generate service requests based on the ESR, both the Standard Event Status Enable Register and the Service Request Enable Register must be programmed. See Figure 8-2 and Section 10 SCPI Status Implementation for further information.

Table 6. Standard Event Status Register

Bit	Hex Value	Description
0	0x01	Operation Complete
1	0x02	Request Control - not used
2	0x04	Query Error
3	0x08	Device Dependent Error
4	0x10	Execution Error (e.g., range error)
5	0x20	Command Error (e.g., syntax error)
6	0x40	User Request - not used
7	0x80	Power On



The SESER (Standard Event Status Enable Register). This register is read using the *ESE? SCPI query command. This register is written to using the *ESE <value> command. A “1” in the appropriate bit location enables that corresponding bit from the SESR to pass through to the input of the OR gate to be included in the SESR summary bit (bit 5) in the SCPI Status Byte.

Bit	Hex Value	Description
0	0x01	Operation Complete
1	0x02	Request Control (Not Used)
2	0x04	Query Error (Not Used)
3	0x08	Device Dependent Error
4	0x10	Execution Error (e.g., range error)
5	0x20	Command Error (e.g., syntax error)
6	0x40	User Request (Not Used)
7	0x80	Power On

Events which set the SESR bits directly
SCPI Status Byte (Bit 5)
Eight bit-wise logical AND operations

The SESR (Standard Event Status Register).   Masking does *not* prevent events from setting bits in the SESR. This facilitates polling as a means of detecting these events, since the SESR can be polled (read) using the *ESR? command, irrespective of the bits set or not set in the Standard Event Status Enable Register.

Figure 8-2. Standard Event Status and Standard Event Status Enable Register

8.2.3 SCPI Status Byte

The SCPI Status Byte registers the status of the instrument, in one of seven bits described in **Table 7**. Read the SCPI Status Byte status register by issuing either the *STB? command or a serial poll. Clear the Status Byte status register by issuing the *CLS command.

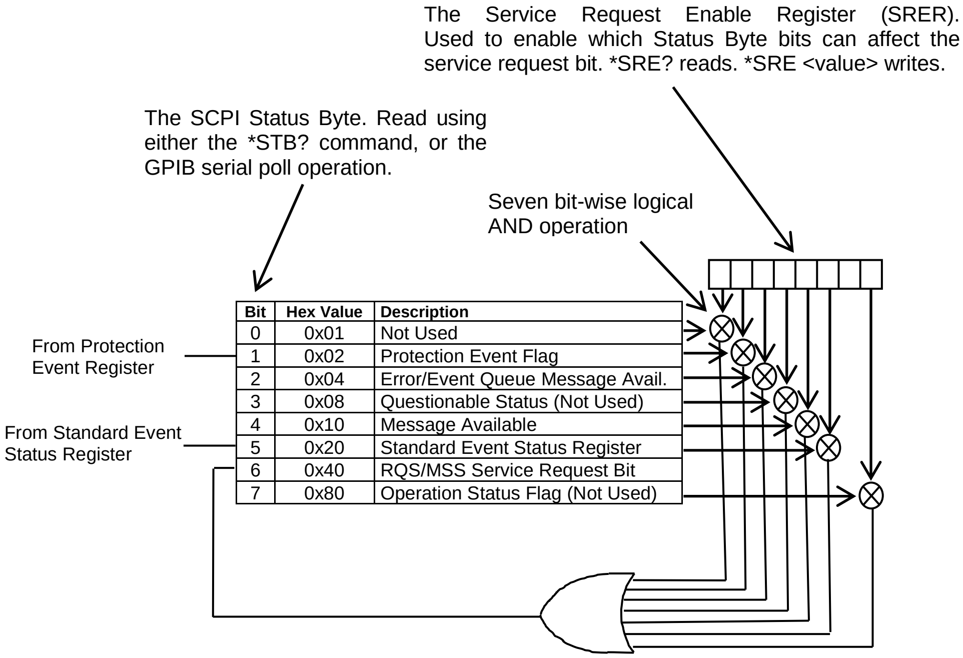
NOTE: Serial poll is applicable only to GPIB interface.

The Power Supply can be configured to request service from any of the communication interfaces, by setting the appropriate bits in the Service Request Enable Register (SRE), which has the same bit pattern as the status byte. Service Request Enable Register (SRE) can be modified by issuing the *SRE <mask> command. Service Request Enable Register (SRE) can be read by issuing *SRE? query command, See **Figure 8-3**.

For example, if the SRE register is set to 0x02 (Protection Event Flag), when the Power Supply has a fault event, Status Byte register will contain 0x42 (RQS and Protection Event Flag) and the SRQ (SRQ is supported only on GPIB) line will be asserted to indicate a request for service. See Figure 8-3 and Section 10 SCPI Status Implementation for further information.

Table 7. SCPI Status Byte

Bit	Hex Value	Description
0	0x01	Not used.
1	0x02	Protection Event Status flag. Indicates the selected protection event occurred.
2	0x04	Error/event queue message available. Set when any error/event is entered in the System Error Queue. It is read using the SYSTem:ERRor? query.
3	0x08	Questionable Status flag. Indicates the quality of the current data being acquired. This bit is not used.
4	0x10	Message available (MAV). Indicates a message is available to read (Only applicable to GPIB Interface).
5	0x20	Standard Event Status Register (ESR). Summary bit for the ESR. Set when any of the ESR bits are set and cleared when the ESR is read.
6	0x40	Request Service flag (RQS) for serial polling or Master Summary Status (MSS) in response to *STB? If service requests are enabled (with the *SRE command), this bit represents the RQS and will be sent in response to a serial poll, then cleared. The MSS bit indicates that the device has at least one reason to request service. Even though the device sends the MSS bit in response to a status query (*STB?), it is not sent in response to a serial poll. It is not considered part of the IEEE-488.1 Status Byte.
7	0x80	Operation Status flag. Indicates the current operational state of the unit. This bit is not used.



The Service Request Enable Register (SRER). Used to enable which Status Byte bits can affect the service request bit. *SRE? reads. *SRE <value> writes.

The SCPI Status Byte. Read using either the *STB? command, or the GPIB serial poll operation.

Seven bit-wise logical AND operation

Bit	Hex Value	Description
0	0x01	Not Used
1	0x02	Protection Event Flag
2	0x04	Error/Event Queue Message Avail.
3	0x08	Questionable Status (Not Used)
4	0x10	Message Available
5	0x20	Standard Event Status Register
6	0x40	RQS/MSS Service Request Bit
7	0x80	Operation Status Flag (Not Used)

From Protection Event Register
From Standard Event Status Register

Figure 8-3. SCPI Status Byte and Service Request Enable Register

8.2.4 Error/Event Queue

The SGX Power Supply maintains an Error/Event Queue as defined by SCPI. The queue holds up to 10 error events. It is queried using the SYSTem:ERRor? command which reads in a First In/First Out (FIFO) manner. The read operation removes the entry from the queue. The *CLS command will clear all entries from the queue.

The following error codes are defined in the SCPI 1995.0 specification and are supported by the SGX Power Supply. Error codes are in the range of [-32768, 32767]. SCPI reserves the negative error codes and 0, while error codes greater than 0 are device specific errors.

Table 8 SCPI Error Codes

Error Code	Description
206	**No channels setup to trigger** This means that an attempt was made to trigger the DIA using the TRIG:TYPE <1|2|3> command when there are no armed trigger settings. This error is *not* generated when the GET is received, even when there are no armed trigger settings.
0	**No error** The error queue is empty.
-102	**Syntax error** An unrecognized command or data type was encountered.

Error Code	Description
-151	**Invalid string data** Incorrect password. Manufacturer, model, or serial number string was more than 16 characters. Invalid mnemonic.
-161	**Invalid block data** The expected number of data values was not received.
-200	**Execution error** An error/event number in the range [-299, -200] indicates that an error has been detected by the instruments execution control block. The occurrence of any error in this class shall cause the execution error bit (bit 4) in the Event Status Register to be set. An execution error can be the result of: • A <program data> element out of range, such as programming 35 volts in a 33-volt device. • A command could not be executed due to the current condition of the device.
-203	**Command protected** Attempted to store calibration values to EEPROM without unlocking.
-221	**Settings conflict** Attempted to set output greater than soft limits or to set soft limits less than output.
-222	**Data out of range** Parameter exceeded range of valid values.
-241	**Hardware missing** A legal command or query could not be executed because of a hardware fault.
-340	**Calibration failed** Error during calculation of calibration values occurred.
-350	**Queue overflow** The error queue can contain up to 10 entries. If more than 10 error/event conditions are logged before the SYSTem:ERRor? query, an overflow will occur; the last queue entry will be overwritten with error -350. When the queue overflows, the least recent error/events remain in the queue and the most recent error/events are discarded.
-360	**Communication error** Communications to a channel was disrupted.

8.2.5 Serial Poll Operation

Performing a serial poll will not modify the SCPI Status Byte other than to clear the RQS (bit 6) for a SGX Power Supply requesting service. Queries affecting the Status Registers and subsequent serial poll are described below:

- `*ESR?` clears the ESR and bit 5 of the SCPI Status Register
- `SYSTem:ERRor?` clears bit 2 of the SCPI Status Register if the queue is empty

8.3 SCPI CONFORMANCE INFORMATION

The syntax of all SCPI commands implemented by the SGX power supplies and documented in this manual, are either SCPI confirmed or they are customized commands not part of the SCPI definition. None of the commands implemented by the SGX power supplies are classified as SCPI approved commands (approved by the SCPI Consortium but not contained in the SCPI version to which the SGX power supplies conform).

To document whether the syntax of each command is SCPI compliant or not, this manual provides a column, labeled "SCPI", in each command reference table. A "C" in the "SCPI" column means that the command syntax is SCPI compliant; an "N" in the "SCPI" column means that the command syntax is not part of the SCPI definition.

8.3.1 Parameter Definitions

The following table describes the format of the command arguments, when applicable.

PARAMETER DEFINITIONS	
Type	**Valid Arguments**
<boolean>	"ON" or 1. "OFF" or 0.
<NR1>	The data format <NR1> is defined in IEEE 488.2 for integers. Zero, positive and negative integer numeric values are valid data.
<NRf>	The data format <NRf> is defined in IEEE 488.2 for flexible Numeric Representation. Zero, positive and negative floating point numeric values are some examples of valid data.
<string>	Characters enclosed by single or double quotes.

8.3.2 Units

The SGX power supplies will accept the following units as suffixes to numeric values:

UNITS	
Type of Unit	**Valid Suffix**
Voltage	"VOLTS" or "volts", "V" or "v", "MV" or "mv" or "mV"
Current	"AMPS" or "amps", "A" or "a", "MA" or "ma" or "mA"
Time	"SEC" or "sec", "S" or "s", "MS" or "ms", "MIN" or "min"
Frequency	"HZ" or "hz"

The default units are VOLTS, AMPS, SEC, and HZ.
For example, "`SOUR:VOLT 1`" programs 1 volt.
To program in units of millivolts, type "`SOUR:VOLT 1mV`".

8.3.3 Conventions

SCPI uses the conventions where optional commands and parameters are enclosed by "[ ]". Additionally the shorthand version of a command is indicated by capital letters.

For example,

```
SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude] 120.0
```

can be written as

```
SOURce:VOLTage 120.0
```

or

```
SOUR:VOLT 120.0
```

8.3.4 Queries

The query syntax is identical to the command syntax with a "?" appended. For example, to query the programmed voltage, send the string: `SOURce:VOLTage?`. A subsequent device read will return a value such as "33.000". All queries are terminated with a carriage return and line feed (0x0D 0x0A). When the power supply has nothing to report, its output buffer will contain two ASCII characters: a carriage return and linefeed (in decimal the values are: <13><10>).

8.4 IEEE 488.2 COMMON COMMAND SUBSYSTEM

The following commands are common to all SCPI instruments and declared mandatory by IEEE 488.2. In the following table, the power supply is defined as the "device" on the GPIB bus.

Command	Description
*CLS	Clears all status reporting data structures including the Status Byte, Standard Event Status Register, and Error Queue. The STAT:PROT:ENAB (protection event enable register) is cleared by this command; other enable registers are not cleared by this command.
*ESE <0+NR1>	Sets the value of the Standard Event Status Enable Register that determines which bits can be set in the Standard Event Status Register. See section 8.2.2 for valid values.
*ESE?	Returns the integer value of the Standard Event Status Enable Register. See section 8.2.2 for valid values. **Response:** <0+NR1>
*ESR?	Returns the integer value of the Standard Event Status Register. The ESR and the Status Byte ESR bit are cleared. See section 8.2.2 for valid values. **Response:** <0+NR1>
*IDN?	Returns the device identification as an ASCII string. **Response:** <Manufacturer>, <model>, <serial number>, <DCI firmware version> <AI firmware version> **Example:** AMETEK, SGX100/150C-1AAA, 0622A00111,1.00,1.00
*OPC	Enables the Operation Complete bit of the Standard Event Status Register to be set when all pending operations are complete. See section 8.2.2.
*RCL <integer>	SGX - Specifies the preset storage location (0 through 9) from which to load into the supply's Voltage, Current, Overvoltage Protection, and Output State settings. Preset location 0 stores the supply's power on default.
*RST	Resets the supply to its Power ON (PON) state. Clears all status reporting data structures including the Status Byte, Standard Event Status Register, and Error Queue. The STAT:PROT:ENAB (protection event enable register) is cleared by this command; other enable registers are not cleared by this command.
*SAV <integer>	Specifies the preset storage location (0 through 9) to store the supply's existing Voltage, Current, Overvoltage Protection, and Output State settings. Preset location 0 stores the supply's power up default.
*SRE <0+NR1>	Sets the value of the Service Request Enable Register, which determines which bits in the Status Byte will cause a service request from the device. See section on Status Byte for valid values.
*SRE?	Returns the integer value of the Service Request Enable Register. See section on Status Byte for valid values. Values range from 0-63 or 128-191. **Response:** <0+NR1>
*STB?	Returns the integer value of the Status Byte with bit 6 representing the Master Summary Status (MSS) instead of RQS. The MSS bit acts as a summary bit for the Status Byte and indicates whether the device has at least one reason to request service based on the MAV and the ESR bits. See section on Status Byte for valid values. Values range from 0-255. **Response:** <0+NR1>
*WAI	Sets the device to wait until all previous commands and queries are complete before executing commands following the *WAI command.

8.5 SOURCE SCPI COMMAND SUBSYSTEM

This section first presents a tree summary of the SOURce commands and then provides a tabular description.

