产

品

手

册

Asterion DC Series

仪器型号：

西安安泰测试科技有限公司

仪器维修|租赁|销售|测试

地址:西安市高新区纬二十六路 369 号 电话：400-876-5512

网址: www.agitekservice.com 座机：029-88827159

AMETEK®
PROGRAMMABLE POWER

Sorensen
Asterion DC Series
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

Notice of Copyright

Exclusion for Documentation

Date and Revision

April 5,2022 Rev B

Part Number

M330461-01

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 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.

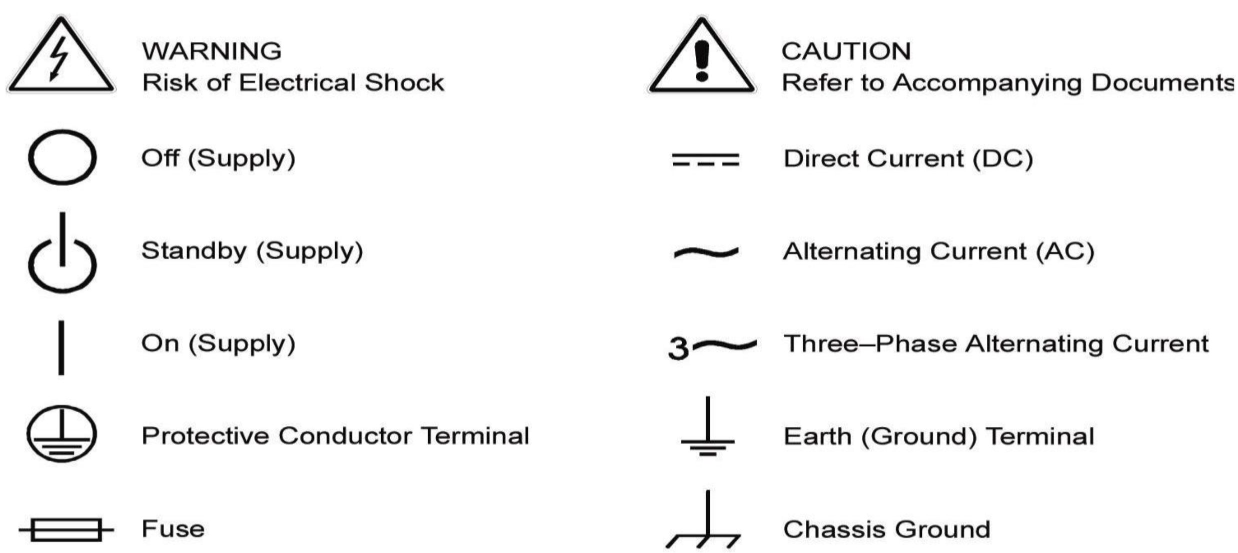
Always ensure that facility AC input power is de-energized prior to connecting or disconnecting any cable.

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: Asterion DC Series Programming Manual 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 USB Interface 1-2
1.4 Ethernet Interface 1-2
1.5 EtherCAT Interface..1-2
1.6 IEEE 488.2 GPIB Interface 1-1

2 FEATURES, FUNCTIONS AND SPECIFICATIONS 2-2

2.1 Introduction...2-2
2.2 Features of RS232 interface 2-2
2.3 Features of USB interface 2-2
2.4 Features of Ethernet interface 2-2
2.5 Features of EtherCAT interface 2-3
2.6 Features of IEEE 488.2 GPIB interface 2-3
2.7 Programmable Functions..2-4
2.8 Readback Functions...2-4
2.9 Specifications 2-4
2.9.1 Programming Resolution 2-4
2.9.2 Programming Accuracy 2-5
2.9.3 Readback Resolution..2-5
2.9.4 Readback Accuracy..2-5
2.10 Minimum System Requirements 2-5
2.10.1 PC Connection 2-5
2.10.2 Internet or LAN Connection 2-5

3 REMOTE/LOCAL SELECTION 3-1

3.1 Introduction...3-1
3.1.1 **Remote mode:**...3-1
3.1.2 **Local Lock out Mode:**..3-2

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

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

5 USB CONFIGURATIONS AND REMOTE PROGRAMMING 5-1

5.1 Rear Panel 5-1
5.2 USB SETUP PROCEDURE...5-2
5.2.1 Port Number Identification 5-2

5.2.2 Communication test using serial interface program 5-3

6 ETHERNET CONFIGURATION AND REMOTE PROGRAMMING 6-6

6.1 Introduction 6-6
6.2 Ethernet/LAN Configuration 6-6
6.2.1 Ethernet Configuration Factory Defaults 6-6
6.3 Rear Panel 6-7
6.4 Ethernet Setup Procedure 6-7
6.4.1 Network Setup Using DHCP Server 6-7
6.4.2 Network Setup Using Auto-IP (Direct Connection between Asterion power Supply and PC using Cross cable) 6-9
6.4.3 **Network Setup using Static IP** 6-10
6.4.4 Raw Socket Interface 6-12
6.4.5 VXI-11 Protocol 6-12
6.4.6 Web Server 6-12
6.5 Ethernet Web Pages, Overview 6-12
6.5.1 Home 6-14
6.5.2 IP Configuration 6-15
6.5.3 Interactive Control 6-17
6.5.4 LXI Identification 6-18

7 ETHERCAT CONFIGURATIONS AND REMOTE PROGRAMMING.7-20

7.1 Introduction 7-20
7.2 Rear Panel 7-20
7.3 EtherCAT Setup Procedure 7-21
7.3.1 Procedure to Setup the EtherCAT 7-21
7.3.2 Installing the Required Drivers 7-22
7.3.3 To Program the EtherCAT using SSC (Slave Stack Code) Tool 7-23
7.3.4 TwinCAT XAE Shell 7-26

8 IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING 8-32

8.1 Rear Panel 8-32
8.2 Remote Programming Via IEEE488.2 GPIB connector 8-33
8.2.1 Address Selection 8-33
8.2.2 Power-On GPIB Service Request (PON SRQ) Selection 8-33
8.2.3 Shield Ground 8-33
8.2.4 IEEE 488.2 GPIB Setup Procedure 8-33

9 REMOTE ANALOG PROGRAMMING AND EXTERNAL USER CONTROL INTERFACE CONNECTOR 9-1

9.1 Introduction 9-1

9.2 REMOTE Analog programming Connector details..9-1
9.3 Remote ANALOG Programming..9-9
9.3.1 REMOTE PROGRAMMING BY RESISTANCE 9-9
9.3.2 Remote Programming BY VOLTAGE 9-11
9.3.3 Remote Programming BY 4-20mA SOURCE................................9-14
9.3.4 Remote Overvoltage Programming..9-16
9.4 Remote MONITOR OUTPUTS 9-17
9.5 REMOTE DIGITAL INPUT AND OUTPUT...9-17
9.5.1 Auxiliary power output 9-17
9.5.2 Digital outputs...9-18
9.5.3 Digital Inputs...9-20
9.5.4 Fault 9-20
9.5.5 Trigger-In & Trigger Out..9-23
9.6 Remote Output On/Off Control 9-23
9.6.1 Remote Output ON/OFF by Contact Closure 9-23
9.6.2 Remote Output ON/OFF Control by External Source.....................9-24

10 RS232/ETHERNET/ IEEE 488.2 GPIB AND SCPI COMMAND OPERATION 10-1

10.1 Introduction...10-1
10.2 Register Definitions 10-1
10.2.1 Protection Condition and Protection Event Status Register............10-1
10.2.2 Standard Event Status Register (ESR) 10-4
10.2.3 SCPI Status Byte...10-5
10.2.4 Error/Event Queue...10-7
10.2.5 Serial Poll Operation..10-8
10.2.6 Self-Test...10-8
10.2.7 Fault Status 10-9
10.3 SCPI Conformance Information..10-10
10.3.1 Parameter Definitions 10-10
10.3.2 Conventions..10-11
10.3.3 Queries...10-11
10.4 IEEE 488.2 Common Command Subsystem 10-11
10.5 SOURCE SCPI Command Subsystem 10-13
10.5.1 SOURCE SCPI Command Summary...10-13
10.5.2 SOURCE SCPI Command Reference 10-15
10.5.3 RAMP FUNCTION...10-21
10.5.4 MEASURE SCPI Command Subsystem.......................................10-22
10.5.5 MEASURE SCPI Command Summary 10-22
10.5.6 MEASURE SCPI Command Reference 10-22
10.6 OUTPUT SCPI Command Subsystem..10-23
10.6.1 OUTPUT SCPI Command Summary 10-23
10.6.2 OUTPUT SCPI Command Reference..10-24

10.7 SYSTEM SCPI Command Subsystem 10-25
10.7.1 SYSTEM SCPI Command Summary 10-25
10.7.2 SYSTEM SCPI Command Reference 10-26
10.8 HTRIGGER SCPI Command Subsystem 10-28
10.8.1 HTRIGGER SCPI Command Summary 10-28
10.8.2 HTRIGGER SCPI Command Reference 10-28
10.9 TRIGGER SCPI Command Subsystem 10-28
10.9.1 TRIGGER SCPI Command Summary 10-28
10.9.2 TRIGGER SCPI Command Reference 10-28
10.10 CALIBRATION SCPI Command Subsystem 10-29
10.10.1 CALIBRATION SCPI Command Summary 10-29
10.10.2 CALIBRATION SCPI Command Reference 10-32
10.12 STATUS SCPI Command Subsystem 10-39
10.12.1 STATUS SCPI Command Summary 10-39
10.12.2 STATUS SCPI Command Reference 10-39
10.13 Examples of Using the SCPI Commands 10-39
10.13.1 VI Mode Example 10-39
10.13.2 OVP Setup Example 10-40
10.13.3 Trigger Example 10-41
10.13.4 Ramp I Example 10-41
10.13.5 Ramp V Example 2 10-42
10.13.6 Power On INIT Example 10-42

11 SEQUENCING FUNCTION 11-43

11.1 Introduction 11-43
11.2 Program SCPI Command Subsystem 11-43
11.2.1 Program SCPI Command Summary 11-43
11.2.2 Program SCPI Command Reference 11-44
11.2.3 Sequence Creation and Execution Examples 11-48

12 CALIBRATION 12-60

12.1 Introduction 12-60
12.2 Setup for Calibration 12-61
12.3 Voltage Programming Calibration 12-62
12.4 Voltage Programming Calibration – REMOTE SENSE 12-64
12.5 Current Programming Calibration 12-67
12.6 Analog Program Adjustment 12-69
12.6.1 Voltage programming – 10V range voltage source calibration 12-69
12.6.2 Voltage programming – 5V range voltage source calibration 12-70
12.6.3 Voltage programming – current source 10k ohms (Resistance Programming) calibration 12-71
12.6.4 Voltage programming – current source 5k ohms (Resistance Programming) calibration 12-71
12.6.5 Voltage programming – 4-20mA source calibration 12-72

12.6.6 Current programming – 10V range voltage source calibration 12-73
12.6.7 Current programming – 5V range voltage source calibration 12-74
12.6.8 Current programming – 10k ohms current source (Resistance Programming) calibration...12-74
12.6.9 Current programming – 5k ohms current source (Resistance Programming) calibration...12-75
12.6.10 Current programming – 4-20mA source calibration.......................12-76
12.6.11 Power programming – 10V range voltage source calibration 12-76
12.6.12 POWER programming – 5V range voltage source calibration.......12-77
12.6.13 POWER programming – current source 10k ohms (Resistance Programming) calibration...12-78
12.6.14 POWER programming – current source 5k ohms (Resistance Programming) calibration...12-79
12.6.15 POWER programming – 4-20mA source calibration 12-79
12.6.16 OVER-Voltage programming – 10V range calibration 12-80
12.6.17 OVER-Voltage programming – 5V range calibration 12-81
12.7 REMOTE MONITOR OUTPUTS CALIBRATION.......................................12-81
12.7.1 Voltage Monitor Output Calibration 12-81
12.7.2 Current Monitor Output Calibration 12-82
12.7.3 Power Monitor Output Calibration 12-83