8.5.1 SOURCE SCPI Command Summary

```
SOURce
     :CURRent
     :CURRent?
          [:LEVel]
          [:LEVel?]
               [:IMMediate]
               [:IMMediate?]
                    [:AMPLitude] <NRf>
                    [:AMPLitude?]
          :LIMit
          :LIMit?
               [:AMPLitude] <NRf>
               [:AMPLitude?]
          :RAMP <NRf> <NRf>
          :RAMP?
               :ABORt
               :ALL?
               :HTRIGgered <NRf>
               :HTRIGgered?
               :TRIGgered <NRf> <NRf>
               :TRIGgered?
          :TRIGgered
          :TRIGgered?
               :CLEar
               :AMPLitude <NRf>
               :AMPLitude?
     :POWer
     :POWer?
          :LEVel
          :LEVel?
               :IMMediate
               :IMMediate?
                    :AMPLitude <NRf>
                    :AMPLItude?
     :TIMeout?
     :VOLTage
     :VOLTage?
          [:LEVel]
          [:LEVel?]
               [:IMMediate]
               [:IMMediate?]
                    [:AMPLitude] <NRf>
                    [:AMPLitude?]
          :LIMit
          :LIMit?
               [:AMPLitude] <NRf>
```

```
      [:AMPLitude?]
:PROTection
:PROTection?
      [:LEVel] <NRf>
      [:LEVel?]
      :TRIPped?
      :STATe?
      :CLEar
:RAMP <NRf> <NRf>
:RAMP?
      :ABORt
      :ALL?
      :HTRIGgered <NRf> <NRf>
      :HTRIGgered?
      :TRIGgered <NRf> <NRf>
      :TRIGgered?
:TRIGgered
:TRIGgered?
      :CLEar
      :AMPLitude <NRf>
      :AMPLitude?
```

8.5.2 SOURCE SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
SOURce	Source subsystem.	C
:CURRent	Sets the output current in amps (default) or in milliamps.	C
:CURRent?	Returns the output current in amps or in milliamps	C
[:LEVel]	Sets the output current in amps (default) or in milliamps.	C
[:LEVel?]	Returns the output current in amps or in milliamps.	C
[:IMMediate]	Sets the output current in amps (default) or in milliamps.	C
[:IMMediate?]	Returns the output current in amps or in milliamps	C
[:AMPLitude] <NRf>	Sets the output current in amps (default) or in milliamps.	C
[:AMPLitude?]	Returns the output current in amps or in milliamps	C
:LIMit	Sets an upper soft limit on the programmed output current for the supply. The soft limit prevents the supply from being inadvertently programmed above the soft limit, thus providing a method for protecting the load against damaging currents.	C
:LIMit?	Returns the upper soft limit on the programmed output current for the supply.	C
[:AMPLitude] <NRf>	Sets an upper soft limit on the programmed output current fo the supply.	C
[:AMPLitude?]	Returns the upper soft limit on the programmed outpu current for the supply.	C

Command	Description	SCPI
:RAMP <NRf> <NRf>	Sets the output current to ramp from the present value to the specified value (first argument) in the specified time (second argument). See Ramp Function description below.	N
:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
:ABORt	Aborts ramping and clears trigger mode.	N
:ALL?	Returns the ramping status of all channels.	N
:HTRIGgered <NRf>	Sets the value of the output current to ramp to be implemented when the hardware trigger is received.	N
:HTRIGgered?	Returns the value of the output current to ramp to be implemented when the hardware trigger is received.	N
:TRIGgered <NRf> <NRf>	Sets the output current to ramp from the present value to the specified value (first argument) in the specified time (second argument) upon the trigger command. See Ramp description below.	C
:TRIGgered?	Returns the value that the output current is to ramp to (first value) and the time that it is to ramp (second value) upon the trigger command.	C
:TRIGgered	Sets the output voltage to the values stored by the SOURce: CURRent:TRIGger:AMPLitude command.	C
:TRIGgered?	Returns the current level that will be set upon receipt of the trigger.	C
:CLEar	Clears the value stored by the SOURce:CURRent:TRIGger:AMPLitude command.	C
:AMPLitude <NRf>	Stores the value of the output current to be set when the SOURce: CURRent:TRIGGered command is sent.	C
:AMPLitude?	Returns the stored value of the output current to be set when the SOURce: CURRent:TRIGGered command is sent.	C
:POWer	see Section 8.13.2.	C
:POWer?		C
:TIMeout?	Returns the integer value 1 (timeout since last query) or 0 (no timeout) of the timeout status of the channel.	N
:VOLTage	Sets the output voltage of the supply in volts (default) or in millivolts.	C
:VOLTage?	Returns the output voltage of the supply in volts or in millivolts.	C
[:LEVel]	Sets the output voltage of the supply in volts (default) or in millivolts.	C
[:LEVel?]	Returns the output voltage of the supply in volts or in millivolts.	C

Command	Description	SCPI
[:IMMediate]	Sets the output voltage of the supply in volts (default) or in millivolts.	C
[:IMMediate?]	Returns the output voltage of the supply in volts or in millivolts.	C
[:AMPLitude] <NRf>	Sets the output voltage of the supply in amps (default) or in milliamps.	C
[:AMPLitude?]	Returns the output voltage of the supply in amps or in milliamps.	C
:LIMit	Sets the upper soft limit on the programmed output voltage. The soft limit prevents the supply from being inadvertently programmed above the soft limit, thus providing a method for protecting the load against damaging voltages.	C
:LIMit?	Returns the upper soft limit set on the programmed output voltage.	C
[:AMPLitude] <NRf>	Sets the upper soft limit on the programmed output voltage.	C
[:AMPLitude?]	Returns the upper soft limit on the programmed output voltage.	C
:PROTection	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C
:PROTection?	Returns the set overvoltage protection trip point in volts (default) or in millivolts.	C
[:LEVel] <NRf>	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C
[:LEVel?]	Returns the set overvoltage protection trip point in volts or in millivolts.	C
:TRIPped?	Returns 1 (TRIPPED) or 0 (UNTRIPPED) state of the overvoltage protection circuit.	C
:STATe?	Returns the state 1 (ON) or 0 (OFF) If the overvoltage protection is enabled.	C
:CLEar	Clears the overvoltage protection circuit.	C
:RAMP <NRf> <NRf>	Sets the output voltage to ramp from the present value to the specified value (first argument) in the specified time (second argument). See Ramp Function description Section 8.5.3.	N
:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
:ABORt	Aborts ramping and clears trigger mode.	N
:ALL?	Returns the ramping status of all channels.	N
:HTRIGgered <NRf>	Sets the value of the output voltage ramp to be implemented when the hardware trigger is received.	N
:HTRIGgered?	Returns the value of the output voltage ramp to be implemented when the hardware trigger is received.	N

Command	Description	SCPI
:TRIGgered <NRf> <NRf>	Sets the output voltage to ramp from the present value to the specified value (first argument) in the specified time (second argument) upon the trigger command. See description of the Ramp Function below.	N
:TRIGgered?	Returns the output voltage to ramp	N
:TRIGgered	Sets the output voltage to the values stored by the SOURce:VOLTage:TRIGger:AMPLitude command.	C
:TRIGgered?	Returns the voltage level that will be set upon receipt of the trigger.	C
:CLEar	Clears the value stored by the SOURce:VOLTage:TRIGger:AMPLitude command.	C
:AMPLitude <NRf>	Stores the value of the output current to be set when the SOURce:VOLTage:TRIGGered command is sent.	C
:AMPLitude?	Returns the stored value of the output current to be set when the SOURce:VOLTage:TRIGGered command is sent.	C

8.5.3 RAMP FUNCTION

The ramp function allows the user to transition from one voltage or current to another linearly in a specified time period (100 ms - 99 sec with 100 ms programming resolution). A unit may ramp only voltage or current, not both at a given time.

For example, `SOUR:VOLT:RAMP:TRIG 1 1` followed by `SOUR:CURR:RAMP:TRIG 2 2` will cause the unit to ramp only the output current to 2 amps in 2 seconds upon the `TRIG:RAMP` command.

8.5.3.1 VOLTAGE RAMPING TO A HIGHER VOLTAGE

Requires a programmed current of at least 20% of the full scale value. Settings less than 20% will significantly lengthen the ramp time due to charging of the large capacitance in the output section of the power supply.

8.5.3.2 VOLTAGE RAMPING TO A LOWER VOLTAGE

Requires an appropriate resistive load. The discharge rate of the large capacitance in the output section of the power supply, plus other user capacitance, significantly lengthens the ramp time.

8.5.3.3 CURRENT RAMPING

Requires an appropriate resistive load.

8.6 MEASURE SCPI COMMAND SUBSYSTEM

This section first presents a tree summary of the MEASure commands and then provides a tabular description.

8.6.1 MEASURE SCPI Command Summary

```
MEASure
      :CURRent
      :CURRent?
            :AVErage <NR1>
            :AVErage?
      :POWer? (see Section 8.13.)
      :VOLTage
      :VOLTage?
            :AVErage <NR1>
            :AVErage?
```

8.6.2 MEASURE SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
MEASure	Measure subsystem.	C
:CURRent?	Returns the floating point value of the DC output current in amps.	C
:CURRent	Measure Current subsystem.	N
:AVErage <NR1>	Enter a value of 3 to 9 to set the number of readings to average together when returning the current value from the MEAS:CURR? command to reduce noise in the readback readings. The value of 3 (factory default) provides the fastest response time in the readings, but less rejection of noise.	N
:AVErage?	Returns the number 3 to 9 to indicate the number of readings to average together when taking a current reading.	N
:POWer?	see Section 8.13	
:VOLTage?	Returns the floating point value of the DC output voltage in volts.	C
:VOLTage	Measure Voltage subsystem.	N
:AVErage <NR1>	Enter a value of 1 to 5 to set the number of readings to average together when returning the voltage value from the MEAS:VOLT? command. This function reduces noise in the readback readings. The value of 1 (factory default) provides the fastest response time in the readings, but less rejection of noise.	N
:AVErage?	Returns the number 1 to 5 to indicate the last set number of readings to average together when taking a voltage reading.	N

8.7 OUTPUT SCPI COMMAND SUBSYSTEM

This section first presents a tree summary of the OUTPut commands and then provides a tabular description.

8.7.1 OUTPUT SCPI Command Summary

```
OUTPut
      :ISOLation <boolean>
      :ISOLation?
      :POLarity <string>
      :POLarity?
      :PROTection
            :DELay <NRf>
            :DELay?
            :FOLD <0|1|2>
            :FOLD?
      :SENSe <boolean>
      :SENSe?
      :STATe <boolean>
      :STATe?
      :TRIPped?
```

8.7.2 OUTPUT SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
OUTPut	Output subsystem.	C
:ISOLation <boolean>	Sets the rear panel isolation relay control signal ON or OFF. Valid arguments are 1/ON or 0/OFF.	N
:ISOLation?	Returns the state of the rear panel isolation relay control signal: 1 = ON 0 = OFF	N
:POLarity <NORM/0/OFF|INV/1/ON>	Changes the state of the polarity relay. This command requires that the isolation relay be open beforehand. If the isolation relay is closed when this command is attempted, the state of the polarity relay will not change, and an error message will be generated.	C
:POLarity?	Returns the state of the polarity relay: <NORM/0/OFF|INV/1/ON>	C
:PROTection	Output Protection subsystem.	N
:DELay <NRf>	Sets the programmable time delay executed by the supply before reporting output protection conditions after a new output voltage or current is specified. Functional granularity of +/- 0.5 seconds	N
:DELay?	Returns the time delay to be executed by the supply.	N

Command	Description	SCPI
:FOLD <0|1|2>	Sets the foldback (program down) mode of the supply. Valid arguments are 0 (OFF or do nothing, do not program down to zero), 1 (program down to zero upon entering constant-voltage mode), or 2 (program down to zero upon entering constant-current mode).	N
:FOLD?	Returns the set foldback (program down) mode of the supply. 0 = OFF; will not program down. 1 = will program down to zero upon entering constant-voltage mode. 2 = will program down to zero upon entering constant current mode	N
:SENSe <boolean>	Sets the sense relay signal open or close. Valid arguments are 1/ON or 0/OFF.	N
:SENSe?	Returns the setting of the sense relay signal: 1 = ON 0 = OFF	N
:STATe <boolean>	Sets the output to zero or the programmed value; opens or closes the isolation relay. Valid arguments are 1/ON or 0/OFF. *RST state value is ON.	C
:STATe?	Returns the state of the output: 1 = ON 0 = OFF	C
:TRIPped?	Returns the integer value 1 (TRIPPED) or 0 (UNTRIPPED) state of the output.	N

8.8 STATUS SCPI COMMAND SUBSYSTEM

This section first presents a tree summary of the STATus commands and then provides a tabular description.

Note: *See Section 5 for further information.*

8.8.1 STATUS SCPI Command Summary

```
STATus
      :OPERation
            :CONDition?
            :ENABle <NR1>
            :ENABle?
            :EVENt?
      :PRESet
      :PROTection
            :CONDition?
            :ENABle <NR1>
            :ENABle?
            :EVENt?
            :SELEct <NR1>
            :SELEct?
      :QUEStionable
            :CONDition?
            :ENABle <NR1>
            :ENABle?
            :EVENt?
```

8.8.2 STATUS SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
`STATus`	Status subsystem.	C
`:OPERation`	Status Operation subsystem.	C
`:CONDition?`	Returns the integer value of the Operation Condition Register. The query is supported but will always return “0” indicating operational condition.	C
`:ENABle <NR1>`	Sets the enable mask of the Operation Event Register allowing true conditions to be reported in the summary bit of the Operation Condition Register. Values are written and queried but have no effect on the Operation Condition Register.	C
`:ENABle?`	Returns the value of the current mask if the Operation Event Register.	C
`:EVENt?`	Returns the integer value of the Operation Event Register. This query is supported but always returns a value of “0” indicating operational condition.	C
`:PRESet`	Sets the enable mask of the Operation Event Register and the Questionable Event Register to all 1’s.	C
`:PROTection`	Status Protection subsystem.	C
`:CONDition?`	Returns the integer value of the Protection Condition Register. Used to read the status of the power hardware. See section 8.2.1 for a detailed table of the various bits that make up this register.	C
`:ENABle <NR1>`	Sets the enable mask of the Protection Event Register, which allows true conditions to be reported in the summary bit of the Protection Condition Register.	C
`:ENABle?`	Returns the value of the current mask of the Protection Event Register.	C
`:EVENt?`	Returns the integer value of the Protection Event Register.	C
`:SELEct <NR1>`	This command provides a means for selecting which fault bits from the protection event register (also called the fault register and can be read using the `STAT:PROT:EVEN?` command) are able to set the protection event flag bit in the SCPI status byte (readable using the `*STB?` command). It defaults to value 255 at power-on time, and never changes unless intentionally programmed to a new value.	N
`:SELEct?`	Returns the last selection value programmed.	N
`:QUEStionable`	Status Questionable subsystem.	C
`:CONDition?`	Returns the integer value of the Questionable Condition Register. The query is supported but will always return “0” indicating operational condition.	C
`:ENABle <NR1>`	Sets the enable mask of the Questionable Event Register allowing true conditions to be reported in the summary bit of the Questionable Condition Register. Values are written and queried but have no effect on the Questionable Condition Register.	C
`:ENABle?`	Returns the value of the current mask of the Questionable Event Register.	C
`:EVENt?`	Returns the integer value of the Questionable Event Register. This query is supported but always returns a value of “0”, indicating operational condition.	C

8.9 SYSTEM SCPI COMMAND SUBSYSTEM

This section first presents a tree summary of the SYSTem commands and then provides a tabular description.

8.9.1 SYSTEM SCPI Command Summary

```
SYSTem
      :ERRor?
      :FAULt?
      :LOCAL <boolean>
      :LOCAL?
      :NET
            :AUTOIP <boolean>
            :AUTOIP?
            :DESC <string>
            :DESC?
            :DHCPMODE <boolean>
            :DHCPMODE?
            :DNS <string>
            :DNS?
            :GATE <string>
            :GATE?
            :HOST <string>
            :HOST?
            :IP <string>
            :IP?
            :LANLED <boolean>
            :LANLED?
            :MAC?
            :MASK <string>
            :MASK?
            :NETBUTTON <string>
            :PORT <NRf>
            :PORT?
            :TERM <NRf>
            :TERM?
      :VERsion?
```

8.9.2 SYSTEM SCPI Command Reference

The letter "C" in the "SCPI" column means that the command syntax is SCPI compliant; an "N" in the "SCPI" column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
`SYSTem`	System subsystem.	
`:ERRor?`	Queries Error Queue for next error/event entry (first in, first out). Entries contain an error number and descriptive text. A 0 return value indicates no error occurred; negative numbers are reserved by SCPI. The maximum return string length is 255 characters. The queue holds up to 10 error/entries. All entries are cleared by the *CLS command.	C
`:FAULt?`	Returns four numeric values separated by commas for the four system fault registers. See System Fault Registers (Table 8). Valid response is 128, 0, 0, 0 or 0, 0, 0, 0. Response: <Fault1–8>, <Fault9–16>, <Fault17–24>, <Fault25–31>	N
`:LOCAL <boolean>`	Forces the supply to local or remote state. <ON> or <1> sets operation to local mode. <OFF> or <0> sets the operation to remote mode. There are two noteworthy circumstances where this command may prove necessary. The first case involves using RS232 to communicate with the supply, since the normal GPIB mechanisms for transition between local and remote and back again do not exist when using RS232. The other case is when the REMOTE/LOCAL switch S1-1 is placed in the ON position—thereby disabling the GPIB mechanism for transition from remote to local.	N
`:LOCAL?`	Returns ON or 1 if in local mode. Returns OFF or 0 if in remote mode.	N
`SYST:NET`	Network device	N
`:AUTOIP <boolean>`	Sets the network Auto IP mode in the Primary configuration without affecting the Secondary configuration.. 0 = disable AutoIP; 1 = enable AutoIP	N
`:AUTOIP?`	Returns 1 if AutoIP is enabled in the Primary configuration. Returns 0 if AutoIP is disabled in the Primary configuration.	N
`:DESC <string>`	Set the network Description, a 36 character alphanumeric string	N
`:DESC?`	Returns the network Description.	N
`:DHCPMODE <boolean>`	Sets the network DHCP Mode in the Primary configuration without affecting the Secondary configuration. 0 = disable DHCP; 1 = enable DHCP	N
`:DHCPMODE?`	Returns 1 if DHCP Mode is enabled in the Primary configuration. Returns 0 if DHCP mode is disabled in the Primary configuration.	N
`:DNS <string>`	Sets the network DNS IP address for the device. String is in the format "NNN.NNN.NNN.NNN" where "NNN" = 0 through 255, inclusive.	N
`:DNS?`	Returns the network DNS address for the device.	N
`:GATE <string>`	Sets the network gateway IP address for the device. String is in the format "NNN.NNN.NNN.NNN" where "NNN" = 0 through 255, inclusive.	N
`:GATE?`	Returns the network gateway IP address for the device.	N

Command	Description	SCPI
:HOST <string>	Set the network Host Name, a 15-character (maximum) alphanumeric string. (Must be limited to 15 characters for LXI compliance)	N
:HOST?	Returns the network Host Name	N
:IP <string>	Sets the Primary configuration to STATICIP mode and sets the network IP address for the device. String is in the format "NNN.NNN.NNN.NNN" where "NNN" = 0 through 255, inclusive.	N
:IP?	Returns two IP addresses: the first is the IP address set to be used when the system boots up; the second is the IP address presently in use by the power supply. (The first address will either be 0.0.0.0. if the Primary configuration is DHCP or DHCP+AUTOIP, or it will be the static IP last specified).	N
:LANLED <boolean>	1 causes LANLED to blink. (Used to identify a unit in a rack). 0 causes LANLED to stop blinking.	N
:LANLED?	Returns blink state of the LAN LED: 0 – not blinking; 1 – blinking.	N
:MAC?	Returns the network MAC address. xx:xx:xx:xx:xx:xx (Hexadecimal digit pairs)	N
:MASK <string>	Set the network Subnet Mask for the device. String is in the format "NNN.NNN.NNN.NNN" where "NNN" = 0 through 255, inclusive.	N
:MASK?	Returns the network Subnet Mask for the device.	N
:NETBUTTON <string>	Returns configuration parameters to factory default. (Software equivalent of pressing the Reset switch on the rear panel of the power supply). You must cycle the power to effect the change. The access string is "6867."	N
:PORT <NRf>	Set the network TCP/IP socket listening port. Valid values are 1025 to 65535.	N
:PORT?	Returns the network TCP/IP socket listening port.	N
:TERM <NRf>	Sets the incoming string termination character to be used by the device. Factory set to 3. The valid range is 1-4. Values indicate the following terminator(s): 1 = 0x0d only (CR), 2 = 0x0a only (LF), 3 = 0x0d 0x0a (CR LF), 4 = 0x0a 0x0d (LF CR)	N
:TERM?	Returns the string terminators to be used by the device.	N
:VERsion?	Returns a numeric value corresponding to the SCPI version number for which the instrument complies. The response is in the format YYYY.V where the Y's represent the year and V represents the approved version number for that year (e.g., 1995.0)	C

Table 9. System Fault Registers

Bit Position	Bit Weight	Fault1–8	Fault9–16	Fault17–24	Fault25–31
7	128	Channel 1	not used	not used	not used
6	64	not used	not used	not used	not used
5	32	not used	not used	not used	not used
4	16	not used	not used	not used	not used
3	8	not used	not used	not used	not used
2	4	not used	not used	not used	not used
1	2	not used	not used	not used	not used
0	1	not used	not used	not used	not used

The `SYStem:FAULt?` query returns 4 numeric values separated by commas. Each value is the decimal equivalent of the total bit weights for that System Fault Register as described in the table above.

8.10 HTRIGGER SCPI COMMAND SUBSYSTEM

This section applies only to units with the External User Interface (Currently only available on Ethernet versions).

The HTRIGGER function allows the user to apply an External User Interface input signal to initiate a sequence or a voltage or current ramp. Once a hardware trigger is run, Arm goes to 0 (not armed); however, the last loaded sequence remains in memory.

8.10.1 HTRIGGER SCPI Command Summary

```
HTRIGger
      :ABORt
      :RAMP
      :SEQuence  (see Section 8.13.)
      :SEQuence?  (see Section 8.13.)
      :ARM  (see Section 8.13.)
      :ARM? (see Section 8.13.)
```

8.10.2 HTRIGGER SCPI Command Reference

Command	Description	SCPI
`HTRIGger`	Hardware trigger subsystem.	N
`:ABORt`	Stops the execution of a currently running hardware trigger function. In addition: For Ramp: Clears all settings of voltage and current. For Sequence: Sets the Arm function to 0 (not armed).	N

`:RAMP`	Executes voltage or current ramping function previously programmed by the SOURce command, i.e.,
	`SOURce:VOLTage:RAMP`
	`SOURce:CURRent:RAMP`
		N

`:SEQuence <string>`	Loads a Sequence to be initialized by the external Hardware Trigger. This command must be followed by the HTRIG:ARM command before the Hardware Trigger becomes operational.	N
`:SEQuence?`	Returns the currently loaded sequence name	N
`:ARM <boolean>`	Readies the last loaded sequence to run when the external Hardware Trigger signal is issued. 1 = Ready to run sequence when Hardware trigger is engaged. 0 = Not Armed, Hardware trigger is ignored NOTE: To arm a different sequence, issue the HTRIG:SEQ command first.	N
`:ARM?`	Returns the ready status of the last loaded sequence. 1 = Ready to run sequence when Hardware trigger is engaged. 0 = Not Armed, Hardware trigger is ignored NOTE: To load a new sequence to be armed you must issue the HTRIG:SEQ command first.	N

8.11 TRIGGER SCPI COMMAND SUBSYSTEM

This section describes the programming soft trigger function.

8.11.1 TRIGGER SCPI Command Summary

```
TRIGger
      :ABORt
      :RAMP

      :TYPE <1|2|3>
```

8.11.2 TRIGGER SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
`TRIGger`	Trigger subsystem.	C
`:ABORt`	Stops the execution of a currently running trigger function, and clears all settings of voltage and current.	N

`:RAMP`	Executes voltage or current ramping function previously programmed by the SOURce command, i.e.,
	`SOURce:VOLTage:RAMP`
	`SOURce:CURRent:RAMP`
		N

`:TYPe<1|2|3>`	Executes voltage and current values previously programmed by the `SOURce command i.e.,`
	`SOURce:VOLTage:LEVel:TRIGger`
	`SOURce:CURRent:LEVel:TRIGger`
	Valid arguments are 1 (Voltage), 2 (Current), or 3 (Both).
		N

8.12 CALIBRATION SCPI COMMAND SUBSYSTEM

Note: *See section 9 for calibration procedures.*

CAUTION

Please refer to the power supply manual for further information before performing calibration procedures. Calibration must be performed by qualified personnel who appropriately deal with attendant hazards. If calibration is not performed properly, functional problems could arise, requiring that the supply be returned to the factory.

8.12.1 CALIBRATION SCPI Command Summary

```
CALibrate
     :DATA <NRf><NRf><NRf><NRf><NRf><NRf><NRf><NRf><NRf><NRf>
     :INITial
          :CURRent <NRf>
          :CURRent?
          :MEASure:CURRent:AVErage <NR1>
          :MEASure:CURRent:AVErage?
          :VOLTage <NRf>
          :VOLTage?
               [:AMPLitude] <NRf>
               [:AMPLitude?]
               :PROTection <NRf>
               :PROTection?
     :MEASure
          :CURRent
               :ADC?
               :CALCulate
               :GAIN <NRf>
               :GAIN?
               :OFFSet <NRf>
               :OFFSet?
               :POINt <1|2> <NRf>
          :VOLTage
               :ADC?
               :CALCulate
               :GAIN <NRf>
               :GAIN?
               :OFFSet <NRf>
               :OFFSet?
               :POINt <1|2> <NRf>
     :MODel
          :LASTCALDATE
          :LASTCALDATE?
          :NEXTCALDATE
          :NEXTCALDATE?
          :POWERON
          :POWERON?
          :RESET
          :RESET?
          :SAVELAST <0|1>
          :SAVELAST?
          :PONS
          :PONS?
     :OUTPut
          :CURRent
               :CALCulate
               :DAC <NR1>
               :FIVEPoint <1|2|3|4|5>
               :FIVEPoint?
               :GAIN <NRf>
               :GAIN?
               :OFFSet <NRf>
               :OFFSet?
               :POINt <1|2> <NRf>
          :VOLTage
               :CALCulate
               :DAC <NR1>
               :FIVEPoint <1|2|3|4|5>
               :FIVEPoint?
               :GAIN <NRf>
               :GAIN?
               :OFFSet <NRf>
               :OFFSet?
               :POINt <1|2> <NRf>
               :PROTection
                    :CALCulate
                    :DAC <NR1>
                    :GAIN <NRf>
                    :GAIN?
                    :OFFSet <NRf>
                    :OFFSet?
     :UNLock <string>
     :STORe
     :LOCK
```

8.12.2 CALIBRATION SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
CALibrate	Calibration subsystem.	C
:DATA <NRf><NRf><NRf> <NRf><NRf><NRf><NRf><NRf> <NRf><NRf>	Sets the values of the ten floating point calibration constants: 1) output voltage DAC gain 2) output voltage DAC offset 3) output current DAC gain 4) output current DAC offset 5) output voltage protection DAC gain 6) output voltage protection DAC offset 7) voltage measurement ADC gain 8) voltage measurement ADC offset 9) current measurement ADC gain 10) current measurement ADC offset Values are separated by space or comma.	N
:INITial	Used to store next power up values.	N
:CURRent <NRf>	Sets the power-on default value of current.	N
:CURRent?	Returns the default value of power-on current.	N
:VOLTage <NRf>	Sets the power-on default voltage.	N
:MEASure:CURRent: AVErage <NR1>	Sets the number of readings to average together when returning the current value with the MEAS:CURR? command to reduce noise in the readback readings. Enter a value of 3 to 9, with the value of 3 (factory default) providing the fastest response time in the readings, but less rejection of noise.	N
:MEASure:CURRent: AVErage?	Returns the number 3 to 9 to indicate the number of readings to average together when taking a current reading.	N
:VOLTage?	Returns the default value of power-on voltage	N
[:AMPLitude] <NRf>	Sets the power-on default voltage.	N
[:AMPLitude]?	Returns the power-on default voltage.	N
:PROTection <NRf>	Sets the power-on default value of the overvoltage protection.	N
:PROTection?	Returns the default value of the power-on overvoltage protection.	N
:MEASure	Calibrate Measure subsystem	N
:CURRent	Calibrate Measure Current subsystem	N
:ADC?	Returns the integer value of the A/D for the current measurement.	N
:CALCulate	Calculates the value of the gain and offset for current measurements.	N
:GAIN <NRf>	Sets the value of the gain for current measurements.	N
:GAIN?	Returns the value of the gain for current measurements.	N
:OFFSet <NRf>	Sets the value of the offset for current measurements.	N
:OFFSet?	Returns the value of the offset for current measurements.	N

Command	Description	SCPI
:POINt <1|2> <NRf>	Sets the current measurement calibration point (1 or 2). The actual output current is measured with an external meter.	N
:VOLTage	Calibrate Measure Voltage subsystem	N
:ADC?	Returns the integer value of the A/D for the voltage measurement.	N
:CALCulate	Calculates the value of the gain and offset for voltage measurements.	N
:GAIN <NRf>	Sets the value of the gain for voltage measurements.	N
:GAIN?	Returns the value of the gain for voltage measurements.	N
:OFFSet <NRf>	Sets the value of the offset for voltage measurements.	N
:OFFSet?	Returns the value of the offset for the voltage measurements.	N
:POINt <1|2> <NRf>	Sets the voltage measurement calibration point (1 or 2). The actual output voltage is measured with an external meter.	N
:MODel	Calibrate Model subsystem	N
:LASTCALDATE<NRf>	Sets the date last calibrated; format: MM DD YYYY (space after MM and DD required)	N
:LASTCALDATE?	Returns the date last calibrated.	N
:NEXTCALDATE<NRf>	Sets the date next calibration is required; format: MM DD YYYY (space after MM and DD required)	N
:NEXTCALDATE?	Returns the date next calibration is required.	N
:POWERON <string>	Sets the default output enable condition at power on. Input format: “ON,INIT” enables the output at next power on “OFF,INIT” disables the output at next power on NOTE: Quotation marks are required in the command string.	N
:POWERON?	Returns the status of the output enable condition at power on	N
:RESET <string>	Sets the default output enable condition when the *RST command is issued. Input format: “ON,INIT” enables the default output “OFF,INIT” disables the default output NOTE: Quotation marks are required in the command string.	N
:RESET?	Returns the status of the default output enable condition when the *RST command is issued	N
:SAVELAST <0|1>	Sets the SAVELAST condition “0” disables SAVELAST feature “1” enables SAVELAST feature	N
:SAVELAST?	Returns the status of the Save Last condition	N
:PONS <NRf>	Sets the GPIB Power ON service request Input format: “0” disables the GPIB Power ON service request “1” enables the GPIB Power ON service request	N
:PONS?	Returns the status of GPIB Power ON service request	N

Command	Description	SCPI
:OUTPut	Calibrate Output subsystem	N
:CURRent	Calibrate Output Current subsystem	N
:CALCulate	Calculates the value of the gain and offset for output current.	N
:DAC <NR1>	Sets the output of the output current D/A converter.	N
:FIVEPOINT <1|2|3|4|5>	Sets output current value for each calibration point (1-5)	N
:FIVEPOINT?	Returns the entered values for 5-point calibration.	N
:GAIN <NRf>	Sets the value of the gain for the output current.	N
:GAIN?	Returns the value of the gain for the output current.	N
:OFFSet <NRf>	Sets the value of the offset for the output current.	N
:OFFSet?	Returns the value of the offset for the output current.	N
:POINt <1|2> <NRf>	Sets the current output calibration point (1 or 2). The actual output current is measured with an external meter.	N
:VOLTage	Calibrate Output Voltage subsystem	N
:CALCulate	Calculates the value of the gain and offset for output voltage.	N
:DAC <NR1>	Sets the output of the output voltage D/A converter.	N
:FIVEPOINT <1|2|3|4|5>	Sets output voltage value for each calibration point (1-5)	N
:FIVEPOINT?	Returns the entered values for 5-point calibration.	N
:GAIN <NRf>	Sets the value of the gain for the output voltage.	N
:GAIN?	Returns the value of the gain for the output voltage.	N
:OFFSet <NRf>	Sets the value of the offset for the output voltage.	N
:OFFSet?	Returns the value of the offset for the output voltage.	N
:POINt <1|2> <NRf>	Sets the voltage output calibration point (1 or 2). The actual output voltage is measured with an external meter.	N
:PROTection	Calibrate Output Voltage Protection subsystem	N
:CALCulate	Calculates the value of the gain and offset for output overvoltage protection. This takes more than 30 seconds to complete. Use `*ESE 1` and a serial poll to detect the completed operation.	N
:DAC <NR1>	Sets the output of the output overvoltage protection D/A converter.	N
:GAIN <NRf>	Sets the value of the gain for the output overvoltage protection.	N
:GAIN?	Returns the value of the gain for the output overvoltage protection.	N
:OFFSet <NRf>	Sets the value of the offset for the output overvoltage protection.	N
:OFFSet?	Returns the value of the offset for the output overvoltage protection.	N
:UNLock <string>	Sets the non-volatile memory available to store calibration constants. The access string is “6867”.	
:STORe	Stores the calibration constants in non-volatile memory.	N
:LOCK	Disables access to the non-volatile memory. Prevents attempts to store calibration values. (Issue after CAL:UNLock and CAL:STORe commands)	N

8.13 SGX-UNIQUE COMMANDS

This section deals with the SGX power supply's unique operation commands. These commands are for programming sequences and power mode operations.