13 SCPI STATUS IMPLEMENTATION....................................... 13-84

13.1 SCPI Status Byte Implementation...13-84

LIST OF TABLES

Table 4-1. RS-232C Interface Connector Type 4-2
Table 4-2. RS-232C Interface Connector Pinout 4-2
Table 4-3. RS-232C Interface Connector Pinout 4-3
Table 9-1. Analog Programming Connector, Designations and Functions 9-8
Table 9-2. List of Faults in DC-Asterion Power Supply 9-23
Table 10-1: Protection Condition and Event Status Registers 10-3
Table 10-2: Standard Event Status Register 10-4
Table 10-3: SCPI Status Byte 10-6
Table 10-4: SCPI Error Codes 10-8
Table 10-5: Self-Test Error Codes 10-9
Table 10-6: Faults 10-10

LIST OF FIGURES

Figure 3-1. Local/Remote Screen....3-1
Figure 3-2. Dashboard Screen....3-1
Figure 3-3. Local Lockout Screen....3-2
Figure 3-4: Local Lockout – Unlock Screen....3-2
Figure 3-5. Home-Screen....3-2
Figure 4-1. Rear Panel – RS232 Interface....4-1
Figure 4-2. RS-232C Interface Connector....4-2
Figure 4-3. RS232 Communications Cable Pinout....4-4
Figure 4-4. RS232 Screen (Configure)....4-4
Figure 4-5. Port Setting in Tera Term....4-5
Figure 4-6. Baud rate Setting in Tera Term....4-5
Figure 4-7. Terminal option in Tera Term....4-6
Figure 4-8. Terminal Setup in Tera Term....4-6
Figure 6-1. Rear Panel - Ethernet Interface....6-7
Figure 6-2. LAN Screen (Configure)....6-8
Figure 6-3. LAN Screen....6-8
Figure 6-4. LAN Screen (Settings)....6-9
Figure 6-5. RS232 Screen (Settings)....6-9
Figure 6-6. LAN configure (Settings)....6-11
Figure 6-7. LAN Screen (Configure Static IP)....6-11
Figure 6-8. Asterion DC power Supply Banner and Tab....6-12
Figure 6-9. Login Window....6-13
Figure 6-10. Asterion DC power Supply Home Page....6-14
Figure 6-11. Asterion DC Power supply Device identify....6-15
Figure 6-12. Asterion DC power supply IP Configuration Page....6-15
Figure 6-13. Interactive Control Page....6-17
Figure 6-14. LXI Identification Page....6-18
Figure 7-1: Rear Panel – EtherCAT Interface....7-21
Figure 7-2: EtherCAT Enable / Disable Screen....7-21
Figure 7-3: Configure EtherCAT....7-22
Figure 7-4: Run Batch File....7-22
Figure 7-5: Install EtherCAT Compatibility Drivers....7-23
Figure 7-6: SSC Tool Launch....7-23
Figure 7-7: Open EEPROM Programmer....7-24
Figure 7-8: XML project selection....7-24
Figure 7-9: Scan for EtherCAT devices....7-24
Figure 7-10: List of the Ethernet Devices....7-25

Figure 7-11: Ethernet Devices 7-25
Figure 7-12: Found EtherCAT Devices 7-25
Figure 7-13: Program EtherCAT device 7-26
Figure 7-14: EEPROM Programming 7-26
Figure 7-15: EEPROM Programming Completed 7-26
Figure 7-16: XML project selection 7-27
Figure 7-17: Scanned EtherCAT Devices 7-27
Figure 7-18: Mapping of variables 7-28
Figure 7-19: Activate Configuraiton 7-28
Figure 7-20: Activating Configuration Window 7-28
Figure 7-21: Real-time Start Diagnostic Window 7-29
Figure 7-22: Restart TwinCAT in Run Mode 7-29
Figure 7-23: Navigation to ADS port 7-29
Figure 7-24: Verifying ADS port 7-30
Figure 7-25: Monitor screen 7-30
Figure 7-26: ViPR Unit Monitor 7-31
Figure 8-1. Rear Panel – IEEE 488.2 GPIB Interface 8-32
Figure 8-2. GPIB Screen 8-33
Figure 9-1. Rear Panel of Asterion DC 1U and 2U – 44 pin Analog Programming Connector 9-1
Figure 9-2 Remote Analog Programming and External User Control Interface Connector Pin-Out 9-2
Figure 9-3 Remote Current Programming Using Resistance 9-10
Figure 9-4 Remote Voltage Programming Using Resistance 9-10
Figure 9-5 Remote Power Programming Using Resistance 9-11
Figure 9-6 Remote Current Programming Using 0-5 VDC or 0-10 VDC Source 9-12
Figure 9-7 Remote Voltage Programming Using 0-10 VDC Source 9-13
Figure 9-8 Remote Power Programming Using 0-10 VDC Source 9-14
Figure 9-9 Remote Current Programming Using 4-20mA Source 9-15
Figure 9-10 Remote Voltage Programming Using 4-20mA Source 9-15
Figure 9-11 Remote Power Programming Using 4-20mA Source 9-16
Figure 9-12 Remote Overvoltage Programming Using DC Voltage Source 9-17
Figure 9-13 External Isolation and Polarity Relay interface to Control interface 9-19
Figure 9-14 Remote Output On/Off Control by Contact Closure 9-23
Figure 9-15 Remote Output On/Off Using Isolated AC or DC Source for Asterion DC 1U and 2U 9-24
Figure 9-16 Remote Output On/Off Using Isolated TTL/CMOS Source for Asterion DC 1U and 2U 9-24
Figure 10-1. Protection Condition and Protection Event Register 10-4
Figure 10-2. Standard Event Status and Standard Event Status Enable Register 10-5

Figure 10-3. SCPI Status Byte and Service Request Enable Register........................10-6
Figure 11-1: Power Supply Output for Example 1 11-49
Figure 11-2: Power Supply Output for Example 2 11-50
Figure 11-3: Power Supply Output for Example 3 11-52
Figure 11-4: End-of-Sequence Pause for Example 4 11-53
Figure 11-5: Power Supply Output Example 5...11-57
Figure 11-6: Power Supply Output Example 6...11-57
Figure 11-7: Power Supply Output for Example 7 11-59
Figure 12-1: RS232 Communications Cable Pinout 12-61
Figure 13-1. SCPI Status Implementation Diagram...13-85

This page intentionally left blank.

1
OVERVIEW

1.1 INTRODUCTION

This manual provides instructions for remote programming control and monitoring for Asterion DC 1U and 2U power supplies. For easy navigation to the applicable instructions, this manual separates RS232, USB, IEEE 488.2 GPIB, Ethernet and EtherCAT setup instructions. The instructions then converge where they are common to all five interface options. See Section 1.2, Section 1.3, Section 1.4 Section 1.5 and Section 1.5 for orientation. The Asterion DC power supply series provides default interface of RS232, USB and Ethernet. IEEE 488.2 GPIB and EtherCAT are optional interfaces. Use this programming manual in conjunction with your Asterion DC 1U, 2U (M330460-01) power supply operational 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

9 for Remote Analog Programming and External user control interface

10 for SCPI Commands and Definitions

11 for Sequencing Function

12 for Calibration Procedures

13 for SCPI Status Implementation

1.3 USB INTERFACE

If you are using the USB interface, go to:

2 for Features, Functions and Specifications

3 for Remote/Local Selection

5 for USB Configuration and Remote Programming

9 for Remote Analog Programming and External user control interface

10 for SCPI Commands and Definitions

11 for Sequencing Function

12 for Calibration Procedures

13 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

9 for Remote Analog Programming and External user control interface

10 for SCPI Commands and Definitions

11 for Sequencing Function

12 for Calibration Procedures

13 for SCPI Status Implementation

1.5 ETHERCAT INTERFACE

If you are using the EtherCAT interface, go to:

2 for Features, Functions and Specifications,

3 for Remote/Local Selection

7 for EtherCAT Configuration and Remote Programming

9 for Remote Analog Programming and External user control interface

10 for SCPI Commands and Definitions

11 for Sequencing Function

12 for Calibration Procedures

13 for SCPI Status Implementation

1.6 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

8 for IEEE 488.2 GPIB Configuration and Remote Programming

9 for Remote Analog Programming and External user control interface

10 for SCPI Commands and Definitions

11 for Sequencing Function

12 for Calibration Procedures

13 for SCPI Status Implementation

2 FEATURES, FUNCTIONS AND SPECIFICATIONS

2.1 INTRODUCTION

This section introduces the features, functions and specifications for RS232, USB, Ethernet, IEEE 488.2 GPIB and EtherCAT. 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(CV), Constant-Current(CC), Constant Power (CP) or Foldback mode(CV/CC/CP), with reset

Voltage Ramp and Current Ramp functions

Soft calibration

Rear panel RS232 control interface

Rear panel Remote Analog Programming and External user control interface (Common to all interfaces)

2.3 FEATURES OF USB INTERFACE

Programming and readback of voltage and current

Programmable overvoltage protection with reset

SCPI compliant command set

User selectable Constant-Voltage(CV), Constant-Current(CC), Constant Power (CP) or Foldback mode(CV/CC/CP), with reset

Voltage Ramp and Current Ramp functions

Soft calibration

Rear panel USB control interface

Rear panel Remote Analog Programming and External user control interface (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(CV), Constant-Current(CC), Constant Power (CP) or Foldback mode(CV/CC/CP), with reset

Voltage Ramp and Current Ramp functions

Field-upgradable firmware via Ethernet

Full calibration through software control

Rear panel Ethernet control interface

Rear panel Remote Analog Programming and External user control interface (Common to all interfaces)

2.5 FEATURES OF ETHERCAT 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(CV), Constant-Current(CC), Constant Power (CP) or Foldback mode(CV/CC/CP), with reset

Voltage Ramp and Current Ramp functions

Field-upgradable firmware via Ethernet

Full calibration through software control

Rear panel Ethernet control interface

Rear panel Remote Analog Programming and External user control interface (Common to all interfaces)

2.6 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(CV), Constant-Current(CC), Constant Power (CP) or Foldback mode(CV/CC/CP), with reset

Voltage Ramp and Current Ramp functions

Soft calibration

Rear panel GPIB IEEE 488.2 control interface

Rear panel Remote Analog Programming and External user control interface (Common to all interfaces)

2.7 PROGRAMMABLE FUNCTIONS

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

Output voltage and current

User limits for voltage and current

Overvoltage protection

Output enable/disable

Maskable fault interrupt

Hold and trigger

External relay control.

Full calibration

2.8 READBACK FUNCTIONS

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

Actual measured voltage and current

Voltage and current settings

User 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

2.9 SPECIFICATIONS

Specifications are subject to change without notice. Refer to your Asterion DC 1U and 2U power supply's 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 Ethernet and EtherCAT).

2.9.1 Programming Resolution

Voltage: 0.012% of full scale

Current: 0.012% of full scale

Overvoltage Protection: 0.1% of full scale (full scale is 110% of max output voltage.)