8.13.1 Restrictions on Sequence Programming:

Do NOT use non-sequence-related commands while performing sequences. Do NOT use sequence commands as stand-alone commands outside a sequence.

In order to allow maximum flexibility for generating small incremental changes during a test sequence, the SGX allows 1ms time resolution on each step. With this capability, however, it is possible to create output changes that generate large, and potentially damaging, currents in the output capacitors of the unit.

⚠ **CAUTION**

When creating test sequences, please use the following guidelines to prevent damage to the unit (see Note below):

- Estimate the AC frequency and peak-to-peak voltage (Vp-p) of the desired test sequence.
- Convert the estimated Vp-p to a % of maximum output voltage (e.g. if Vp-p is 10V and maximum voltage of the supply is 100V, then %Vp-p – 10%)
- Verify that the frequency and %Vp-p does not exceed the values below:

Frequency	**% Vp-p**
10Hz	25%
50Hz	5.0%
100Hz	2.5%
150Hz	1.67%
200Hz	1.25%

Another consideration is the actual rise and fall capabilities of the output of the supply. Although damage will not occur, the shape of the output waveform will be affected by these differences. These vary widely, depending on the load conditions; contact the factory for further information.

NOTE: Contact the factory for detailed information if the desired waveform exceeds the recommended limits as discussed.

8.13.2 SGX SOURCE SCPI Command Subsystem

The Power command and its corresponding query were added to the Source subsystem for power regulation (Section 8.5 contains the Source commands that can be used for SGX). The letter "C" in the "SCPI" column means that the command syntax is SCPI compliant; an "N" in the "SCPI" column means that the command syntax is not part of the SCPI definition.

NOTE: Must be preceded by commands to set maximum voltage, overvoltage protection and current (see Description in table below).

SOUR:POWer
:POWer?

Command	Description	SCPI
SOURce	Source subsystem.	C
:POWer <NRf>	Enter a value for power regulation. This command must be used in the following manner to avoid excess power to the load: 1) Set the output state to off using OUTP:STAT OFF 2) Program a desired voltage maximum by programming SOUR:VOLT <voltmax> 3) Program a desired OVP maximum by programming SOUR:VOLT:PROT <ovpmax> 4) Program a desired current maximum by programming SOUR:CURR <currmax> 5) Then issue the power command for the desired number of watts SOUR:POW <watts> 6) Engage the output using OUTP:STAT ON After the above actions the supply should be on and regulating at the desired power level or below. (Lower level is possible because the load may draw less than the given power setting if the voltage or current maximums programmed in the preceding steps cause such limitation). Subsequently the SOUR:POW <watts> command may be issued to adjust the power level up or down. ⚠ *Power mode is easily exited unintentionally by re-programming a voltage or current value, or issuing certain other commands that have a material influence over the power control loop. This possibility would cause power regulation to cease and then expose the load to either the voltage maximum or the current maximum or both;* *THIS COULD CAUSE EXCESS POWER BEING DELIVERED TO THE LOAD.* Consequently, when using power regulation, the best practice is to restrict SCPI command traffic to queries until such a time power mode is to be exited altogether. The suggestion for exiting power mode intentionally is to program the output voltage and output current to zero using SOUR:VOLT 0.0 and SOUR:CURR 0.0 See the PROG:EXEC POWERSETTINGS <watts> <vlimit> <ilimit> <ovp> command for an alternate method for entering all the relevant settings on a single command line.	C
:POWer?	After a SOUR:POWer programming command has been issued, this query command returns all of the settings inherent in that command. The return takes the following format: <watts>w @<vlimit>v max, <ilimit>a max, <ovp>v ovp	C

8.13.3 SGX PROGRAM SCPI Command Subsystem

The PROGRAM subsystem, available in SGX, provides a means to program sequences. Through *sequences* power supply output settings and duration for each of those individual settings, called sequence steps, can be programmed. A sequence step includes a voltage value *with* a current value *with* an over-voltage-protection value *with* a duration value. A single sequence is a program of up to 20 steps (21 steps if the RETURN or STOP or GOTO is counted), where one step consists of a setting for the power supply output. There may be up to 50 different named sequences (programs) with up to 20 steps in each. (The default name for a program is TEST[n], where [n] is a number 01 thru 50, but may be reprogrammed with a unique name up to 15 characters.)

NOTE: Do NOT use non-sequence-related commands while performing sequences. Also do NOT use sequence commands as stand-alone commands outside a sequence.

8.13.3.1 PROGRAM SCPI COMMAND SUMMARY

```
PROGram

     :INITializing?
     :CATalog?
     :[SELected]
          :ARM
          :ARM?
          :DEFine <step#>,VIMODE,<volts>,<amps>,<ovp>,<sec>
          :DEFine <step#>,RAMPTOV,<volts>,<volts>,<amps>,<ovp>,<sec>
          :DEFine <step#>,RAMPTOC,<volts>,<amps>,<amps>,<ovp>,<sec>
          :DEFine <step#>,POWERSETTINGS,<watts>,<volts>,<amps>,<ovp>,<sec>
          :DEFine <step#>,REPEAT
          :DEFine <step#>,SUBCALL, "SEQNAME"
          :DEFine <step#>,RETURN
          :DEFine <step#>,LOOP,<count>
          :DEFine <step#>,NEXT
          :DEFine <step#>,STOP
          :DEFine <step#>,GOTO, "SEQNAME"
          :DEFine <step#>,PAUSE
          :DEFine <step#>,NOP
          :DEFine? <step#>
          :DELete
               :SELected
               :ALL
          :EXECute <value1>,<value2>,<value3>,…,<valueN>
          :EXECute VIMODE <NRf> <NRf> <NRf>
          :EXECute RAMPTOV <NRf> <NRf> <NRf> <NRf> <NRf>
          :EXECute RAMPTOC <NRf> <NRf> <NRf> <NRf> <NRf>
          :EXECute POWERSETTINGS <NRf> <NRf> <NRf> <NRf>
          :MALLocate DEFAULT
          :NAME "name"
          :NAME "name",<chan#>
          :NAME?
```

```
      :SAVe
            :SELected
            :ALL
      :STATe <value>
      :STATe?
```

8.13.3.2 PROGRAM SCPI COMMAND REFERENCE

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Following this command reference table, are some annotated examples showing how to create a sequence (program) in your power supply.

Command	Description	SCPI
`PROGram`	Program subsystem	C
`:INITializing?`	This query returns 1 after mains power-on, and continues to return 1 until the program/sequence functionality is fully initialized, after which point 0 is returned. *Do not attempt to use any of the other sequence related commands--i.e., commands in the SCPI PROG command tree --while program/sequence functionality is initializing. (Typical initialization time is approx. 15 seconds after power-on.)*	N
`:CATalog?`	Lists all the defined sequence names. The format for the response to the query is a list of comma-separated test names.	C
`:[SELected]`	Selected section of program subsystem	C
`:ARM`	Prepares the presently selected sequence for execution; this command may take a few seconds depending the length of the selected sequence and all subsequences that the selected sequence may depend upon. The command to use to execute an armed sequence is `PROG:STAT RUN`.	N
`:ARM?`	A return of 1 means the sequence is armed. A return of 0 means it is not.	N
`:DEFine <step#>, VIMODE, <volts>, <amps>, <ovp>, <s>`	Programs the `VIMODE` sequence command into the selected sequence (see `PROG:SEL:NAME` command) at <step#>. The following values set by this command: voltage <volts>, current <amps>, over voltage protection <ovp>, and duration of <s> seconds. <s> may have a decimal value with a granularity of 0.001 seconds. This command is valid for steps 1 thru 20.	C
`:DEFine <step#>, RAMPTOV,<start volts>,<end volts>, <amps>, <ovp>,<s>`	Programs the `RAMPTOV` sequence command into the selected sequence at <step#>. The following values are programmed: starting and ending voltage ramp values <start volts> and <end volts>, current <amps>, over voltage protection value <ovp>,and voltage ramping duration in <s> seconds. <s> may have a decimal value with a granularity of 0.001 seconds. This command is valid for steps 1 thru 20.	C

Command	Description	SCPI
:DEFine <step#>, RAMPTOC, <volts>, <start amps>,<end amps>, <ovp>, <s>	Programs the RAMPTOC sequence command into the selected sequence at <step#>. The following values are programmed: the starting and ending current ramp values <start amps> and <end amps>, voltage <volts>, over voltage protection <ovp>,and current ramping duration in <s> seconds. <s> may have a decimal value with a granularity of 0.001 seconds. This command is valid for steps 1 thru 20.	C
:DEFine <step#>, POWERSETTINGS, <watts>, <volts>, <amps>, <ovp>, <s>	Programs the constant power POWERSETTINGS sequence command into the selected sequence at <step#>. The following values define the power setting: constant power limit <watts>,voltage limit<volts>,current limit<amps>,over voltage protection <ovp>, and time duration in <s> seconds. <s> may have a decimal value with a granularity of 0.001 seconds. This command is valid for steps 1 thru 20. If a VIMODE, or a RAMPTOV, or a RAMPTOC command follows the POWERSETTINGS command, then when the POWERSETTINGS command has completed execution, the subsequent command will take control and the constant power mode regulation shall cease. However, if the POWERSETTINGS command is immediately followed by a PAUSE command, then as long as the PAUSE is in effect the constant power mode regulation will continue.	C
:DEFine <step#>, REPEAT	Programs the REPEAT sequence command into the selected sequence at <step#>. This sequence command causes sequence execution to jump back to the starting location where sequence execution began, resume execution from there, and continue repeating endlessly. To stop, issue the STOP command. This command is valid for steps 1 thru 20. (To program a finite number of steps to repeat, see the LOOP command).	C
:DEFine <step#>, SUBCALL, "name"	Programs the SUBCALL sequence command into the selected sequence at <step#>. The SUBCALL sequence command causes sequence execution to jump to the beginning of a sub-sequence named "name". If the sub-sequence has a RETURN command at its end, then when the RETURN command is encountered, execution will resume at the step immediately following the SUBCALL. This command is valid for steps 1 thru 20.	C
:DEFine <step#>, RETURN	Programs the RETURN sequence command into the selected sequence at <step#>. The RETURN sequence command, if it occurs in a sequence that was called with a SUBCALL command, causes execution to resume at the step immediately following the SUBCALL. If the RETURN command occurs in a sequence executed as the primary sequence (i.e., not a sub-sequence), then the RETURN shall be interpreted as though it were a STOP command. The RETURN command is valid for steps 1 thru 21.	C

Command	Description	SCPI
:DEFine <step#>, LOOP,<count>	Programs the LOOP sequence command into the selected sequence at <step#>. The LOOP sequence command, together with its associated <count> value and the NEXT sequence command, provides a means of repeating a set of sequence steps for a defined number of times. All sequence steps that exist between the LOOP sequence command and the NEXT sequence command shall be executed for <count> number of times. It is recommended that the LOOP command and its corresponding NEXT command be in the same named sequence; nevertheless, they may be in different named sequences. The ability to place these two commands in different named sequences allows for the chaining of a number of named sequences together using the GOTO command, and then to put a loop around that entire chain to be repeated a number of times. The LOOP NEXT command pair does support nesting to 10 deep, and the count value must be between 0 and 65535.	C
:DEFine <step#>,NEXT	Programs the NEXT sequence command into the selected sequence at <step#>. The NEXT command must follow a matching LOOP command. The NEXT command causes sequence execution to resume at the matching LOOP command, with a count decreased by 1.	C
:DEFine <step#>,STOP	Programs the STOP sequence command into the selected sequence at <step#>. This sequence command causes sequence execution to stop while the unit remains at the state of the last command within the sequence. This command is valid for steps 1 thru 21. When the PROG:MALLOCATE DEFAULT command is used, a STOP command is automatically loaded into step 21 of that new sequence. This STOP may be overwritten to become a RETURN or GOTO command.	C
:DEFine <step#>,GOTO,”name”	Programs the GOTO sequence command into the selected sequence at <step#>. During sequence execution, the effect of this sequence command is to cause execution to transfer to the beginning of the sequence named “name”. This step is valid for steps 1 thru 21. The name must be in double quotes. See the PROG:NAME “name” command for how sequences may be given user defined names.	C
:DEFine <step#>,PAUSE	Programs the PAUSE sequence command into the selected sequence at <step#>. During sequence execution the effect of this command is to cause execution to suspend until a RESUME command is issued to resume execution. This step is valid for steps 1 thru 20.	C
:DEFine? <step#>	Queries the selected sequence for the program contents at step <step#>. The response will read back the step type and defined parameters when programmed. The resolution is defined by the step type.	C
:DELete	Program Delete subsystem	C
:SELected	Causes the presently selected sequence to be deleted from ram and non-volatile memory. Its previously allocated memory goes back into the memory pool. The memory pool is the memory from which the MALLocate command allocates memory.	C

Command	Description	SCPI
:ALL	This command causes all defined sequences to be deleted from ram and non-volatile memory.	C
:SAVe	The sub tree for the SAVe commands.	N
:SELected	Saves the presently selected sequence to non-volatile memory for preservation while the power supply is off. Up to 50 sequences a maximum of 20 steps long may be saved.	N
:ALL	Saves all defines sequences to non-volatile memory.	N
:EXECute	The EXECute commands provide a means of explicitly programming the supply to perform a single action that would normally have been done in a sequence step. This in turn provides a means of simulating a sequence. However, each step is significantly slowed by the need to parse the command defining the sequence of actions.	C
:EXECute VIMODE, <volts>, <amps>, <ovp>	Allows setting of active voltage, current, and ovp.	C
:EXECute RAMPTOV, <start volts>,<end volts >, <amps>, <ovp>,<s>	Sets the voltage ramp starting from <start volts> to <end volts> over time period <s> with Values <amps> and <ovp> being set at the beginning of the ramp.	C
:EXECute RAMPTOC, <volts>, <start amps>,<end amps>, <ovp>, <s>	Sets the current ramp starting from <start amps> and going to <end amps> over time period <s> with values <volts> and <ovp> being set at the beginning of the ramp.	C
:EXECute POWERSETTINGS, <watts>, <volts>, <amps>, <ovp>	Command directs the supply to regulate in a constant power mode to a value of <watts> , with a voltage limit of <volts>, current limit of <amps> and with voltage protection setting of <ovp>.	C
:MALLocate DEFAULT	Allocates program memory for a newly named sequence. This command has *no* effect on already existing sequences, and shall generate an error message if an attempt is made to allocate memory to an already existing sequence. After allocating memory to a newly named sequence, that sequence goes from the EMPTY state to the EDIT state. The state of a sequence may be queried by the PROG:STAT? command. In no case can the total number of sequences exceed 50.	C
:NAME "sequence name"	Performs one of two possible actions. Either selects an already existing sequence for use. (See the PROG:CAT? command for a list of saved sequences that may be selected.) Or provides a name for a new sequence. The action that is performed depends upon the "sequence name" and whether is already exists in sequence memory or not. A sequence name must not be longer than 15 characters. After naming a new sequence, the sequence is in the EMPTY state. The next required action to the sequence is to use the MALLocate command to allocate memory for the newly named sequence. After the MALLocate command is issued, the new sequence goes from the EMPTY state to the EDIT state.	C
:NAME?	Returns the name of the presently selected sequence. If no sequence is presently selected, such as occurs after a *RST command, then the default sequence will be TEST01.	C

<table>
<tr><th>Command</th><th>Description</th><th>SCPI</th></tr>
<tr><td>:STATe <state name&gt;</td><td>Provides a means to change the state of a sequence. The states that may be issued are as follows: RUN, RESUME, PAUSE, STOP, and COMPLETE.
A table showing allowable state transitions:
<table>
<tr><th>requested
present</th><th>RUN</th><th>RESUME</th><th>PAUSE</th><th>STOP</th><th>COMPL</th></tr>
<tr><td>RUNNING</td><td>*error*</td><td>*error*</td><td>PAUSED</td><td>STOPPED</td><td>*error*</td></tr>
<tr><td>PAUSED</td><td>*error*</td><td>RUNNING</td><td>PAUSED</td><td>STOPPED</td><td>*error*</td></tr>
<tr><td>STOPPED</td><td>RUNNING</td><td>*error*</td><td>STOPPED</td><td>STOPPED</td><td>*error*</td></tr>
<tr><td>EDIT</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td><td>STOPPED</td></tr>
<tr><td>EMPTY</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td></tr>
<tr><td>ERROR</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td></tr>
<tr><td>INITIALIZE</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td><td>*error*</td></tr>
</table></td><td>N</td></tr>
<tr><td>:STATe?</td><td>Returns the present state for the selected sequence. It returns a state for the RAM copy of the sequence, and a state for the slave processor copy. (Internally, the SGX uses a two processor architecture, where one processor processes the SCPI commands and the user interface, and another processor (the slave processor) processes commands to control the power hardware.) The state of the slave is the actual state of the sequence, since the slave processor actually executes a sequence. The RAM copy will normally reflect the state of the slave. The response of the query is in the following format:

Ram{n} = "rstate", slave[m] = "sstate"

where rstate may be any of the following:

"RUNNING", "PAUSED", "STOPPED", "EDIT", "EMPTY", or "ERROR"

and sstate may be any of the following:

"RUNNING", "PAUSED", "STOPPED", or "ERROR"</td><td>N</td></tr>
</table>

8.13.4 SGX MEASURE SCPI Command Subsystem

The POWer query was added to the Measure subsystem in SGX to measure power once it is set up. (Section 8.6 contains the Measure commands that can be used in SGX):

`MEASure:POWer?` (see Section 8.13.)

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
MEASure		N
:POWer?	Returns the value for present power in watts being dissipated by the load. The power is measured by taking voltage and current measurement pairs three times, and averaging the result to a single wattage reading.	N

8.13.5 SGX HTRIGGER SCPI Command Subsystem

This section applies only to units with the External User Interface (Currently only available on Ethernet versions).

The SEQuence command was added to the Htrigger subsystem in SGX to trigger a preprogrammed ramp or sequence. (Sections 8.10 contains the Htrigger commands that can be used in SGX).

`HTRIGger:SEQuence` (see Section 8.13.)

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition.

Command	Description	SCPI
HTRIGger		N
:SEQuence <string>	Loads a Sequence to be initialized by the external Hardware Trigger. This command must be followed by the HTRIG:ARM command before the Hardware Trigger becomes operational.	N
:SEQuence?	Returns the currently loaded sequence name	N
:ARM <boolean>	Readies the last loaded sequence to run when the external Hardware Trigger signal is issued. 1 = Ready to run sequence when Hardware trigger is engaged. 0 = Not Armed, Hardware trigger is ignored NOTE: To arm a different sequence, issue the HTRIG:SEQ command first.	N
:ARM?	Returns the ready status of the last loaded sequence. 1 = Ready to run sequence when Hardware trigger is engaged. 0 = Not Armed, Hardware trigger is ignored NOTE: To load a new sequence to be armed you must issue the HTRIG:SEQ command first.	N

8.14 EXAMPLES OF USING THE SCPI COMMANDS

The following examples demonstrate programming a power supply to control and to readback the output using the SCPI commands. The maximum voltage and current output is dependent on the particular model. The examples list only the SCPI commands; the code required to send the commands is dependent on the type of language you are using (e.g., C or BASIC) and GPIB hardware (e.g., National Instruments).

8.14.1 VI Mode Example

Program a unit with no load at the output to 5 VDC @ 1A, and verify the output.

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on the unit.
*CLS                // clear the unit to its power-on default settings.
*RST                // reset the unit.
SOUR:CURR 1.0       // program output current to 1.0 A.
```

```
SOUR:CURR?          // confirm the output current setting (response: 1.0).
SOUR:VOLT 5.0       // program output voltage to 5.0 VDC.
SOUR:VOLT?          // confirm the output voltage setting (response: 5.0).
MEAS:CURR?          // measure the actual output current (response: ~ 0.0 with no load on
                    output).
MEAS:VOLT?          // measure the actual output voltage (response: ~ 5.0).
```

8.14.2 OVP Setup Example

Program a unit with no load at the output to generate a GPIB service request upon an overvoltage protection trip condition. (Must use GPIB not RS232.)

```
// Use SYST:ERR? after each command to verify no programming errors.
// assure that PON is not selected on the rear panel switch.
// Turn on the unit.
*CLS                   // clear the unit to its power-on default settings.
*RST                   // reset the unit.
SOUR:VOLT:PROT 4.0     // program the OVP trip point to 4.0 VDC.
SOUR:VOLT:PROT?        // confirm the OVP trip point setting (response: 4.0).
SOUR:CURR 1.0          // program output current to 1.0 A.
SOUR:VOLT 3.0          // program output voltage to 3.0 VDC.
STAT:PROT:ENABLE 8     // program the unit to report OVP trip.
STAT:PROT:ENABLE?      // confirm that OVP fault is enabled (response: 8).
*SRE 2                 // enable the GPIB service request upon a fault.
*SRE?                  // confirm the GPIB service request enabled (response 2).
STAT:PROT:EVENT?       // confirm no faults occurred (response: 0).
                       // confirm that the OVP led and SRQ led is not active.
SOUR:VOLT 7.0          // program output voltage to 7.0 VDC - cause OVP trip!
                       // confirm that unit issued a GPIB service request (use a serial
                          poll).
```

8.14.3 Trigger Example

Program a unit with no load at the output to change its output voltage and current to 5 VDC @ 1A at the same time.

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on the unit.
*CLS                    // clear the unit to its power-on default settings.
*RST                    // reset the unit.
SOUR:CURR:TRIG 1.0      // program output current to 1.0 A upon trigger.
SOUR:CURR:TRIG?         // confirm output current set to 1.0 A upon trigger.
SOUR:VOLT:TRIG 5.0      // program output voltage to 5.0 VDC upon trigger
SOUR:VOLT:TRIG?         // confirm output current set to 5.0 VDC upon trigger.
MEAS:CURR?              // measure the actual output current (response: 0.0).
MEAS:VOLT?              // measure the actual output voltage (response: 0.0).
TRIG:TYPE 3             // trigger the unit to implement curr and volt programming.
MEAS:CURR?              // measure the actual output current (response: ~ 0. 0 with no load
                        on output).
MEAS:VOLT?              // measure the actual output voltage (response: ~ 5.0).
TRIG:ABORT              // turn off trigger mode.
```

8.14.4 Hardware Trigger Example

This example applies only to units with the External User Interface (Currently only available on Ethernet versions).

Program a unit to execute a test sequence “TEST02” when a hardware trigger is issued using the External User Interface (the sequence must be programmed before it can be loaded).

```
HTRIG:SEQ “TEST02”      // load the sequence named “TEST02” into memory.
HTRIG:SEQ?              // verify that “TEST02” was loaded.
HTRIG:ARM 1             // arm the hardware trigger.
HTRIG:ARM?              // verify that the hardware trigger was armed.
                        // invoke the hardware trigger.
                        // sequence “TEST02” will run.
PROG:STAT?              // verify that the sequence is running or finished.
```

8.14.5 Ramp V Example

Program a unit with no load at the output to ramp its output voltage from 5 VDC to 25 VDC in 30 seconds.

Note: *The maximum output voltage is dependent upon the power supply rating.*

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on the unit.
*CLS                        // clear the unit to its power-on default settings.
*RST                        // reset the unit.
SOUR:CURR 33.0              // program output current to 33.0 A.
SOUR:VOLT 5.0               // program output voltage to 5.0 VDC.
SOUR:VOLT:RAMP 25.0 30.0    // program voltage to ramp from the present
                            // value (5.0 VDC) to 25.0 VDC in 30 seconds.
```

8.14.6 Ramp I Example

Program a unit with the output shorted to ramp its output current from 5A to 25A in 30 seconds.

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on with no load at the output.
*CLS                        // clear the unit to its power-on default settings.
*RST                        // reset the unit.
                            // short the output.
SOUR:VOLT 33.0              // program output voltage to 33.0 VDC.
SOUR:CURR 5.0               // program output current to 5.0 A.
SOUR:CURR:RAMP 25.0 30.0    // program current to ramp from the present
                            // value (5.0 A) to 25.0 A in 30 seconds.
```

8.14.7 Ramp V Example 2

Program a unit with no load at the output, to ramp its output voltage from 5 VDC to 25 VDC in 30 seconds upon the trigger command.

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on the unit.
*CLS                           // clear the unit to its power-on default settings.
*RST                           // reset the unit.
SOUR:CURR 33.0                 // program output current to 33.0 A.
SOUR:VOLT 5.0                  // program output voltage to 5.0 VDC.
SOUR:VOLT:RAMP:TRIG 25.0 30.0  // program voltage to ramp from the present
                               // value (5.0 VDC) to 25.0 VDC in 30 secs.
                               // upon the trigger command.
```

```
TRIG:RAMP                   // start ramp execution.
TRIG:ABORT                  // turn off trigger mode.
```

8.14.8 Power On INIT Example

Program a unit to power-on and initialize to 2 VDC @ 1A with an overvoltage protection level of 3 VDC. Verify proper power-on initialization.

```
// Use SYST:ERR? after each command to verify no programming errors.
// turn on the unit.
*CLS                     // clear the unit to its power on default settings.
*RST                     // reset the unit.
CAL:INIT:CURR 1.0        // set power-on initial current to 1.0 A.
CAL:INIT:CURR?           // confirm power-on initial current setting.
CAL:INIT:VOLT 2.0        // set power-on initial voltage to 2.0 VDC.
CAL:INIT:VOLT?           // confirm power-on initial voltage setting.
CAL:INIT:VOLT:PROT 3.0   // set power-on initial overvoltage protection to 3.0 VDC.
CAL:INIT:VOLT:PROT?      // confirm power-on initial overvoltage protection setting.
CAL:UNLOCK "6867"        // unlock nonvolatile memory for calibration value storage.
CAL:STORE                // store the calibration values in nonvolatile memory.
CAL:LOCK                 // lock nonvolatile memory for calibration value protection.
// cycle power to the unit.
// note voltage is initialized to 2.0 VDC via front panel.
SOUR:CURR?               // confirm power-on initial current setting.
SOUR:VOLT?               // confirm power-on initial voltage setting.
SOUR:VOLT:PROT?          // confirm power-on initial overvoltage protection setting.
```

8.14.9 Sequence Creation and Execution Examples

These examples assume that the sequence memory is empty. In other words, no sequences are presently defined. If you are doing these examples for a second time, then the sequence memory must first be cleared. Clear all sequence memory by issuing the command PROG:DEL:ALL.

Delete all sequences from non-volatile memory using the following command:

```
PROG:DEL:ALL
```

EXAMPLE 1

Issue *RST to reset the unit to a known state.

To create a sequence named "SEQ1", first issue the NAME command as follows:

```
   PROG:NAME  "SEQ1"
```

If you query the state of the selected sequence, it responds EMPTY.

```
PROG:STAT?
```

Response: “EMPTY”

Then memory needs to be allocated to the newly named SEQ1 sequence:

```
PROG:MALL DEFAULT
```

After memory has been allocated, the state of the SEQ1 sequence is EDIT, as the following query shows:

```
PROG: STAT?
```

Response: “EDIT”

Once the sequence is in the EDIT state, we are ready to send a sequence of programming steps to define a sequence of power supply settings:

```
PROG:DEF  1, VIMODE,3,4,11,10     //go to 3 volts, 4 amps, 11 volts ovp, for 10 seconds

PROG:DEF  2, RAMPTOV,3,5,4,11,10  //ramp from 3 to 5 volts in 10 sec.

PROG:DEF  3, VIMODE,  5, 4, 11, 10    //hold 5 volts for 10 seconds

PROG:DEF  4, RAMPTOV, 5, 3, 4, 11, 10  //ramp from 5 to 3 volts in 10 sec.

PROG:DEF  5, VIMODE,  3, 4, 11, 10    //hold 3 volts for 10 seconds

PROG:DEF  6, STOP                          //stops running the sequence while the unit
                                        remains at the state of the last command within
                                        the sequence.
```

Only 6 steps are intended for this example, so the sequence is complete. To take the sequence out of the EDIT state, it needs to be sent to the COMPLETE state, which becomes the STOPPED state, using the following command:

```
PROG:STAT COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[0]=“STOPPED”,Slave[0]=”STOPPED”

Once in the STOPPED state, the sequence is ready to run again. To rerun the sequence, issue the following two commands:

```
OUTP:STAT  ON

PROG:STATE  RUN
```

The sequence should run, and the output of the power supply for Example 1 should look like the following:

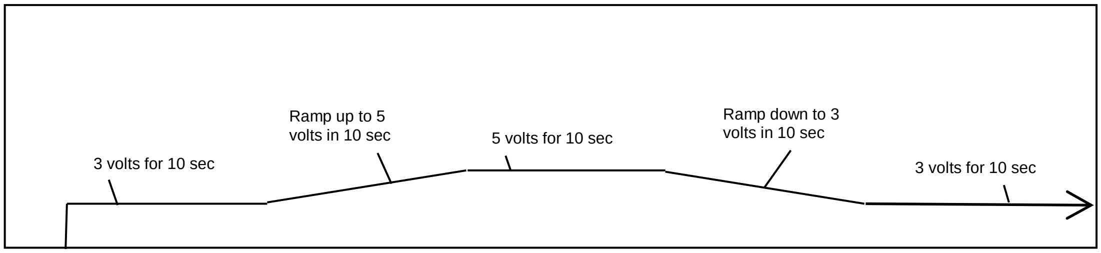


Figure 8-4. Power Supply Output for Example 1

Leave SEQ1 in the power supply's sequence memory, and create a second example sequence to also keep in sequence memory. (There is enough memory for a total of 50 different sequences.)

EXAMPLE 2

Issue `*RST` to establish a known state.

```
PROG:NAME "SEQ2"

PROG:MALL DEFAULT

PROG:DEF  1, VIMODE,10,4,11,5    //go to 10 volts, 4 amps, 11 volts ovp, for 5 seconds

PROG:DEF 2, RAMPTOV,10,2,4,11,9 //ramp from 10 volts down to 2 volts in 10 seconds

PROG:DEF  3,RETURN

PROG:STAT COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[22]="STOPPED",Slave[22]="STOPPED"

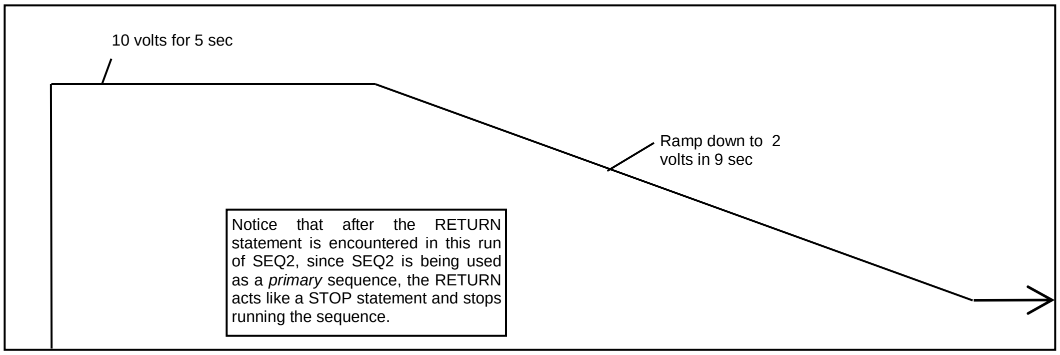
(Notice in this example the sequence is being ended with a RETURN rather than a STOP. A RETURN is more flexible because it automatically acts like a stop when SEQ2 is run directly; and yet, if SEQ2 is run as a sub-sequence, then the RETURN shall act as a statement to return to the calling sequence.)

To run SEQ2 directly, issue the command as before:

```
OUTP:STAT  ON

PROG:STAT  RUN
```

For the running sequence, the power supply output for Example 2 should look like the following:



Notice that after the RETURN statement is encountered in this run of SEQ2, since SEQ2 is being used as a *primary* sequence, the RETURN acts like a STOP statement and stops running the sequence.

Figure 8-5. Power Supply Output for Example 2

Once the ramp goes down to 2 volts, the sequence stops and the unit remains at the state of the last command within the sequence.

EXAMPLE 3

Issue `*RST` to establish a known state.

In this third example, redefine SEQ1 so that it calls SEQ2 as a subroutine.

To modify SEQ1 requires that it be deleted, and re-written. Overwriting a completed, existing sequence is not allowed by the SCPI command interface for code readability reasons. Even though overwriting is not permitted, re-writing is permitted after a sequence has been deleted. Delete the SEQ1 sequence as follows:

```
PROG:NAME “SEQ1”

PROG:DEL:SEL
```

That deletes SEQ1.

Now create a new SEQ1 that calls SEQ2 as a subsequence.