Power Output: 0.012% of full scale

2.9.2 Programming Accuracy

Voltage: ± (0.1% of maximum output voltage)
Current: ± (0.2% of maximum output current) *
Overvoltage Protection: ± (1.0% of max output voltage)
Power Output: ± (0.3% of rated output power)

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

2.9.3 Readback Resolution

Voltage: ± 0.012% of full scale
Current: ± 0.012% of full scale
Power: ± 0.012% of full scale

2.9.4 Readback Accuracy

Voltage: ± (0.1% of full scale output voltage)
Current: ± (0.2% of full scale output current) *****
Power: ± (0.3% of rated output power)

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

2.10 MINIMUM SYSTEM REQUIREMENTS

The minimum software and hardware requirements to operate your Asterion DC power supply depend on whether it is connected directly to your PC or connected to the Internet or to a Local Area Network (LAN).

2.10.1 PC Connection

To operate your Asterion DC 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

2.10.2 Internet or LAN Connection

To operate your Asterion DC 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

This section describes the various mode of operations available in the front panel.

3.1.1 Remote mode:

Unit can be set to Remote mode by Sending the command **SYST:LOCAL 0** using any of the interface available on the unit (RS232,USB,LAN, GPIB and EtherCAT).

Note: Unit can also be set to remote by sending any SCPI command using any of the interface available on the unit. (Refer to Figure 3-1)

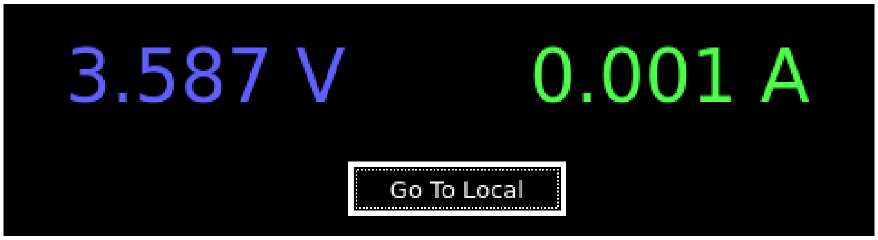


Figure 3-1. Local/Remote Screen

Upon pressing the Go To Local button Asterion DC power supply front panel screen will navigate to Dash board. (Refer to Figure 3-2)

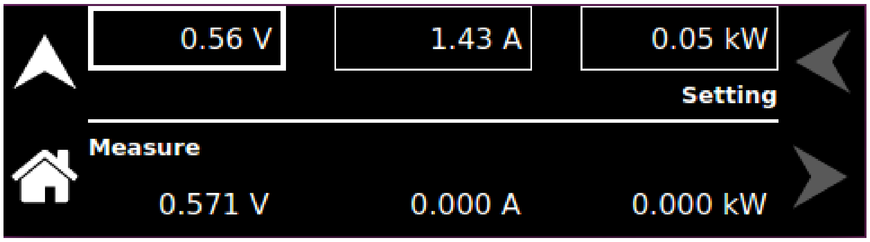


Figure 3-2. Dashboard Screen

3.1.2 Local Lock out Mode:

Using the interface available on the unit (RS232, USB, LAN, GPIB and EtherCAT), sending the command **SYST:LOCAL:LOCK** will change the unit to Local Lock out mode.

Figure 3-3. Local Lockout Screen

During this mode, All the front panel options and hardware buttons will be disabled.

To enable the device into local mode, use the special command: **SYST:LOCAL:UNLOCK "6867"** or pressing the Unlock button and entering "6867", unit mode will be changed to local mode. Refer to Figure 3-5 and Figure 3-5

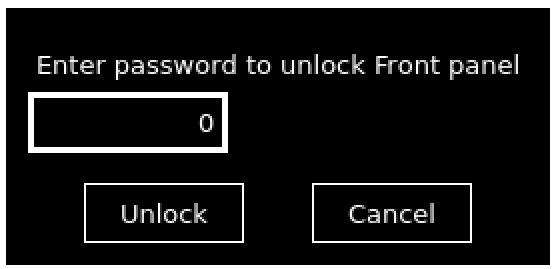


Figure 3-4: Local Lockout – Unlock Screen

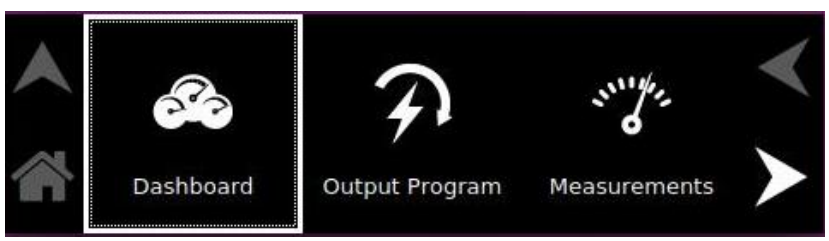


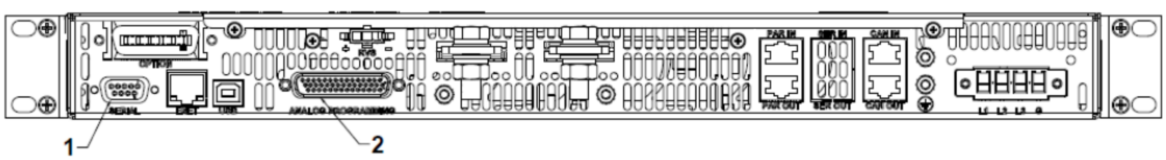
Figure 3-5. Home-Screen

This page intentionally left blank.

4
RS232
CONFIGURATIONS
AND REMOTE PROGRAMMING

4.1 REAR PANEL

This section provides illustration of the Asterion DC power supply's rear panel layout, which differs among the Asterion DC 1U and 2U power supply models. For example, refer to Figure 4-1 for Asterion DC 1U power supply's rear panel. Regardless of the layout, the component functions are common across all models and those that are pertinent to the RS232 options are described here.



1- RS232C 9-contact receptacle (female) Subminiature-D
2- Remote Analog Programming and External User Control Interface connector

Figure 4-1. Rear Panel – RS232 Interface

4.2 REMOTE PROGRAMMING VIA RS232

4.2.1 RS232 Connector Pinout

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

The RS232 interface is accessible through the rear panel DB9 connector (Refer to Figure 4-2), labeled Serial on the power supply's rear panel (see Figure 4-1)

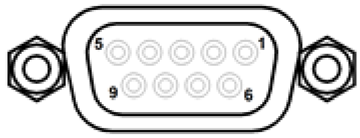


Figure 4-2. RS-232C Interface Connector

Connector	Type
RS-232C Interface	9-contact receptacle (female) Subminiature-D.

Table 4-1. RS-232C Interface Connector Type

Pin #	Name	DCE Signal	Direction
1	N/C	N/A	N/A
2	TxD	Transmit Data	Output
3	RxD	Receive Data	Input
4	N/C	N/A	N/A
5	Common	N/A	N/A
6	N/C	N/A	N/A
7	RTS	Request to Send	Input
8	CTS	Clear to Send	Output
9	N/C	N/A	N/A

Table 4-2. RS-232C Interface 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 115200	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):

D-Subminiature 9-Pin Connector Male Connector (Power supply Interface)	D-Subminiature 9-Pin Connector Female Socket (PC interface)
2(RxD)	3(TxD)
3(TxD)	2(RxD)
No Connection	No Connection
5(GND)	5(GND)
7(RTS)	8(CTS)
8(CTS)	7(RTS)

Table 4-3. RS-232C Interface Connector Pinout

TxD-Transmit
RxD-Receive
RTS-Request to Send
CTS-Clear to Send

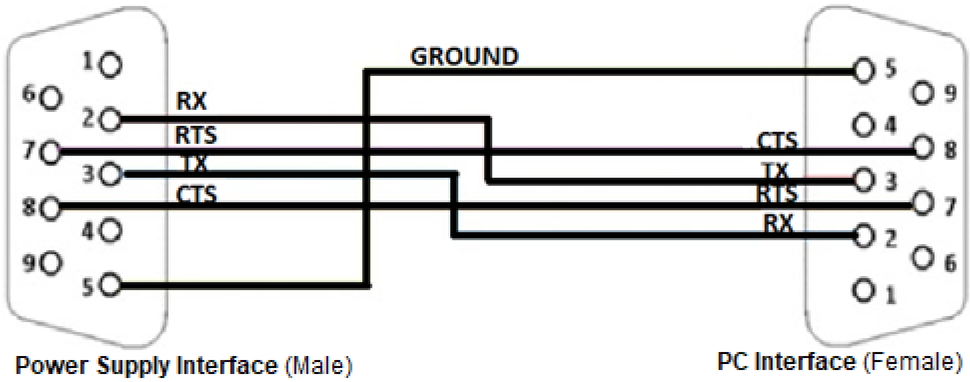
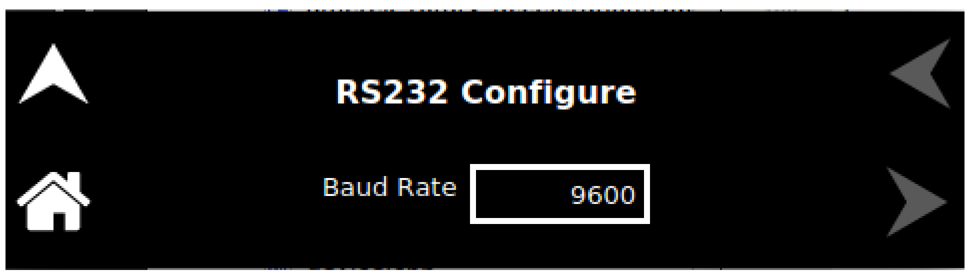


Figure 4-3. RS232 Communications Cable Pinout

2. Connect power to the unit and turn on the unit.
3. The baud rate is selectable through the front panel, see Figure 4-4. Change the baud rate for RS232 to 9600.

   To navigate to RS232, configure screen, go to Home → Control Interface → RS232 → RS232 Configure.