```
PROG:NAME “SEQ1”

PROG:MALL DEFAULT

PROG:DEF 1, VIMODE, 3, 4, 11, 10          //go to 3 volts, 4 amps, 11 volts ovp, for 10 sec

PROG:DEF 2, RAMPTOV, 3, 5, 4, 11, 10      //ramp from 3 volts to 5 volts in 10
                                          sec

PROG:DEF 3, VIMODE, 5, 4, 11, 10          //hold 5 volts for 10 sec
```

```
PROG:DEF 4, RAMPTOV, 5, 3, 4, 11, 10      //ramp from 5 volts to 3 volts in 10 sec

PROG:DEF 5, VIMODE, 3, 4, 11, 10          //hold 3 volts for 10 sec

PROG:DEF 6, SUBCALL, "SEQ2"               //call SEQ2 as a subsequence

PROG:DEF 7, VIMODE, 4, 5, 11, 6           //go to 4 volts, 5 amps, 11 volts ovp, for 6 sec

PROG:DEF 8, STOP                          //stop running the sequence while the unit remains at the state of the last command within the sequence.

PROG:STAT  COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[0]="STOPPED",Slave[0]="STOPPED"

```
OUTP:STAT  ON

PROG:STAT  RUN
```

The output of the supply for Example 3 should look like the following diagram:

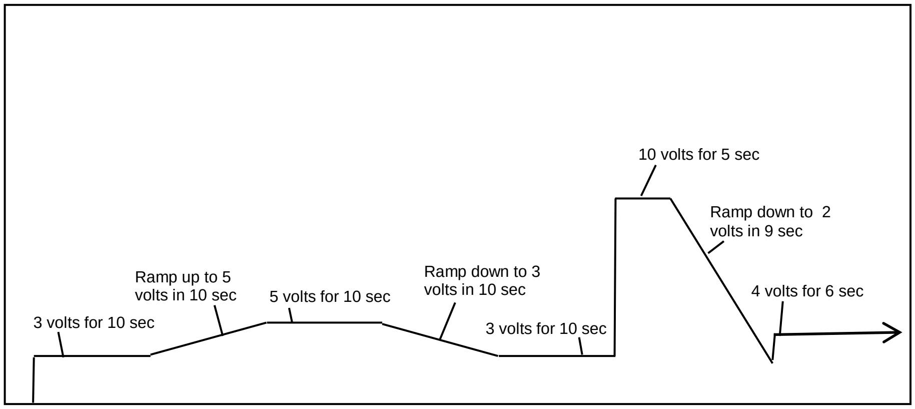


Figure 8-6. Power Supply Output for Example 3

The above diagram shows how the new SEQ1 sequence looks when SEQ2 is installed as a subsequence.

EXAMPLE 4

Issue `*RST` to establish a known state.

In this example, create a sequence that pauses at its end to let the last settings remain in effect. To do this use the `PAUSE` command.

Now create a new SEQ3:

```
PROG:SEL:NAME “SEQ3”

PROG:MALL DEFAULT

PROG:DEF 1, VIMODE, 3, 4, 11, 10    //go to 3 volts, 4 amps, 11 volts ovp, for 10
                                               sec

PROG:DEF 2, RAMPTOV, 3, 5, 4, 11, 10 //ramp from 3 volts to 5 volts in 10 sec

PROG:DEF 3, VIMODE, 5, 4, 11, 10   //hold 5 volts for 10 sec

PROG:DEF 4, RAMPTOV, 5, 3, 4, 11, 10 //ramp from 5 volts to 3 volts in 10 sec

PROG:DEF 5, VIMODE, 3, 4, 11, 10   //hold 3 volts for 10 sec

PROG:DEF 6, SUBCALL, “SEQ2”          //call SEQ2 as a subsequence

PROG:DEF 7, VIMODE, 4, 5, 11, 6   //go to 4 volts, 5 amps, 11 volts ovp, for 6 sec

PROG:DEF 8, PAUSE                  //allow the output to remain at the last setting

PROG:DEF 9, STOP                     //stop running the sequence while the unit
                                      remains at the state of the last command
                                      within the sequence.

PROG:STAT  COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[44]=“STOPPED”,Slave[44]=”STOPPED”

```
OUTP:STAT  ON

PROG:STAT  RUN
```

Following is a diagram for Example 4.

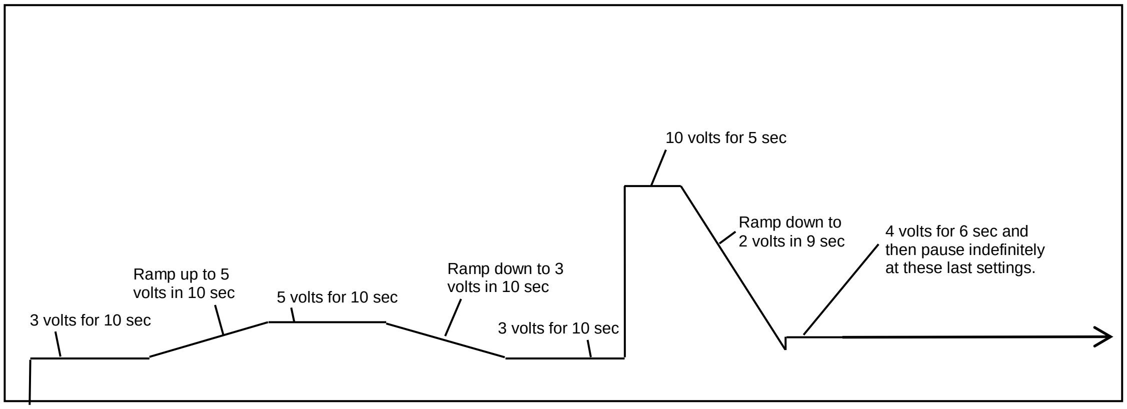

10 volts for 5 sec
Ramp down to 2 volts in 9 sec
4 volts for 6 sec and then pause indefinitely at these last settings.
Ramp up to 5 volts in 10 sec
5 volts for 10 sec
Ramp down to 3 volts in 10 sec
3 volts for 10 sec
3 volts for 10 sec

Figure 8-7. End-of-Sequence Pause for Example 4

EXAMPLE 5

To make a sequence longer than the maximum of 20 steps in an ordinary sequence, there are either one of two ways or a combination of the two. One way is to use the `SUBCALL` command. The second way is to use the `GOTO` command. The use of the `SUBCALL` command was shown in Examples 3 and 4 above. Now join two 20-step sequences with a `GOTO` command.

First, create a sequence wherein the power supply output will simulate a capacitor charge curve to a maximum of 10 volts and discharge curve to zero volts.

The formula for charging is $10(1-e^{-\frac{t}{RC}})$.

Issue `*RST` to establish a known state.

```
PROG:NAME "Charge"                            //This sequence will simulate an RC=1
                                              charge curve

PROG:MALL DEFAULT

PROG:DEF 1, VIMODE, 0, 5, 20, 1               //go to 0 volts for 1 second

PROG:DEF 2, RAMPTOV, 0, 0.95, 5, 20, 0.1      //ramp from 0 to 0.95 volts in 0.1 sec

PROG:DEF 3, RAMPTOV, 0.95, 1.81, 5, 20, 0.1   //ramp from 0.95 volts to 1.81 volts in
                                                  0.1sec

PROG:DEF 4, RAMPTOV, 1.81, 2.59, 5, 20, 0.1   //ramp from 1.81 volts to 2.59 volts in
                                                  0.1sec

PROG:DEF 5, RAMPTOV, 2.59, 3.30, 5, 20, 0.1   //ramp from 2.59 volts to 3.30 volts in
```

```
                                                  0.1sec

PROG:DEF 6, RAMPTOV, 3.30, 3.93, 5, 20, 0.1  //ramp from 3.30 volts to 3.93 volts in
                                                  0.1sec

PROG:DEF 7, RAMPTOV, 3.93, 4.51, 5, 20, 0.1  //ramp from 3.93 volts to 4.51 volts in
                                                  0.1 sec

PROG:DEF 8, RAMPTOV, 4.51, 5.03, 5, 20, 0.1  //ramp from 4.51 volts to 5.03 volts in
                                                  0.1 sec

PROG:DEF 9, RAMPTOV, 5.03, 5.51, 5, 20, 0.1  //ramp from 5.03 volts to 5.51 volts in
                                                  0.1 sec

PROG:DEF 10, RAMPTOV, 5.51, 5.93, 5, 20, 0.1  //ramp from 5.51 volts to 5.93 volts in
                                                  0.1 sec

PROG:DEF 11, RAMPTOV, 5.93, 6.32, 5, 20, 0.1 //ramp from 5.93 volts to 6.32 volts in
                                                  0.1 sec

PROG:DEF 12, RAMPTOV, 6.32, 6.67, 5, 20, 0.1 //(ramp from 6.32 volts to 6.67 volts in
                                                  0.1 sec

PROG:DEF 13, RAMPTOV, 6.67, 6.99, 5, 20, 0.1  //ramp from 6.67 volts to 6.99 volts in
                                                  0.1 sec

PROG:DEF 14, RAMPTOV, 6.99, 7.27, 5, 20, 0.1  //ramp from 6.99 volts to 7.27 volts in
                                                  0.1 sec

PROG:DEF 15, RAMPTOV,  7.27, 7.53, 5, 20, 0.1     (ramp from 7.27 volts to 7.53 volts
                                                  in 0.1 sec)

PROG:DEF 16, RAMPTOV,  7.53, 7.77, 5, 20, 0.1     (ramp from 7.53 volts to 7.77 volts
                                                  in 0.1 sec)

PROG:DEF 17, RAMPTOV,  7.77, 7.98, 5, 20, 0.1     (ramp from 7.77 volts to 7.98 volts
                                                  in 0.1 sec)

PROG:DEF 18, RAMPTOV,  7.98, 8.17, 5, 20, 0.1     (ramp from 7.98 volts to 8.17 volts
                                                  in 0.1 sec)

PROG:DEF 19, RAMPTOV,  8.17, 8.31, 5, 20, 0.1     (ramp from 8.17 volts to 8.31 volts
                                                  in 0.1 sec)

PROG:DEF 20, RAMPTOV,  8.31, 8.50, 5, 20, 0.1     (ramp from 8.31 volts to 8.50 volts
                                                  in 0.1 sec)

PROG:DEF 21, GOTO, “Discharge”                      (step 21 may only be a STOP,
                                                  RETURN, or GOTO)

PROG:DEF 22, STOP

PROG:STAT  COMPLETE
```

```
PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[66]=“STOPPED”,Slave[66]=”STOPPED”

The formula for discharging is $Ae^{-\frac{t}{RC}}$, where A is the calculated amplitude achieved by the previous charge cycle.

```
PROG:NAME “Discharge”

PROG:MALL DEFAULT

PROG:DEF 1, RAMPTOV, 8.50, 7.69, 5, 20, 0.1 //ramp from 8.50 volts down to
                                                  7.69 volts in 0.1 sec

PROG:DEF 2, RAMPTOV, 7.69, 6.95, 5, 20, 0.1 //ramp from 7.69 volts down to
                                                  6.95 volts in 0.1 sec

PROG:DEF 3, RAMPTOV, 6.95, 6.29, 5, 20, 0.1 //ramp from 6.95 volts down to
                                                  6.29 volts in 0.1 sec

PROG:DEF 4, RAMPTOV, 6.29, 5.70, 5, 20, 0.1 //ramp from 6.29 volts down to
                                                 5.70 volts in 0.1 sec

PROG:DEF 5, RAMPTOV, 5.70, 4.66, 5, 20, 0.1 //ramp from 5.70 volts down to
                                                 4.66 volts in 0.1 sec

PROG:DEF 6, RAMPTOV, 4.66, 4.22, 5, 20, 0.1 //ramp from 4.66 volts down to
                                                 4.22 volts in 0.1 sec

PROG:DEF  7, RAMPTOV, 4.22, 3.82, 5, 20, 0.1 //ramp from 4.22 volts down to
                                                 3.82 volts in 0.1 sec

PROG:DEF  8, RAMPTOV, 3.82, 3.46, 5, 20, 0.1 //ramp from 3.82 volts down to
                                                 3.46 volts in 0.1 sec

PROG:DEF  9, RAMPTOV, 3.46, 3.12, 5, 20, 0.1 //ramp from 3.46 volts down to
                                                 3.12 volts in 0.1 sec

PROG:DEF 10,RAMPTOV, 3.12, 2.83, 5, 20, 0.1 //ramp from 3.12 volts down to
                                                 2.83 volts in 0.1 sec

PROG:DEF 11,RAMPTOV, 2.83, 2.56, 5, 20, 0.1 //ramp from 2.83 volts down to
                                                 2.56 volts in 0.1 sec

PROG:DEF 12,RAMPTOV, 2.56, 2.31, 5, 20, 0.1 //ramp from 2.56 volts down to
                                                 2.31 volts in 0.1 sec

PROG:DEF 13,RAMPTOV, 2.31, 2.10, 5, 20, 0.1 //ramp from 2.31 volts down to
                                                 2.10 volts in 0.1 sec

PROG:DEF 14,RAMPTOV, 2.10, 1.90, 5, 20, 0.1 //ramp from 2.10 volts down to
```

```
                                             1.90 volts in 0.1 sec

PROG:DEF 15,RAMPTOV, 1.90, 1.72, 5, 20, 0.1 //ramp from 1.90 volts down to
                                             1.72 volts in 0.1 sec

PROG:DEF 16,RAMPTOV, 1.72, 1.55, 5, 20, 0.1 //ramp from 1.72 volts down to
                                             1.55 volts in 0.1 sec

PROG:DEF 17,RAMPTOV, 1.55, 1.40, 5, 20, 0.1 //ramp from 1.55 volts down to
                                             1.40 volts in 0.1 sec

PROG:DEF 18,RAMPTOV, 1.40, 1.27, 5, 20, 0.1 //ramp from 1.40 volts down to
                                             1.27 volts in 0.1 sec

PROG:DEF 19,RAMPTOV, 1.27, 1.15, 5, 20, 0.1 //ramp from 1.27 volts down to
                                             1.15 volts in 0.1 sec

PROG:DEF 20,RAMPTOV, 1.15, 1.04, 5, 20, 0.1 //ramp from 1.15 volts down to
                                             1.04 volts in 0.1 sec

PROG:DEF 21, RETURN                         //step 21 may only be a STOP,
                                             RETURN, or GOTO

PROG:STAT COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[88]="STOPPED",Slave[88]="STOPPED"

To run the combined sequence, simply select the first sequence, and then run it.