4. Upon tapping the text box, following baud rates selection screen will appear.

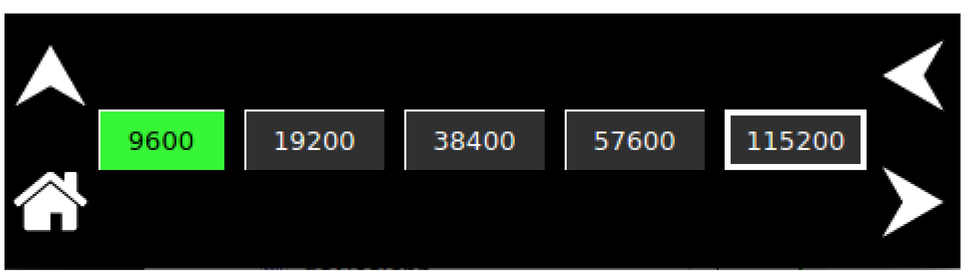


Figure 4-4. RS232 Screen (Configure)

5. Use one of the available programs for serial communication, such as Tera Term.

   If you choose to use Tera Term:

6. Select serial option in the Tera term window and select the port as COM1 and press OK. (Refer to Figure 4-5)

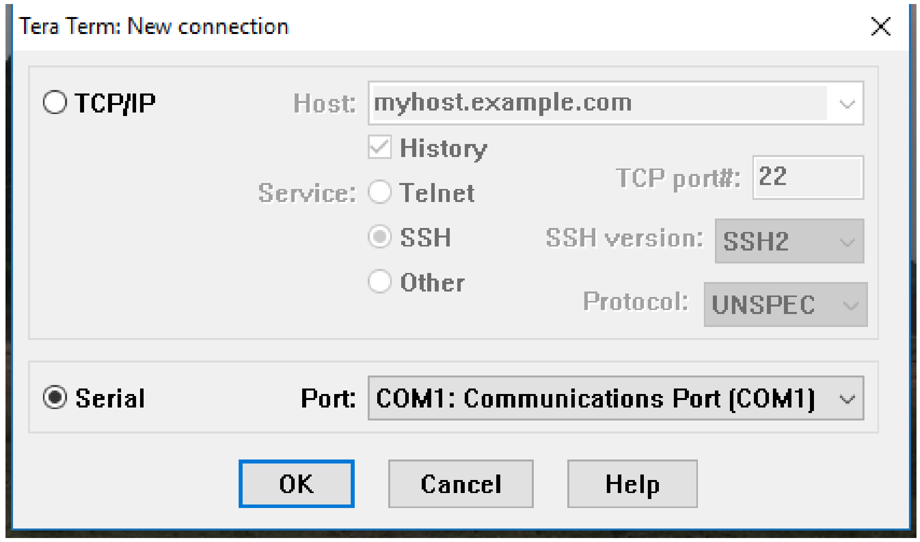


Figure 4-5. Port Setting in Tera Term

7. Select setup and select serial port in the setup window.

8. Set the baud rate to 9600, 8 data bits, no parity, 1 stop bit, and no flow control. (Refer to Figure 4-6)

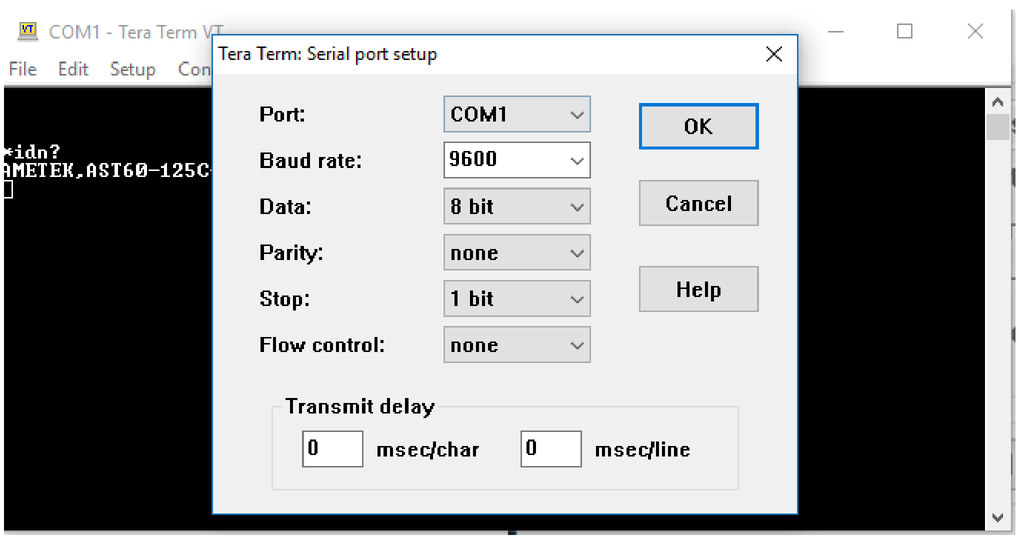


Figure 4-6. Baud rate Setting in Tera Term

9. Select the Terminal from the Tera Term. (Refer to Figure 4-7)

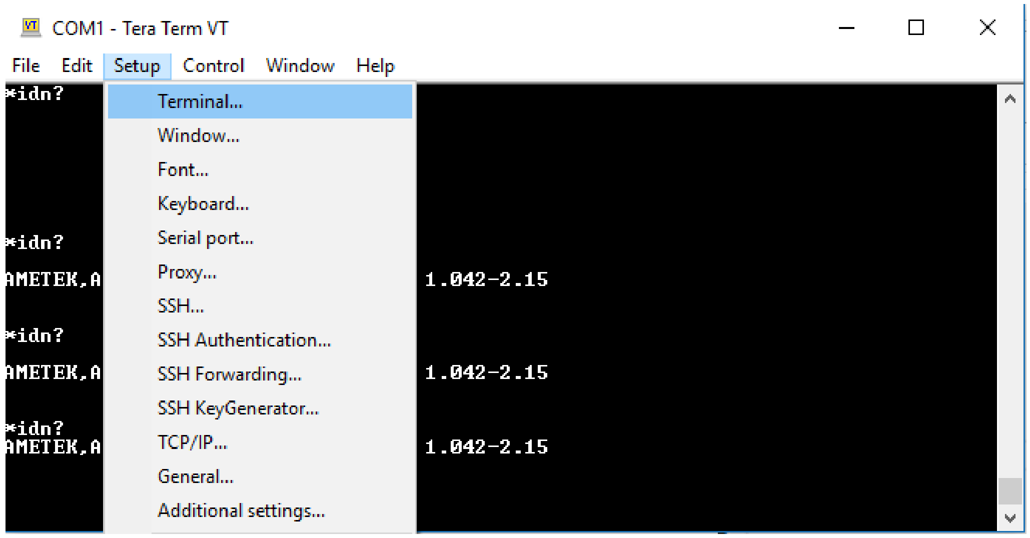


Figure 4-7. Terminal option in Tera Term

10. Change the settings as per the below screen shot. (Refer to Figure 4-8)

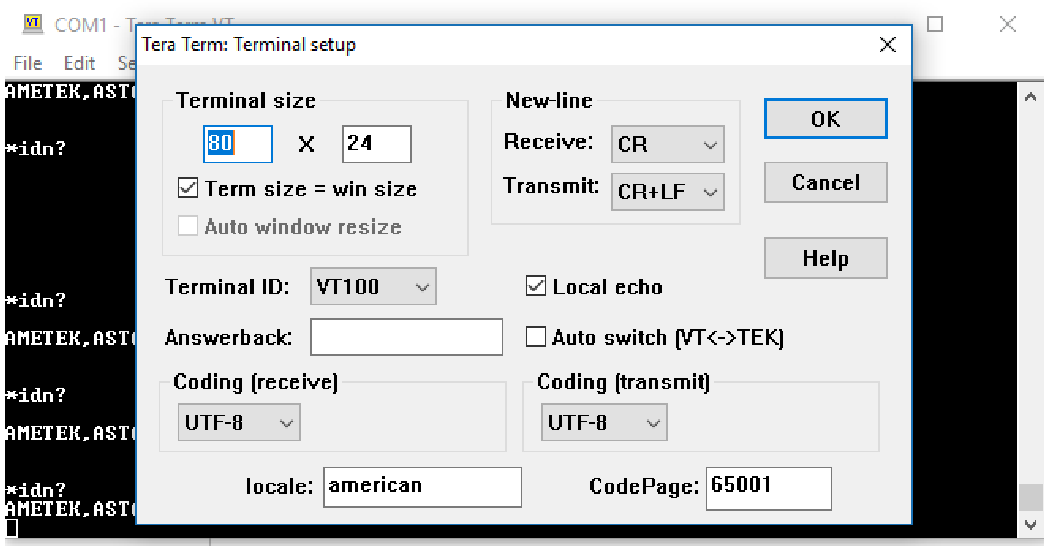


Figure 4-8. Terminal Setup in Tera Term

11. 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.

This page intentionally left blank.

5
USB CONFIGURATIONS
AND REMOTE PROGRAMMING

5.1 REAR PANEL

This section provides illustration of the Asterion DC power supply's rear panel layout, which differs among the Asterion DC models. For example, (Refer to Figure 5- 1). Regardless of the layout, the component functions are common across all models and those that are pertinent to the USB options are described here.

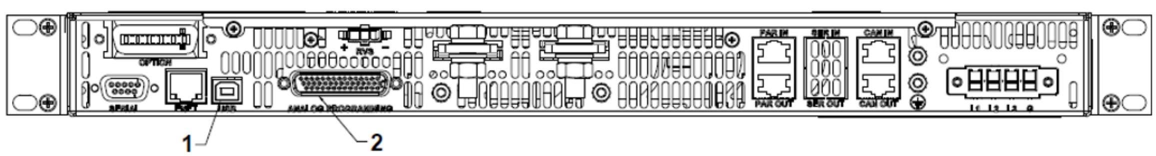


Figure 5- 1 Rear Panel – Type-B USB Interface

1-Type-B USB Connector

2- Remote Analog Programming and External User Control Interface connector

5.2 USB SETUP PROCEDURE

5.2.1 Port Number Identification

Port Number of the Asterion DC power supply can be identified in the device manager by following the steps below:

1. Navigate to device manager in the windows system.
2. Go to Ports and check for the new COM port available in the list. (Refer to Figure 5- 2)
3. Note: If Multiple COM ports are shown, disconnect all the COM ports and connect only the Asterion DC power supply and check for the new COM port.

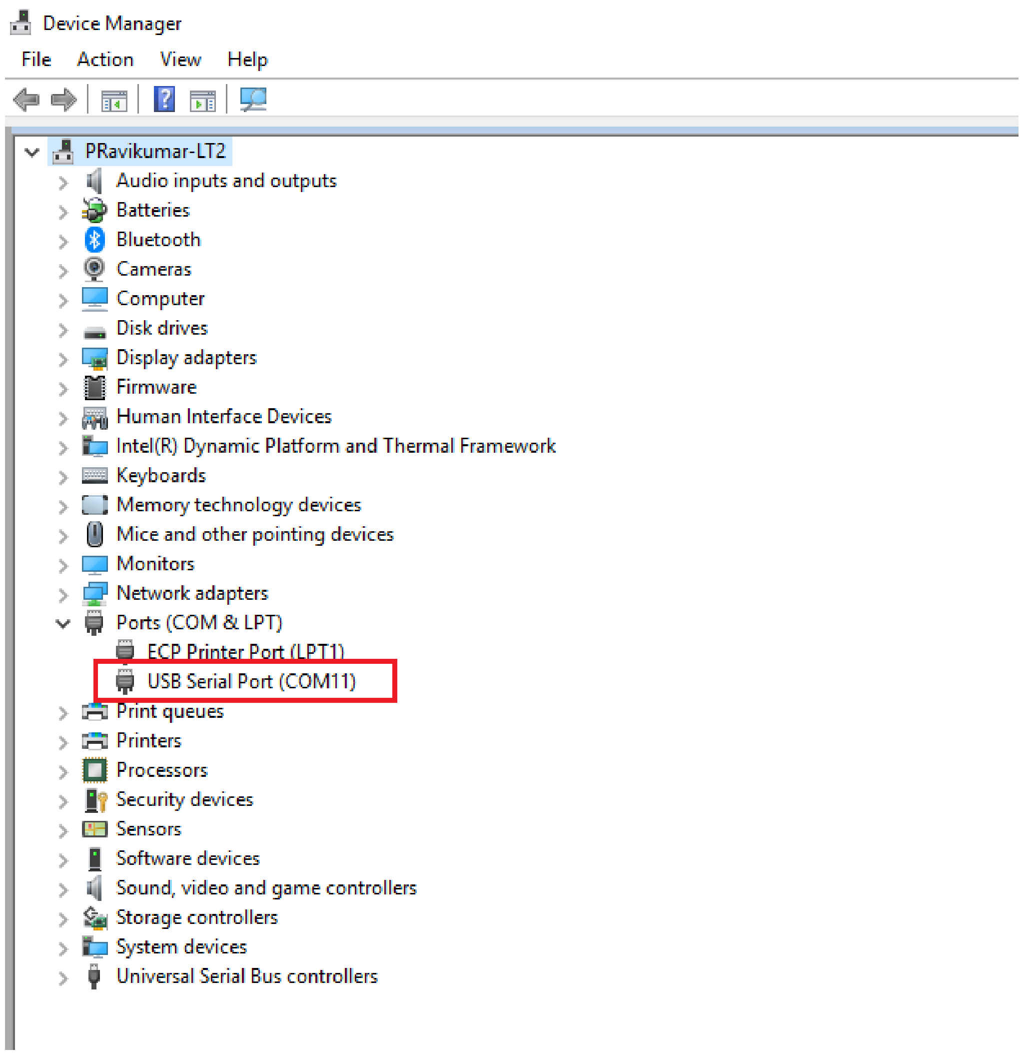


Figure 5- 2 Port identification

5.2.2 Communication test using serial interface program

1. Use one of the available programs for serial communication, such as Tera Term.

   If you choose to use Tera Term:

2. Select serial option in the Tera Term window and select the port identified in the device manager and press OK. (Refer to Figure 5- 3)

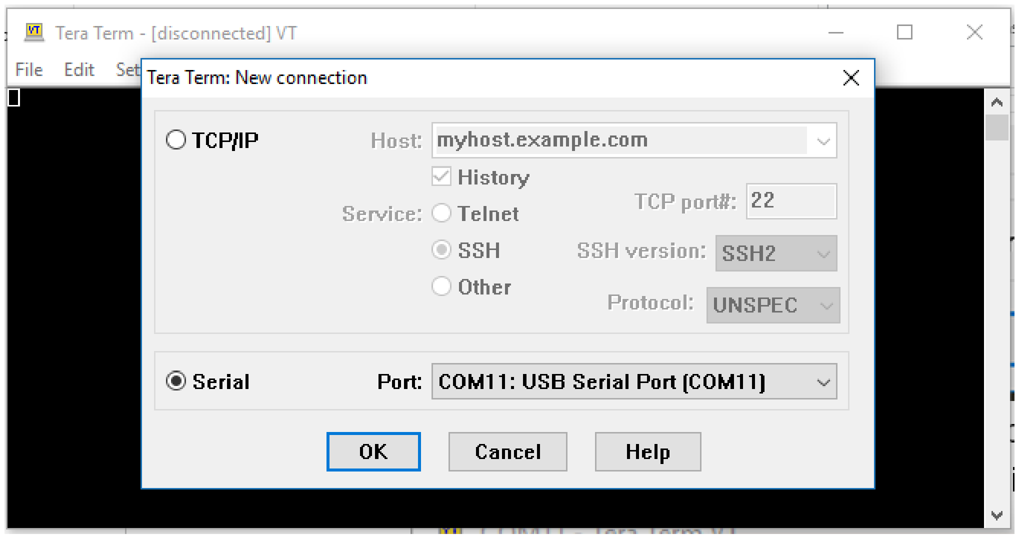


Figure 5- 3 Port identification in Tera Term

3. Select setup and select serial port in the setup window.

4. Set the baud rate to 115200, 8 data bits, no parity, 1 stop bit, and none flow control.

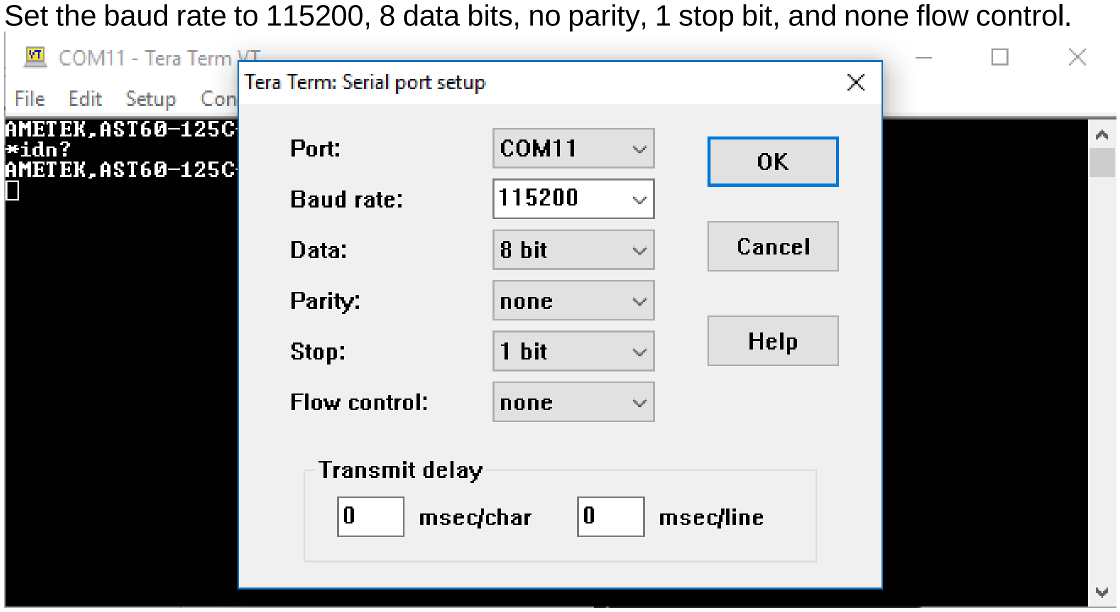


Figure 5- 4 Baud rate selection in Tera Term

5. Select the Terminal from the Tera Term. (Refer to Figure 5- 5)

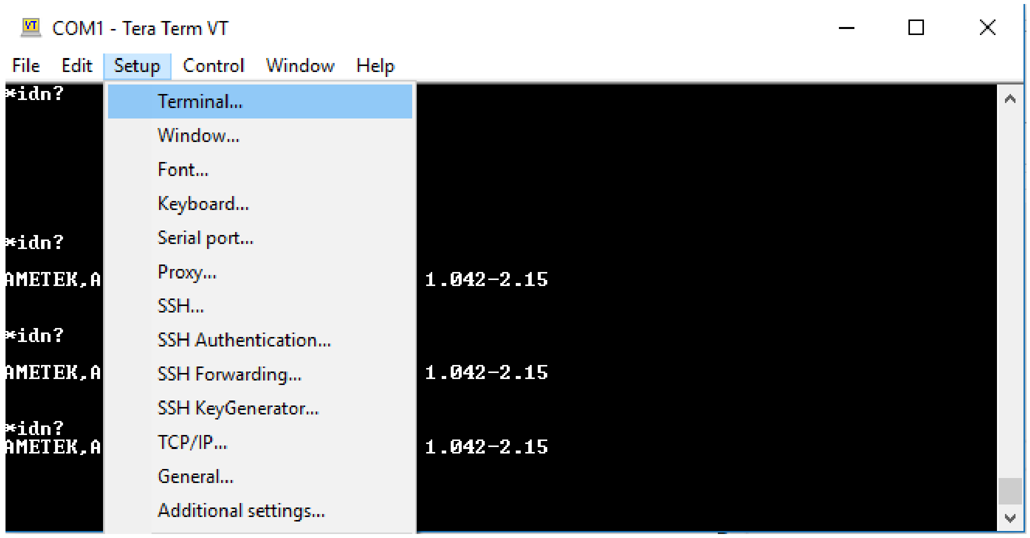


Figure 5- 5 Terminal selection in Tera Term

6. Change the settings. (Refer to Figure 5- 6)

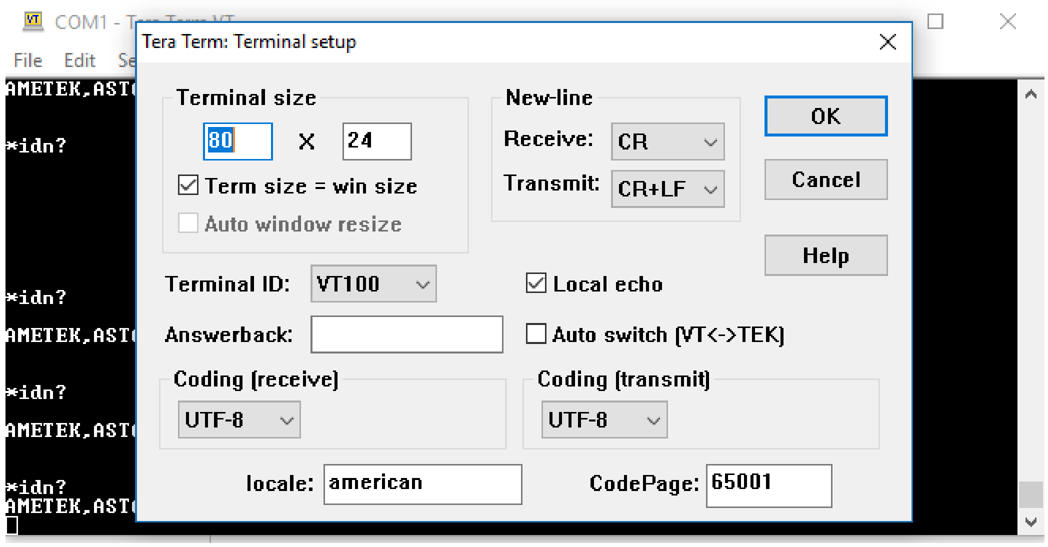


Figure 5- 6 Terminal setup in Tera Term

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.

This page intentionally left blank.

6 ETHERNET CONFIGURATION AND REMOTE PROGRAMMING

6.1 INTRODUCTION

This section covers the Remote Programming Ethernet Interface for the Asterion DC power supply. This configuration enables you to operate Asterion DC 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	AST<base model>-<last four digits of serial number>
Description	AST 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	52000

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 Asterion DC power supplies rear panel layout, which differs among the Asterion models. For example, refer to 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.

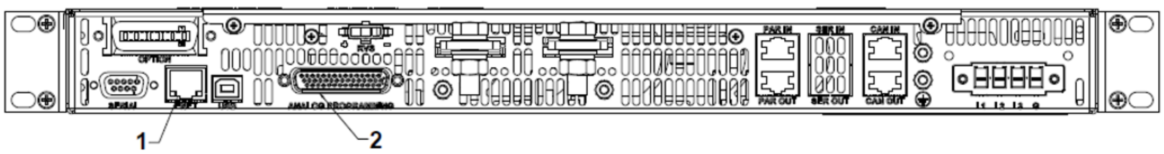


Figure 6-1. Rear Panel - Ethernet Interface

1-RJ45-Ethernet connector

2- Remote Analog Programming and External User Control Interface Connector

6.4 ETHERNET SETUP PROCEDURE

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

Network setup using DHCP Server

Network Setup using Auto-IP (Direct Connection between Asterion power supply 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 Asterion DC 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.

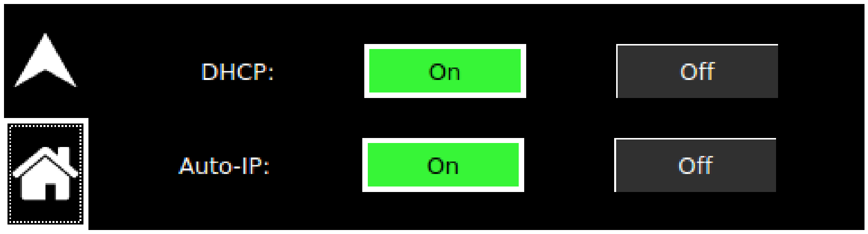


Figure 6-2. LAN Screen (Configure)

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.

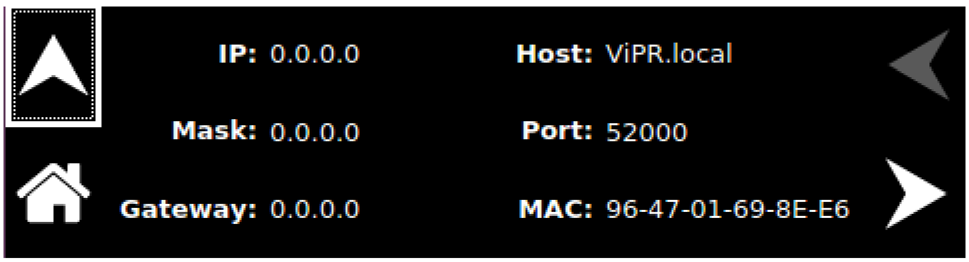


Figure 6-3. LAN Screen

6.4.1.2 DHCP MODE SELECTION USING SERIAL INTERFACE

1. Connect using a computer serial communications program such as Tera Term 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. Refer to Figure 6-4. LAN Screen (Settings).

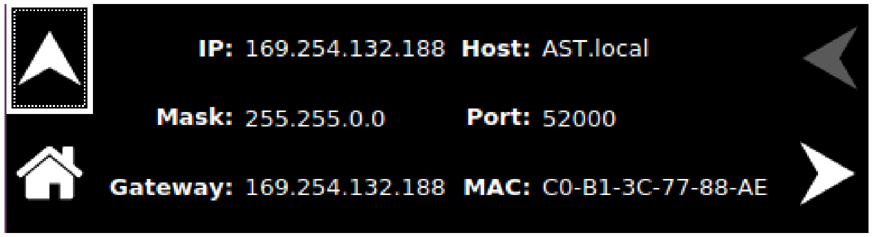


Figure 6-4. LAN Screen (Settings)

d. Using a computer serial communications program such as Tera Term, set for the same baud as Asterion DC power supply (Refer to 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.

Figure 6-5. RS232 Screen (Settings)

5. The Asterion power supply 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. Refer to Figure 6-10.

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

This setup requires that DHCP and Auto-IP is ON (Refer to 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, Asterion DC 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 Asterion DC 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. Refer to Figure 6-4.
   c. Using a computer serial communications program such as Tera Term, set for the same baud as Asterion DC 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 Asterion DC power supply 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. Refer to Figure 6-10.

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

Navigate to Home → Control Interface → LAN → LAN Configure. Make the DHCP as OFF, See Figure 6-2. LAN Screen (Configure). IP address button will be enabled.

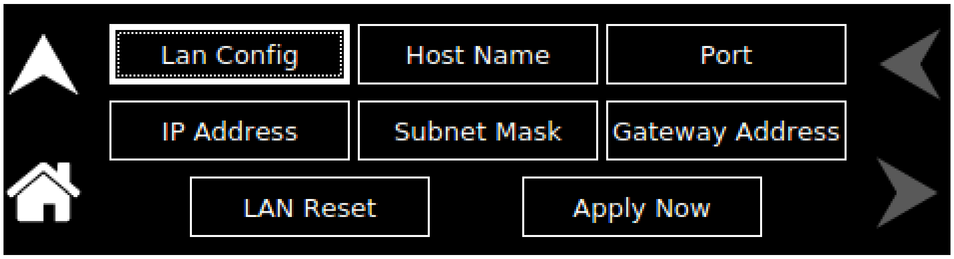


Figure 6-6. LAN configure (Settings)

Upon Pressing the IP Address, keypad to enter IP address will pop-up.

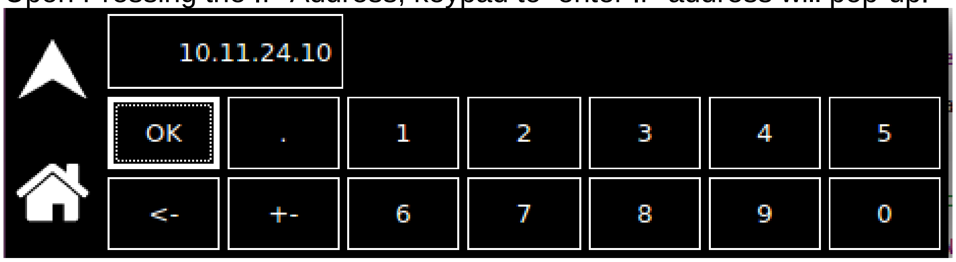


Figure 6-7. LAN Screen (Configure Static IP)

11. Enter the required IP address, Subnet Mask and Gateway (Refer to Figure 6-6).
12. Press "Apply" button for the settings to take effect.
13. The Asterion DC power supply 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. Refer to Figure 6-10.

6.4.3.2 STATIC IP SETUP USING THE SERIAL INTERFACE

1. Connect using a computer serial communications program such as Tera Term, set for the same baud as Asterion DC power supply (Refer to 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 Asterion DC power supply 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. Refer to Figure 6-10.

10. 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

6.4.4 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 = 52000, 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 6.5.2) or the RS232C interface.

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.

6.4.5 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.

6.4.6 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.

6.5 ETHERNET WEB PAGES, OVERVIEW

The layout of each of the Web pages includes the banner showing the Model (e.g AST 60/125), Manufacturer (AMETEK Programmable Power), AMETEK Logo and the Device name (e.g. LXI-ASTDC60-125D1C-E00). Below the Asterion DC power supply banner are four tabs, each linked to its corresponding page. Refer to Figure 6-8.

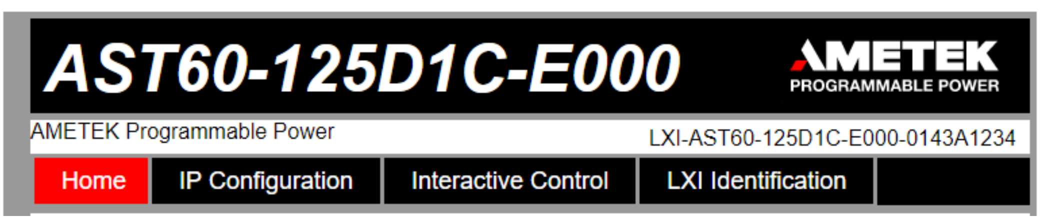


Figure 6-8. Asterion DC power Supply Banner and Tab

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. Refer to Figure 6-9.

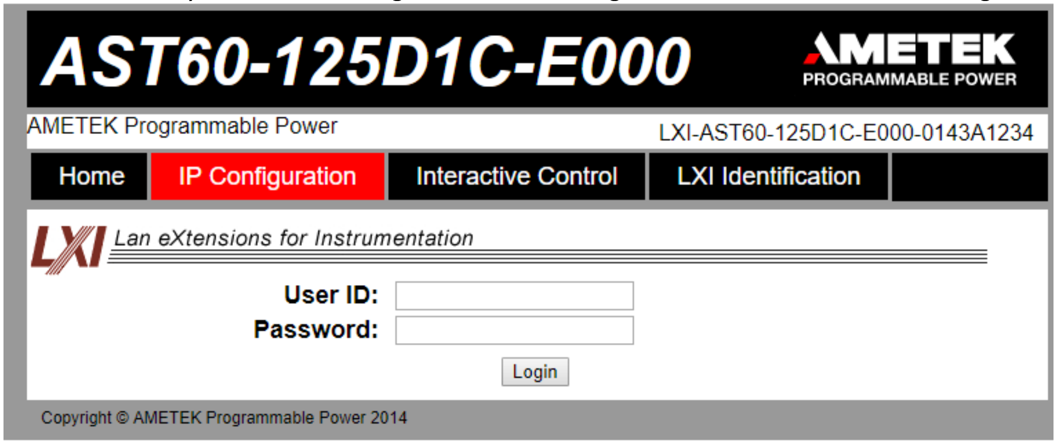


Figure 6-9. Login Window

6.5.1 Home

This is the default, information-only page (see Figure 6-10). It displays all the current information about the supply that you are connected to:

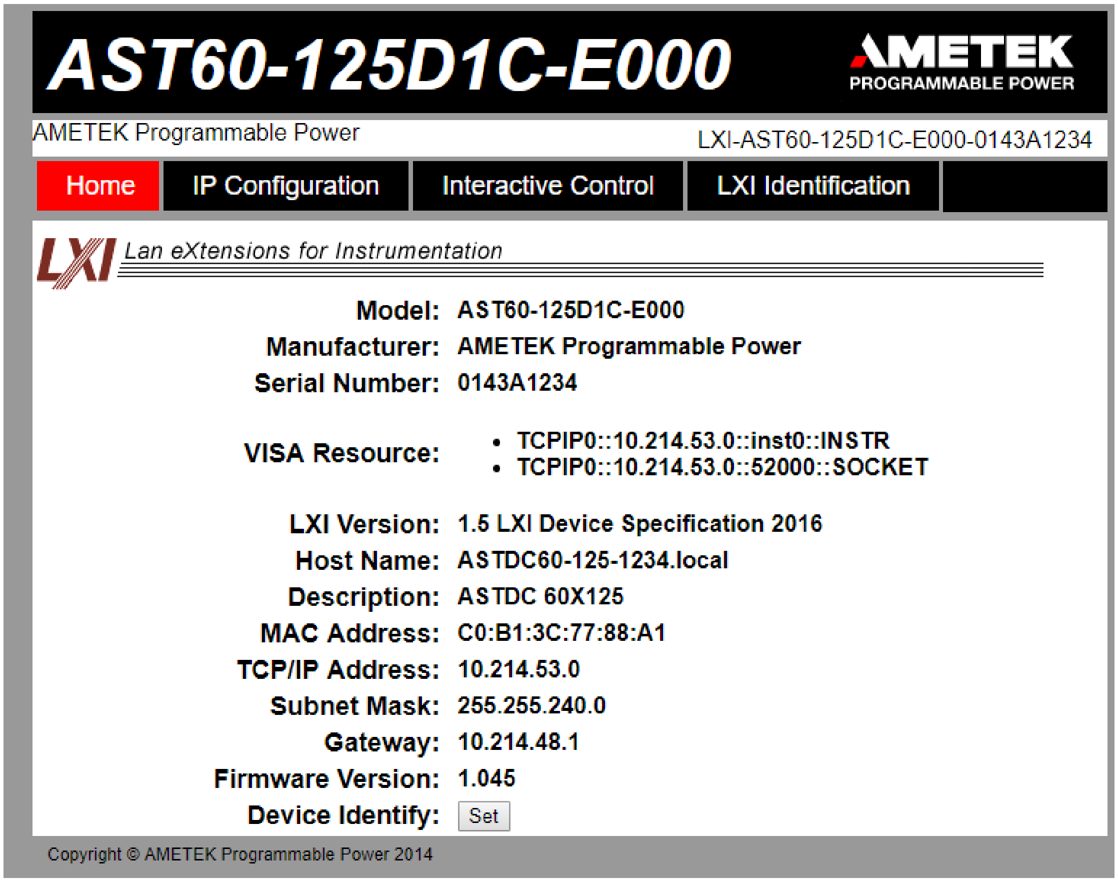


Figure 6-10. Asterion DC power Supply Home Page

- The **Model** number, the **Manufacturer**, and the **Serial Number** of your Asterion DC 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**: When set button is pressed the following screen will appear. Refer to Figure 6-11

Figure 6-11. Asterion DC Power supply Device identify

6.5.2 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 (Refer to Figure 6-12).