```
PROG:NAME "Charge"

OUTP:STAT ON

PROG:STAT  RUN
```

The output for Example 5 should look like the following diagram:

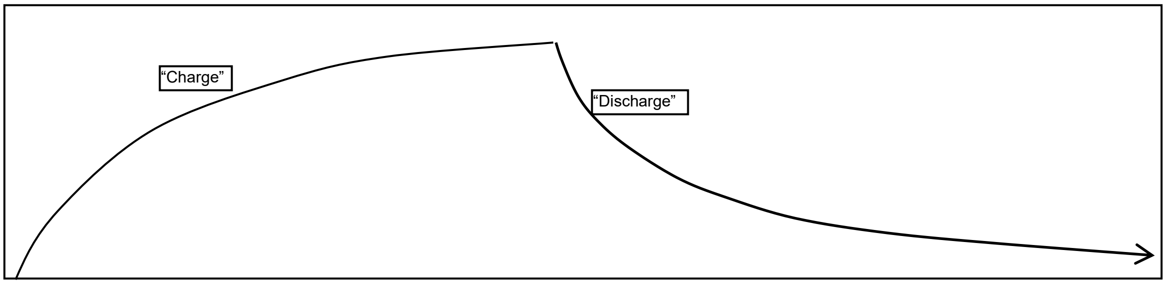


Figure 8-8. Power Supply Output for Example 5

EXAMPLE 6

This example shows how to make an infinite loop for the creation of a continuous square wave:

Issue `*RST` to establish a known state.

```
PROG:NAME  "Square Wave"

PROG:MALL  DEFAULT

PROG:DEF 1, VIMODE, 0, 5, 15, 0.5  //go to 0 volts, 5 amps, 15 volts ovp, for 0.5
                                          seconds

PROG:DEF 2, VIMODE, 10,5,15,0.5    //go to 10 volts, 5 amps, 15 volts ovp, for 0.5
                                          seconds

PROG:DEF 3,GOTO, "Square Wave"            //go to top of this sequence and loop
                                          indefinitely)

PROG:STAT  COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[110]="STOPPED",Slave[110]="STOPPED"

```
OUTP:STAT ON

PROG:STAT  RUN
```

The output for Example 6 should look something like the following:

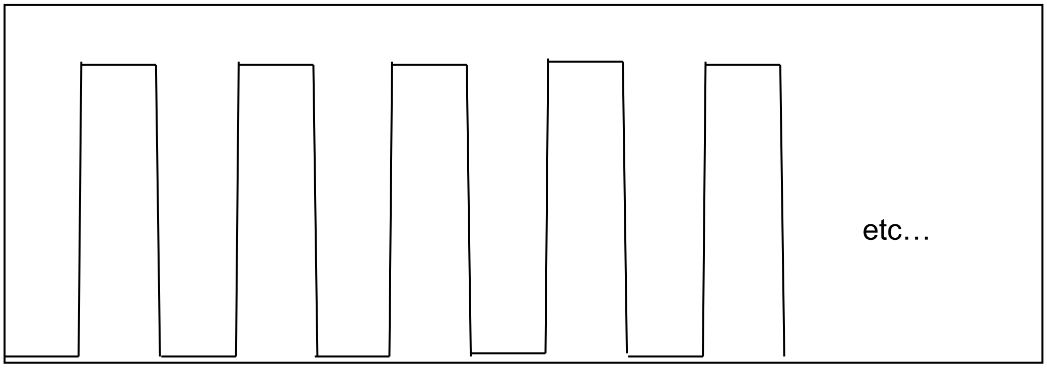


Figure 8-9. Power Supply Output for Example 6

To stop the execution of the square wave sequence, issue the STOP state command:

```
PROG:STAT  STOP
```

The sequence should stop running while the unit remains at the state of the last command within the sequence.

To pause instead of stop, issue the PAUSE state command instead:

```
PROG:STAT  PAUSE
```

After pausing, the output will remain at its present value.

EXAMPLE 7

This example shows how to make a definite length square wave (as opposed to an infinite length square wave) that has a specified number of cycles.

Issue `*RST` to establish a known state.

```
PROG:NAME “Pulse Train”

PROG:MALL DEFAULT

PROG:DEF 1, VIMODE, 0, 5, 15, 1

PROG:DEF 2, RAMPTOV, 0, 4, 5, 15, 1

PROG:DEF 3, LOOP, 10

PROG:DEF 4, VIMODE, 4, 5, 15, 1

PROG:DEF 5, VIMODE, 0, 5, 15, 1

PROG:DEF 6, NEXT

PROG:DEF 7, VIMODE, 4, 5, 15, 1

PROG:DEF 8, RAMPTOV, 4, 0, 5, 15, 1

PROG:DEF 9, STOP

PROG:STAT  COMPLETE

PROG:SAVE:SEL

PROG:STAT?
```

Response: Ram[132]=“STOPPED”,Slave[132]=”STOPPED”

```
OUTP:STAT ON

PROG:STAT  RUN
```

The power supply output for Example 7 should look something like the following:

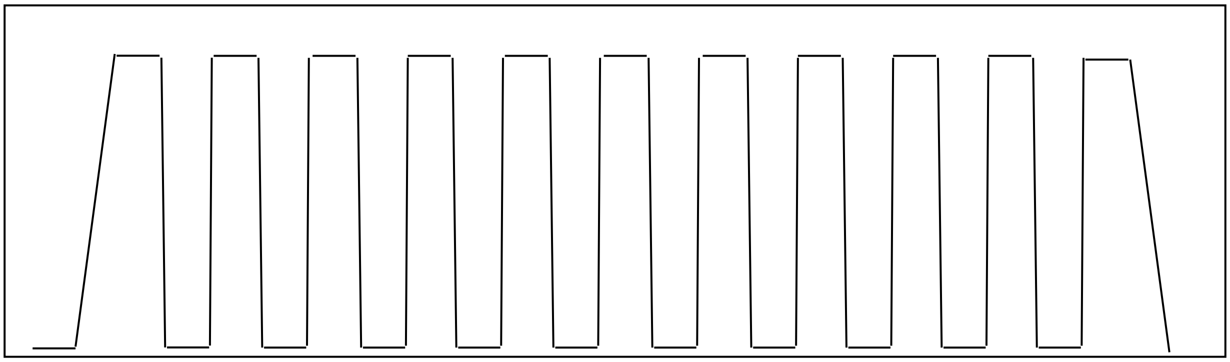


Figure 8-10. Power Supply Output for Example 7

This page intentionally left blank.

9
CALIBRATION

> **CAUTION**
> Please refer to the power supply operation manual for further information before performing calibration procedures. Only qualified personnel who deal with attendant hazards in power supplies, are allowed to perform calibration procedures. If calibration is not performed properly, functional problems could arise, requiring that the supply be returned to the factory.

9.1 INTRODUCTION

The SGX Power Supply is calibrated to adjust internal signal levels to correspond to the expected supply output signal levels. You must perform the calibration procedures if the power supply's programming or readback performance falls out of specification due to component aging drifts. Refer to your power supply operation manual to find the required calibration interval. The SGX Power Supply is calibrated for output voltage programming, output current programming, output overvoltage protection programming, voltage readback, and current readback.

The calibration procedures in the following sections are designed to be performed at ambient temperature of 25°C ± 5°C, after the unit has had a stable output and a stable load for at least 30 minutes.

The following test equipment is required in addition to the computer system to complete the following calibration:

- Model HP 344401A or equivalent 6-digit digital voltmeter (DVM)
- Current shunt rated for an accuracy of 0.25% or better and a minimum of 125% of the UUT full scale output current

NOTE: All calibration procedures steps should be performed. Omitting any step or applicable section may affect the specified performance of the unit.

The CAL:MOD:LASTCALDATE and CAL:MOD:NEXTCALDATE commands (Section 8.12) should be issued once calibration procedures have been performed.

9.2 SETUP FOR CALIBRATION

This section provides calibration setup for RS232, Ethernet and GPIB; note that Step 6 provides separate instructions for each control interface.

STEP DESCRIPTION

1. Disconnect the Unit Under Test (UUT) AC input power. (The UUT is the power supply that will be calibrated.)
2. Disconnect all loads from the UUT's output terminals.
3. Connect the UUT sense lines to the output terminals. Refer to the power supply operation manual for further information on remote sensing.
4. Connect the DVM to the UUT output terminals.
5. Set the UUT to REMOTE mode by the rear panel Remote/Local Configuration Switch, see Figure 3-2.
6. Perform one of the following:

   For **GPIB**:
   Ensure that the correct GPIB primary address has been set by the UUT by using the front panel menu, see Figure 5-2. Connect the GPIB controller to the rear panel of the UUT using a GPIB cable.

   For **RS232**:
   Connect the RS232 controller to the rear panel of the UUT using the RS232 null modem cable (see Figure 4-3).

   For **Ethernet**:
   Refer to Section 6.4 for Ethernet communication setup. Connect the RJ-45 Ethernet cable to the rear panel of the UUT.
7. Reconnect the AC input power. Turn the unit ON and allow the unit to warm up for at least 30 minutes.

The UUT is ready for the calibration procedure to be performed.

WARNING

Exercise caution when using and servicing power supplies:

- High energy levels can be present at the output terminals on all power supplies in normal operation.
- Potentially lethal voltages exist within the power supply and on the output terminals of power supplies that are rated at 40V and over.
- Filter capacitors store potentially dangerous energy for some time after power is removed.

9.3 VOLTAGE PROGRAMMING CALIBRATION

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Program the overvoltage protection to maximum to prevent nuisance trips:

```
CAL:OUTP:VOLT:PROT:DAC 65535
```

4. Program the output current to full scale to prevent Constant-Current operation:

```
CAL:OUTP:CURR:DAC 65535
```

5. Program the output of the first calibration point by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 3275
```

6. Let the output settle and measure the voltage with a high precision voltmeter; this is value 1.

7. Enter the actual output voltage value of the first calibration point:

```
CAL:OUTP:VOLT:FIVEPOINT 1 <value 1>
```

8. Program the output of the second calibration point by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 19000
```

9. Let the output settle and measure the voltage with a high precision voltmeter; this is value 2.

10. Enter the actual output voltage value of the second calibration point:

```
CAL:OUTP:VOLT:FIVEPOINT 2 <value 2>
```

11. Program the output of the third calibration point by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 32000
```

12. Let the output settle and measure the voltage with a high precision voltmeter; this is value 3.

13. Enter the actual output voltage value of the third calibration point:

```
CAL:OUTP:VOLT:FIVEPOINT 3 <value 3>
```

14. Program the output of the fourth calibration point by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 45000
```

15. Let the output settle and measure the voltage with a high precision voltmeter; this is value 4.

16. Enter the actual output voltage value of the fourth calibration point:

```
CAL:OUTP:VOLT:FIVEPOINT 4 <value 4>
```

17. Program the output of the fifth calibration point by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 62250
```

18. Let the output settle and measure the voltage with a high precision voltmeter; this is value 5.

19. Enter the actual output voltage value of the fifth calibration point:

```
CAL:OUTP:VOLT:FIVEPOINT 5 <value 5>
```

20. To review entered data for five–point voltage calibration, issue the following query:

```
CAL:OUTP:VOLT:FIVEPOINT?
```

21. Program the supply to unlock the non-volatile memory for calibration value storage:

```
CAL:UNLOCK “6867”
```

22. Program the supply to store the calibration values in non-volatile memory:

```
CAL:STORE
```

Allow 15 seconds for the non-volatile memory to be updated with the new calibration values.

23. Program the supply to lock the non-volatile memory for calibration value protection.

```
CAL:LOCK
```

9.4 VOLTAGE PROGRAM GAIN/OFFSET AND MEASUREMENT READBACK CALIBRATION

NOTE: This calibration procedure is required in order to use Constant Power Mode programming.

STEP DESCRIPTION

1. Program the overvoltage protection to maximum to prevent nuisance trips:

```
CAL:OUTP:VOLT:PROT:DAC 65535
```

2. Program the output current to full scale to prevent Constant-Current operation:

```
CAL:OUTP:CURR:DAC 65535
```

3. Program the output of the first calibration point to approximately 15% of full scale voltage by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 9830
```

4. Let the output settle and measure the voltage with the meter.

5. Enter the actual voltage readback corresponding to the DAC value 9830 of the first calibration point:

```
CAL:OUTP:VOLT:POINT 1 <voltage>
CAL:MEAS:VOLT:POINT 1 <voltage>
```

6. Program the output of the second calibration point to approximately 85% of full scale voltage by sending the following command string from the computer:

```
CAL:OUTP:VOLT:DAC 55700
```

7. Let the output settle and measure the voltage with the meter.

8. Enter the actual voltage readback corresponding to the DAC value 5570 of the second calibration point:

```
CAL:OUTP:VOLT:POINT 2 <voltage>
CAL:MEAS:VOLT:POINT 2 <voltage>
```

9. Reset the output voltage to 0 volts.

```
CAL:OUTP:VOLT:DAC 0
```

10. Program the SGX Power Supply to calculate the voltage readback calibration gain and offset values:

```
CAL:OUTP:VOLT:CALC
CAL:MEAS:VOLT:CALC
```

11. Program the SGX Power Supply to unlock the non-volatile memory for calibration value storage:

```
CAL:UNLOCK “6867”
```

12. Program the SGX Power Supply to store the calibration values in non-volatile memory:

    ```
    CAL:STORE
    ```

    Allow 15 seconds for the non-volatile memory to be updated with the new calibration values.

13. Program the SGX Power Supply to lock the non-volatile memory for calibration value protection:

    ```
    CAL:LOCK
    ```

9.5 OVERVOLTAGE PROTECTION PROGRAMMING CALIBRATION

The overvoltage protection calibration procedure requires calibrated output voltage programming and voltage readback for proper function.

STEP DESCRIPTION

1. Program the SGX Power Supply to self-calibrate the overvoltage protection:

    ```
    CAL:OUTP:VOLT:PROT:CALC
    ```

    The overvoltage protection calibration function requires over 30 seconds. Use *ESE 1 and serial polls to detect when calibration is done.

2. Program the SGX Power Supply to unlock the non-volatile memory for calibration value storage:

    ```
    CAL:UNLOCK “6867”
    ```

3. Program the SGX Power Supply to store the calibration values in non-volatile memory:

    ```
    CAL:STORE
    ```

    Allow 15 seconds for the non-volatile memory to be updated with the new calibration values.

4. Program the SGX Power Supply to lock the non-volatile memory for calibration value protection

    ```
    CAL:LOCK
    ```

9.6 CURRENT PROGRAMMING CALIBRATION

Preparation:

1. Power down the Unit Under Test (UUT) and remove the input power for safety.
2. Allow 5 minutes for the energy in the output to bleed down to a safe level.
3. Attach the precision shunt between the power supply's output terminals.
4. Attach the voltmeter across the shunt.
5. Apply input power and power up the UUT.
6. After the UUT has initialized, begin the calibration procedure.

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Program the overvoltage protection to maximum to prevent nuisance trips:

```
CAL:OUTP:VOLT:PROT:DAC 65535
```

4. Program the output voltage to full scale to prevent Constant-Voltage operation:

```
CAL:OUTP:VOLT:DAC 65535
```

5. Program the output of the first calibration point by sending the following command string from the computer:

```
CAL:OUTP:CURR:DAC 3275
```

6. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 1.

7. Enter the actual output current of the first calibration point:

```
CAL:OUTP:CURR:FIVEPOINT 1 <value 1>
```

8. Program the output of the second calibration point by sending the following command string from the computer:

```
CAL:OUTP:CURR:DAC 19000
```

9. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 2.

10. Enter the actual output current of the second calibration point:

```
CAL:OUTP:CURR:FIVEPOINT 2 <value 2>
```

11. Program the output of the third calibration point by sending the following command string from the computer:

    ```
    CAL:OUTP:CURR:DAC 32000
    ```

12. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 3.

13. Enter the actual output current of the third calibration point:

    ```
    CAL:OUTP:CURR:FIVEPOINT 3 <value 3>
    ```

14. Program the output of the fourth calibration point by sending the following command string from the computer:

    ```
    CAL:OUTP:CURR:DAC 45000
    ```

15. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 4.

16. Enter the actual output current of the fourth calibration point:

    ```
    CAL:OUTP:CURR:FIVEPOINT 4 <value 4>
    ```

17. Program the output of the fifth calibration point by sending the following command string from the computer:

    ```
    CAL:OUTP:CURR:DAC 62250
    ```

18. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 5.