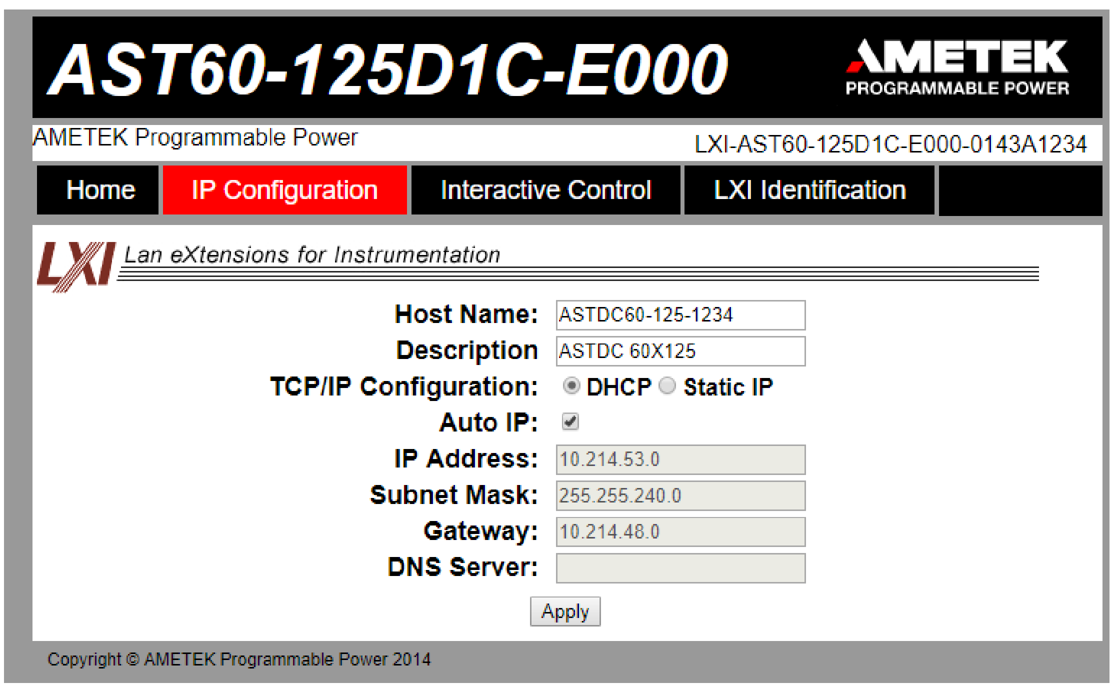


Figure 6-12. Asterion DC power supply IP Configuration Page

Host Name: the default host name is **ASTDCV-I-SI:No**. For eg: for 60V/125A device; the default host name is ASTDC60-125-1234.

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. Refer to Figure 6-10. 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) (Refer to Figure 6-11)

6.5.3 Interactive Control

This web page allows to input a properly formatted SCPI command (refer to Section 10). Click on Send Command button to send the command to the Asterion DC power Supply. The commands and response to the query command can be seen on the web page (Refer to Figure 6-13).

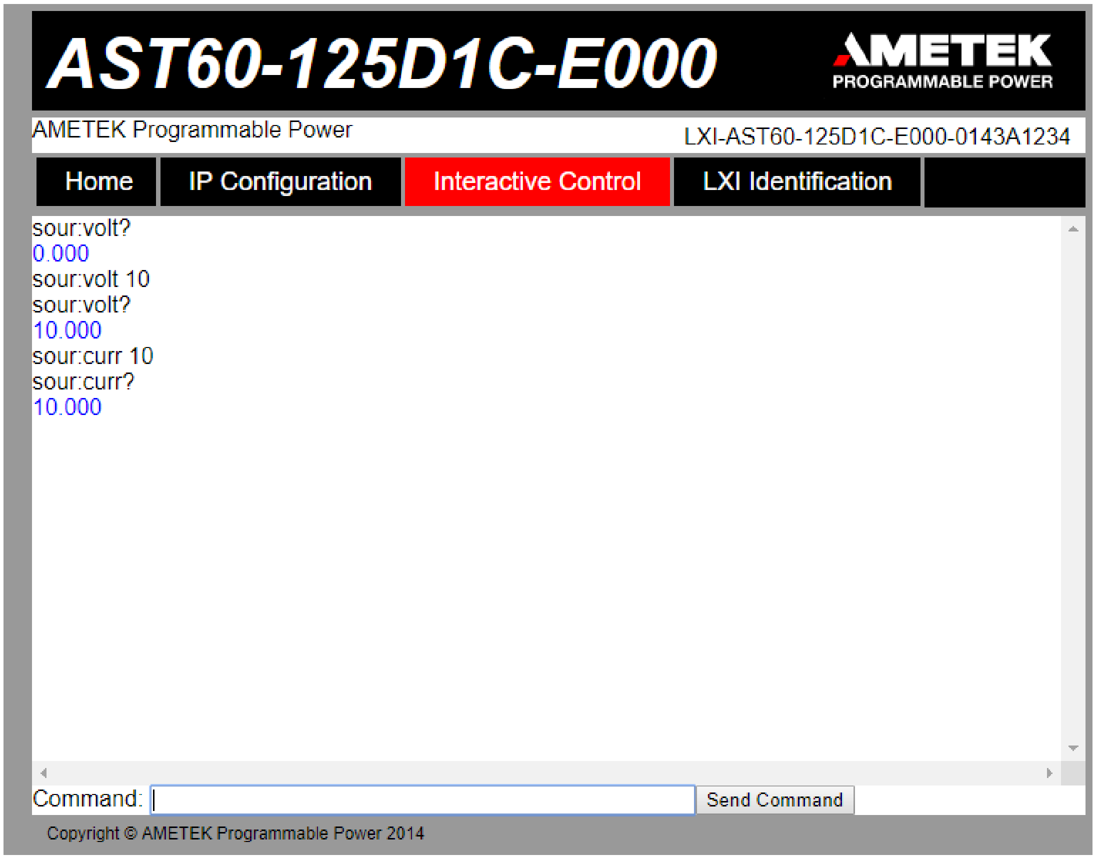


Figure 6-13. Interactive Control Page

6.5.4 LXI Identification

The LXI Identification web page displays the LXI parameters of the Asterion DC Power Supply (Refer to Figure 6-14).

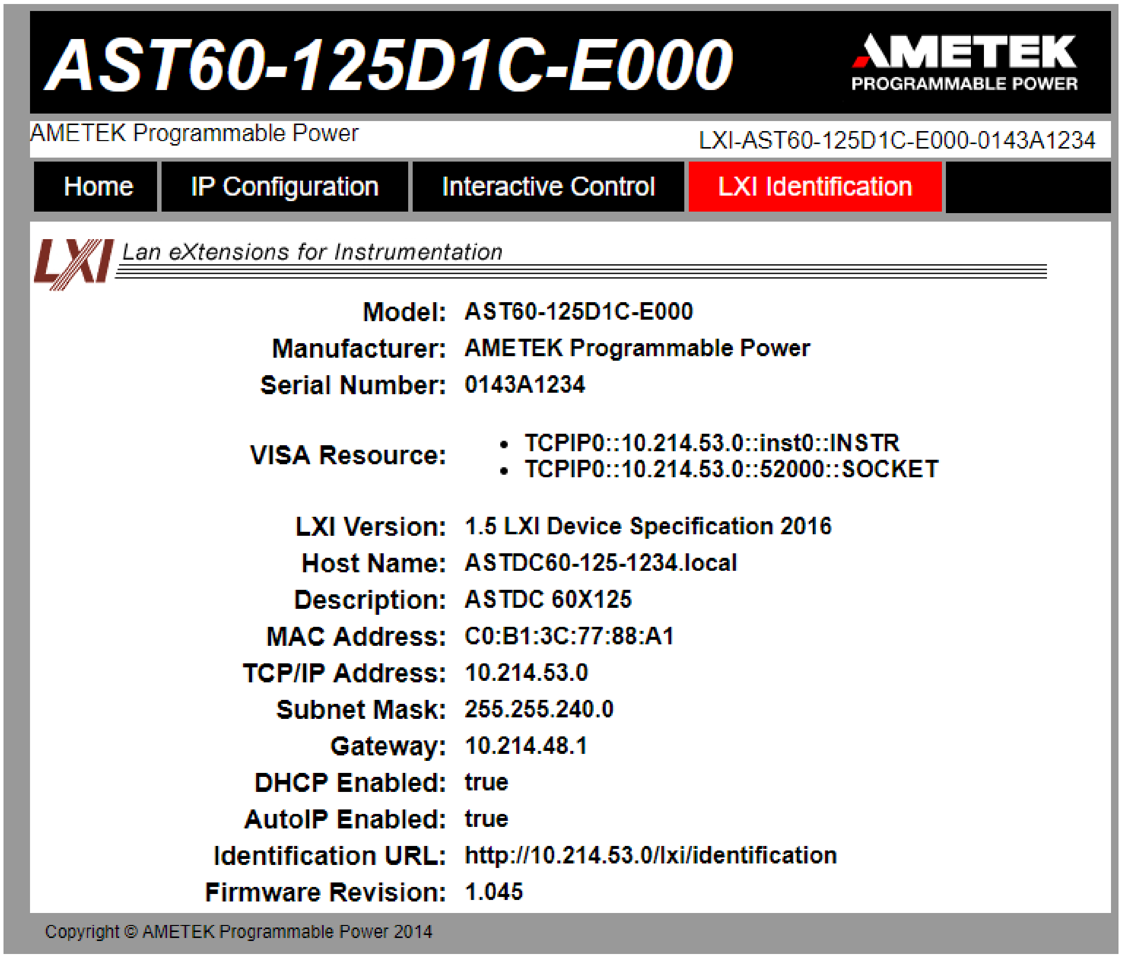


Figure 6-14. LXI Identification Page

This page intentionally left blank.

7
ETHERCAT CONFIGURATIONS AND REMOTE PROGRAMMING

7.1 INTRODUCTION

This section covers the Remote Programming EtherCAT Interface (Optional) for the Asterion DC power supply. This configuration enables you to operate Asterion DC 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.

For connecting EtherCAT devices only Ethernet cables that meet at least the requirements of category 5 (CAT5) according to EN 50173 or ISO/IEC 11801 should be used.

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

7.2 REAR PANEL

This section provides illustration of the Asterion DC power supply's rear panel layout, which differs among the Asterion DC models. For example, refer to Figure 7-1, Regardless of the layout, the component functions are common across all models and those that are pertinent to the EtherCAT options are described here.

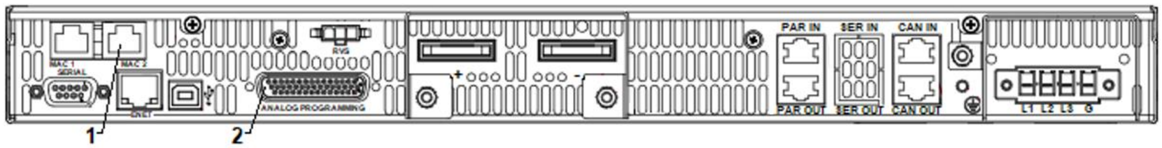


Figure 7-1: Rear Panel – EtherCAT Interface

1- Two RJ45-Ethernet connector

2- Remote Analog Programming and External User Control Interface Connector

7.3 ETHERCAT SETUP PROCEDURE

The following subsections that follow are the setup procedure that applies to your system and application to configure the EtherCAT.