19. Enter the actual output current of the fifth calibration point:

    ```
    CAL:OUTP:CURR:FIVEPOINT 5 <value 5>
    ```

20. To review entered data for five–point current calibration, issue the following query:

    ```
    CAL:OUTP:CURR:FIVEPOINT?
    ```

21. Program the supply to unlock the non-volatile memory for calibration value storage:

    ```
    CAL:UNLOCK "6867"
    ```

22. Program the supply to store the calibration values in non-volatile memory:

    ```
    CAL:STORE
    ```

    Allow 15 seconds for the non-volatile memory to be updated with the new calibration values.

23. Program the supply to lock the non-volatile memory for calibration value protection:

```
CAL:LOCK
```

9.7 CURRENT PROGRAMING GAIN/OFFSET AND MEASUREMENT READBACK CALIBRATION

NOTE: This calibration procedure is required in order to use Constant Power Mode programming.

STEP DESCRIPTION

1. Program the overvoltage protection to maximum to prevent nuisance trips:

```
CAL:OUTP:VOLT:PROT:DAC 65535
```

2. Program the output voltage to full scale to prevent Constant-Voltage operation:

```
CAL:OUTP:VOLT:DAC 65535
```

3. Program the output of the first calibration point to approximately 15% of full scale current by sending the following command string from the computer:

```
CAL:OUTP:CURR:DAC 9830
```

4. Let the output settle and measure the current with the current shunt and the meter.

5. Enter the actual current readback corresponding to the DAC value 9830 of the first calibration point:

```
CAL:OUTP:CURR:POINT 1 <current>
CAL:MEAS:CURR:POINT 1 <current>
```

6. Program the output of the second calibration point to approximately 85% of full scale current by sending the following command string from the computer:

```
CAL:OUTP:CURR:DAC 55700
```

7. Let the output settle and measure the current with the current shunt and the meter.

8. Enter the actual current readback corresponding to the DAC value 55700 of the second calibration point:

```
CAL:OUTP:CURR:POINT 2 <current>
CAL:MEAS:CURR:POINT 2 <current>
```

9. Reset the output current to 0 amps.

```
CAL:OUTP:CURR:DAC 0
```

10. Program the SGX Power Supply to calculate the current readback calibration gain and offset values:

```
CAL:OUTP:CURR:CALC

CAL:MEAS:CURR:CALC
```

11. Program the SGX Power Supply to unlock the non-volatile memory for calibration value storage:

```
CAL:UNLOCK “6867”
```

12. Program the SGX Power Supply to store the date:

```
CAL:MOD:LASTCALDATE MM DD YYYY
CAL:MOD:NEXTCALDATE MM DD YYYY
```

13. Program the SGX Power Supply to store the calibration values in non-volatile memory:

```
CAL:STORE
```

Allow 15 seconds for the non-volatile memory to be updated with the new calibration values.

14. Program the SGX Power Supply to lock the non-volatile memory for calibration value protection

```
CAL:LOCK
```

Calibration is complete.

9.8 ANALOG PROGRAM ADJUSTMENT

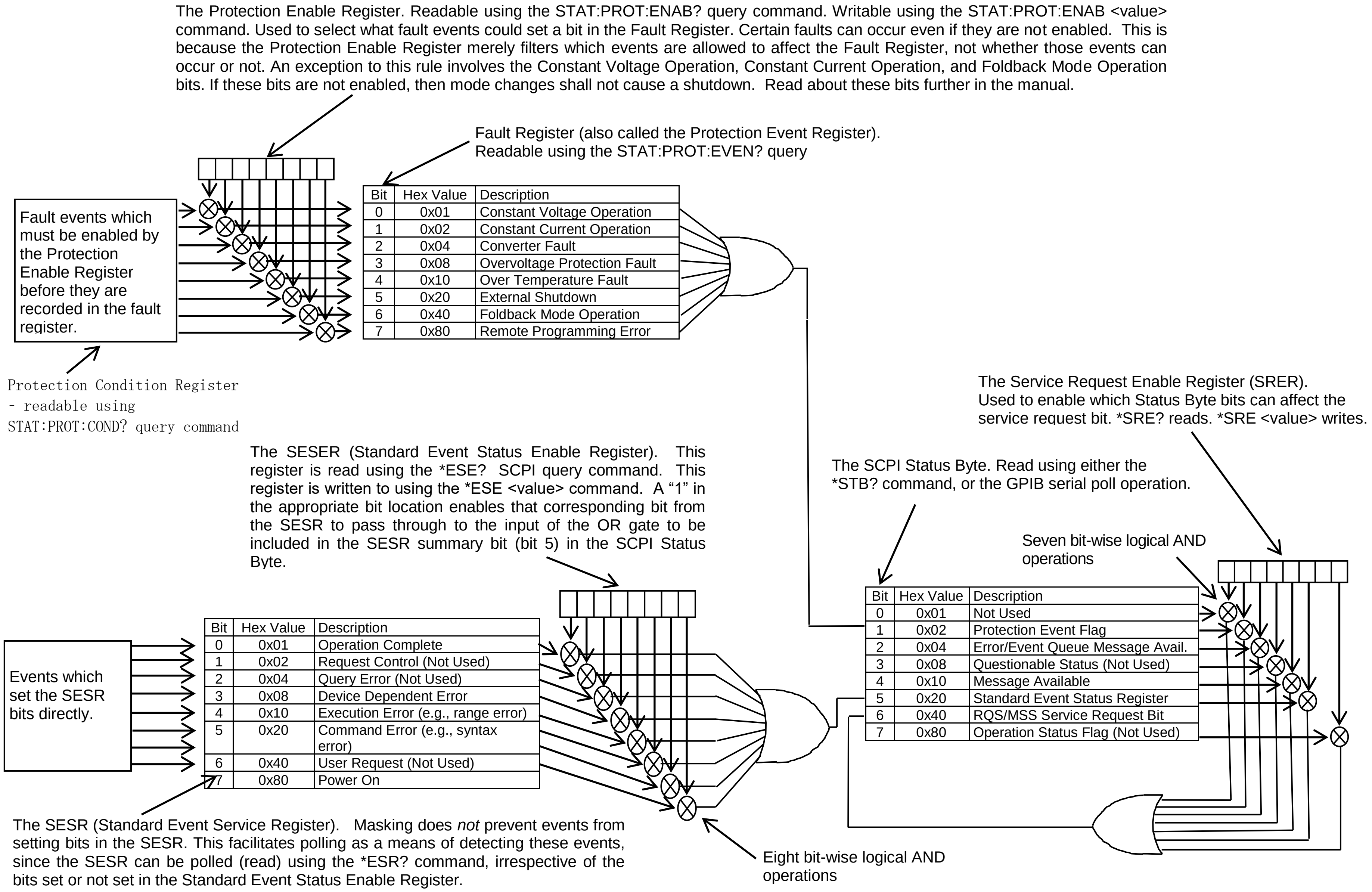
The analog control interface calibration requires opening of the chassis top cover and it should be carried out by service personnel only. Contact service and maintenance department for the same.

10
SCPI STATUS IMPLEMENTATION

10.1 SCPI STATUS BYTE IMPLEMENTATION

SCPI status byte implementation for the SGX power supply is described in Figure 10-1.

This page intentionally left blank.



The Protection Enable Register. Readable using the STAT:PROT:ENAB? query command. Writable using the STAT:PROT:ENAB <value> command. Used to select what fault events could set a bit in the Fault Register. Certain faults can occur even if they are not enabled. This is because the Protection Enable Register merely filters which events are allowed to affect the Fault Register, not whether those events can occur or not. An exception to this rule involves the Constant Voltage Operation, Constant Current Operation, and Foldback Mode Operation bits. If these bits are not enabled, then mode changes shall not cause a shutdown. Read about these bits further in the manual.

Fault Register (also called the Protection Event Register). Readable using the STAT:PROT:EVEN? query

Fault events which must be enabled by the Protection Enable Register before they are recorded in the fault register.

Bit	Hex Value	Description
0	0x01	Constant Voltage Operation
1	0x02	Constant Current Operation
2	0x04	Converter Fault
3	0x08	Overvoltage Protection Fault
4	0x10	Over Temperature Fault
5	0x20	External Shutdown
6	0x40	Foldback Mode Operation
7	0x80	Remote Programming Error

Protection Condition Register - readable using STAT:PROT:COND? query command

The Service Request Enable Register (SRER). Used to enable which Status Byte bits can affect the service request bit. *SRE? reads. *SRE <value> writes.

The SESER (Standard Event Status Enable Register). This register is read using the *ESE? SCPI query command. This register is written to using the *ESE <value> command. A "1" in the appropriate bit location enables that corresponding bit from the SESR to pass through to the input of the OR gate to be included in the SESR summary bit (bit 5) in the SCPI Status Byte.

The SCPI Status Byte. Read using either the *STB? command, or the GPIB serial poll operation.

Seven bit-wise logical AND operations

Bit	Hex Value	Description
0	0x01	Not Used
1	0x02	Protection Event Flag
2	0x04	Error/Event Queue Message Avail.
3	0x08	Questionable Status (Not Used)
4	0x10	Message Available
5	0x20	Standard Event Status Register
6	0x40	RQS/MSS Service Request Bit
7	0x80	Operation Status Flag (Not Used)

Events which set the SESR bits directly.

Bit	Hex Value	Description
0	0x01	Operation Complete
1	0x02	Request Control (Not Used)
2	0x04	Query Error (Not Used)
3	0x08	Device Dependent Error
4	0x10	Execution Error (e.g., range error)
5	0x20	Command Error (e.g., syntax error)
6	0x40	User Request (Not Used)
7	0x80	Power On

The SESR (Standard Event Service Register). Masking does *not* prevent events from setting bits in the SESR. This facilitates polling as a means of detecting these events, since the SESR can be polled (read) using the *ESR? command, irrespective of the bits set or not set in the Standard Event Status Enable Register.

Eight bit-wise logical AND operations

Figure 10-1. SCPI Status Implementation Diagram

This page intentionally left blank.

INDEX

C

CALibrate, 8-24
Calibration
  Current Measurement/Readback Calibration, 9-9
  Setup for Calibration, 9-2
  Voltage Programming Calibration, 9-3
  Warning, 9-1
Calibration Procedrues
  Current Programming Gain/Offset and Measurement Readback, Ethernet, GPIB, *9-9*
  Introduction, *9-1*
  Voltage Programming, Ethernet, 9-3
Calibration Procedures, 9-1
  Analog Program Adjustment, *9-10*
  Current Programming, Ethernet, 9-7
  Setup, *9-2*
  Voltage Program Gain/Offset and Measurement Readback, Ethernet, GPIB, *9-5*
CALIBRATION SCPI Command Reference, 8-26
CalibrationProcedrues
  Overvoltage Protection Programming Calibration, Ethernet, GPIB, 9-6
Configuration
  Ethernet/Lan, 6-1
  RS232 Setup Procedure, 4-3
  Setup Procedure, Ethernet, 6-3
Configuration and Remote Programming
  Ethernet, 6-1
Configurations
  IEEE 488.2 GPIB Setup Procedures, 5-2
  Setup Procedure, DHCP, 6-3
Conventions
  [ ], 8-8

E

Error/Event Queue, 8-5
  SCPI Error Codes, 8-5
Ethernet Interface, 1-2
Ethernet Web Pages, Overview, *6-9*
Examples, Using SCPI Commands, 8-37
  Hardware Trigger, *8-39*
  OVP Setup, *8-38*
  Power On INIT, *8-41*
  Ramp I, *8-40*
  Ramp V, *8-40*
  Ramp V, 2, *8-40*
  Sequences, *8-41*
  Trigger, *8-39*
  VI Mode, *8-37*
External User Control Signal Connector
  FAULT output signal, 7-3
  FOLDBACK output signal, 7-3
  ISOLATION output signal, 7-3
  POLARITY output signal, 7-3
  SENSE output signal, 7-3
  TRIGGER OUT output signal, 7-3
External User Control Signal Connector, 7-1
External User Control Signal Connector
  SHUTDOWN input signal, 7-3
External User Control Signal Connector
  TRIGGER IN input signal, 7-3
External User Control Signal Connector
  Warning, 7-5

F

Features
   IEEE 488.2 GPIB, 2-1
Features of Ethernet interface, 2-2
Features of RS232 interface, 2-1
Features, Functions and Specifications, 2-1
Functions, **2-2**
   Programmable, **2-2**
   Readback, 2-2

G

Global Channel 0, 5-2

H

HTRIGGER SCPI Command Reference, *8-22*

I

IEEE 488.2 GPIB Configurations and Remote Programming, 5-1
IEEE 488.2 GPIB Interface, 1-2
IEEE-488.2 Common Command Subsystem, 8-8
   *CLS, 8-9
   *ESE, 8-9
   *ESR?, 8-9
   *IDN?, 8-9
   *OPC, 8-9
   *RST, 8-9
   *SRE, 8-9
   *STB?, 8-9
   *WAI, 8-9
Interactive Control, 6-13

L

Local Operation, 3-2
LXI Identification, 6-14

M

MEASURE SCPI Command Reference, 8-15
MEASURE SCPI Command Subsystem, 8-15

N

Network Setup Procedure
   DHCP, 6-3

O

OUTPUT SCPI Command Reference, 8-16
OUTPUT SCPI Command Subsystem, 8-16
Overview, 1-1

P

Parameter Definitions, 8-7
   boolean, 8-7
   NR1, 8-7
   NRf, 8-7
   string, 8-7
Power-On Conditions
   Default, 3-3
Programming Accuracy
   IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Programming Resolution
   IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Programming/Communication via Ethernet, 6-8
   Raw Socket Interface, 6-8
   Web Server, 6-8
Protection Event Status Register, 8-1

Q

Queries, 8-8

R

Ramp Funciton, *8-14*
Ramping
   Description, 8-14, 8-22
Readback Accuracy
   IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Readback Resolution
   IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Rear Panel, 4-1
   Ethernet Layout, 6-2
Rear Panel Configuration Switch
   Address Selection, GPIB, 5-2
   Power-On GPIB Service Request (PON SRQ) Selection, 5-2
Remote Operation, 3-2
Remote Programming via RS-232, 4-2
Remote/Local Selection, 3-1
Restrictions
   Sequence Programming, 8-29
RS232 Configurations and Remote Programming, 4-1
RS232 Interface, 1-1
RS232 Rear Panel RJ-type 6P6C Connector Pinout, 4-2
RS232/Ethernet/IEEE 488.2 GPIB and SCPI Command Operation, 8-1

S

SCPI Command Operation
   CALibrate, 8-24
   STATus, 8-17
SCPI Command Reference, 8-32

SCPI Command Subsystem
SGX Program, 8-31
SGX Source, 8-29
SCPI Commands
Application Examples, *8-37*
CALIBRATION, 8-24
HTRIGGER, *8-22*
Measure, *8-15*
Output, *8-16*
SGX Measure, 8-36
SGX Trigger, 8-37
Source, *8-10*
Status, *8-17*
System, *8-19*
SCPI Conformance, 8-7
SCPI Conventions, 8-8
SCPI Parameter Definitions, *8-7*
SCPI STATUS IMPLEMENTATION, 10-1
SCPI Status Byte, 8-4
Sequence Programming
Restrictions, 8-29
Serial Poll Operation, 8-6
Setup Procedure, 6-3
SGX Banner and Tab, 6-9
Shield Ground, GPIB, 5-2
Source SCPI Command Reference, *8-11*
Specifications
IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Specifications, IEEE 288.2 GPIB, RS232 and Ethernet, 2-3
Standard Event Status Register (ESR), 8-3
STATus, 8-17
STATUS SCPI Command Reference, 8-18
STATUS SCPI Command Subsystem, 8-17
Synchronization Pulse, 7-2
SYSTEM FAULT REGISTERS, 8-21
System Requirements, Ethernet Minimum, 2-4
SYSTEM SCPI Command Reference, 8-20
SYSTEM SCPI Command Subsystem, 8-19

T

TRIGGER SCPI Command Reference, *8-23*
TRIGGER SCPI Command Subsystem, 8-23

U

Units, 8-8

W

Web Server
Configuration page, 6-11
Home page, 6-10
Login window, 6-9