7.3.1 Procedure to Setup the EtherCAT

1. Build an EtherCAT communications cable as per the pinout description illustrated in the table below:

Pin #	EtherCAT Signal	Core Coloring	Description
1	TD +	Yellow	Transmit Data +
2	TD -	Orange	Transmit Data -
3	RD +	White	Receiver Data +
6	RD -	Blue	Receiver Data 0

2. Connect power to the unit and turn on the unit.
3. Using front panel, navigate to EtherCAT Interface screen, go to Home → Control Interface → EtherCAT → Enable.

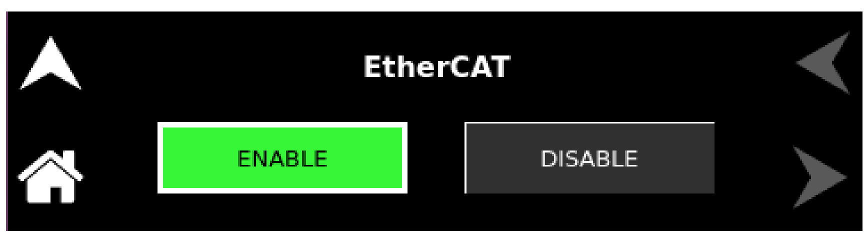


Figure 7-2: EtherCAT Enable / Disable Screen

4. Open Command prompt as administrator to location “C:\TwinCAT\3.1\System”cmd

```
Command Prompt
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vsubramani>cd ../../

C:\>cd TwinCAT\3.1\System

C:\TwinCAT\3.1\System>
```

Figure 7-3: Configure EtherCAT

5. Run the Batch file “win8settick.bat” as administrator from the location “C:\TwinCAT\3.1\System” in Command prompt.

```
C:\TwinCAT\3.1\System>win8settick.bat

C:\TwinCAT\3.1\System>bcdedit /set UseLegacyApicMode yes
The operation completed successfully.

C:\TwinCAT\3.1\System>bcdedit /set disabledynamictick yes
The operation completed successfully.

C:\TwinCAT\3.1\System>bcdedit /set useplatformtick yes
The operation completed successfully.

C:\TwinCAT\3.1\System>
```

Figure 7-4: Run Batch File

6. Restart the System.

7.3.2 Installing the Required Drivers

1. Connect EtherCAT and the Windows PC via Ethernet cable.
2. Open TwinCAT XAE Shell.
3. Select File → open → project/Solution option and select TwinCAT-ViPR (XML project)
4. Select “TwinCAT → “Show real time Ethernet Compatible Devices” option.
5. Select the Ethernet adapter connected to the EtherCAT device from the list. And Select “Install”. It will install Beckhoff driver for compatibility for EtherCAT. Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

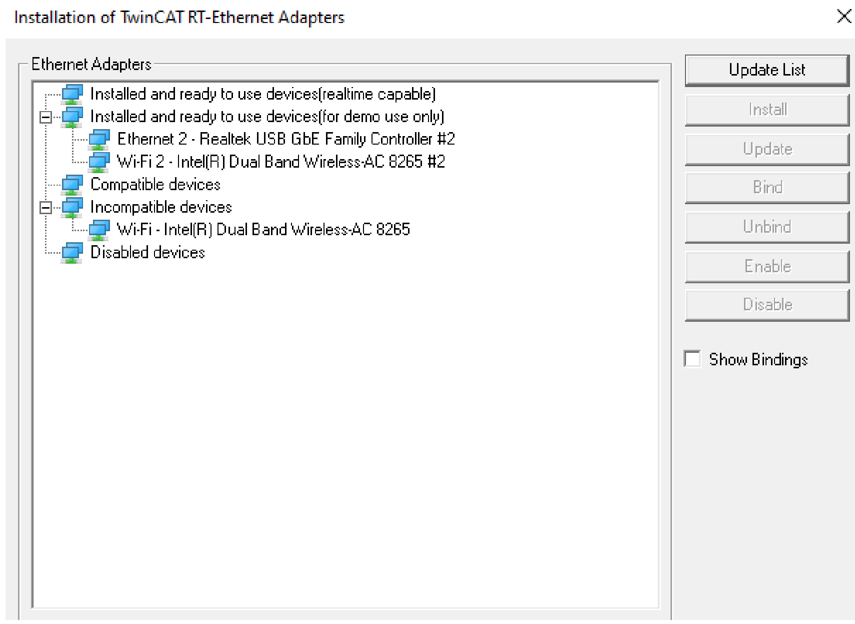


Figure 7-5: Install EtherCAT Compatibility Drivers

7.3.3 To Program the EtherCAT using SSC (Slave Stack Code) Tool

1. Open SSC Tool. Upon opening the SSC tool, following window will pop up and press ok.

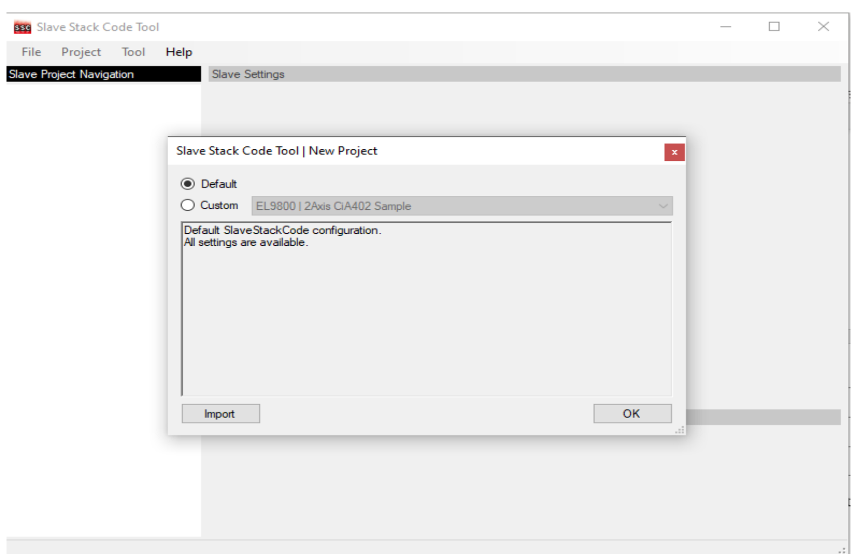


Figure 7-6: SSC Tool Launch

2. Navigate to Tool → EEPROM Programmer.

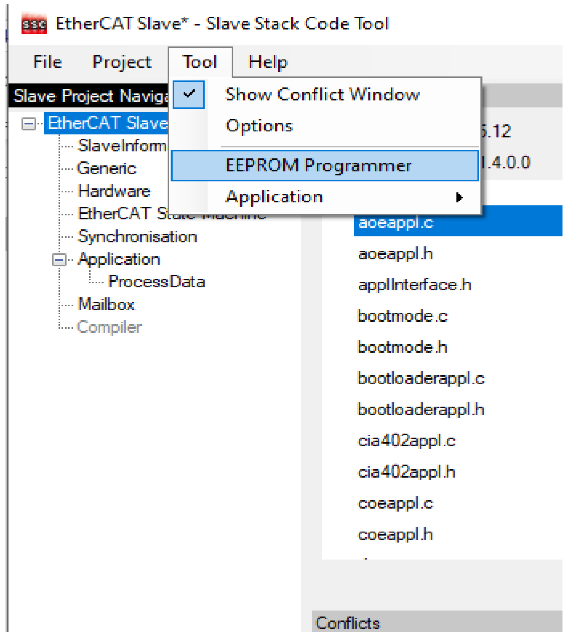


Figure 7-7: Open EEPROM Programmer

3. EEPROM Programmer will be opened, click on File → open and select the XML file.

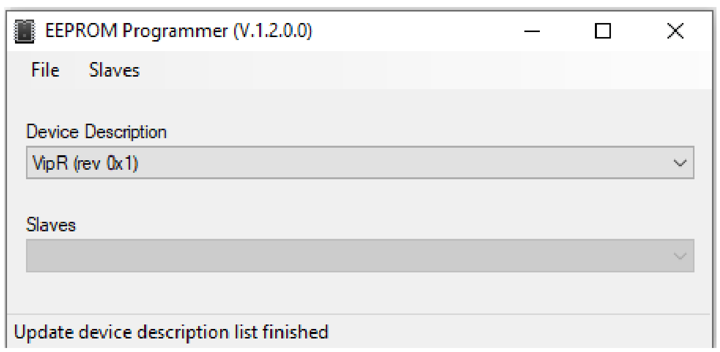


Figure 7-8: XML project selection

4. Click on Slaves → scan option.

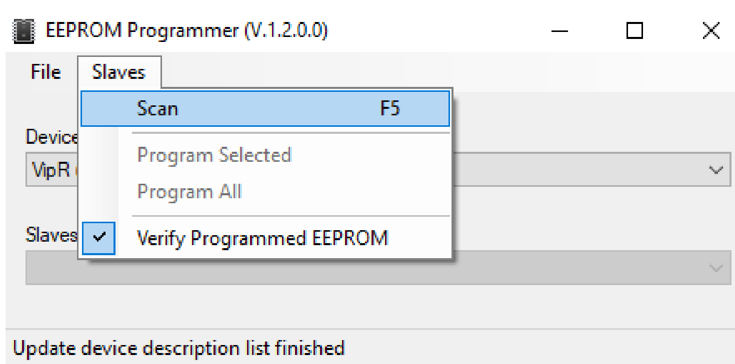


Figure 7-9: Scan for EtherCAT devices

5. List of ethernet option will be listed and select the ethernet port to which the EtherCAT board is connected.

Figure 7-10: List of the Ethernet Devices

6. To find the appropriate ethernet port, navigate to ethernet settings of the system. In the below example Ethernet (4) is connected to EtherCAT Board.

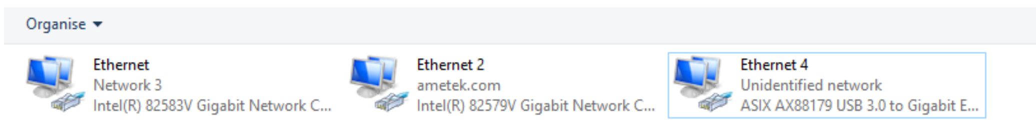


Figure 7-11: Ethernet Devices

7. It will scan for the EtherCAT device and once found, the following window is displayed. This window ensures that the EtherCAT is connected.

Figure 7-12: Found EtherCAT Devices

8. Select Slaves → Program Selected, it will start to program the device.

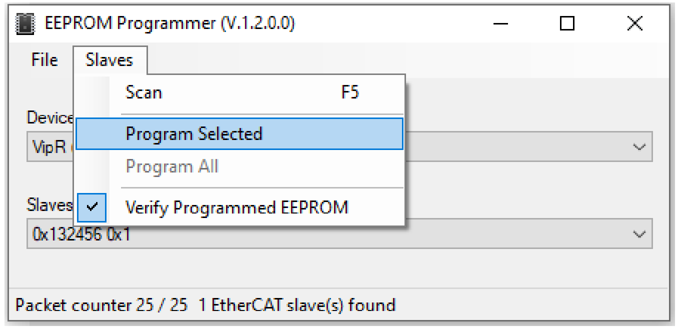


Figure 7-13: Program EtherCAT device

9. During programming the following window is displayed, refer to Figure 7-14 and once the programming is completed, refer to Figure 7-15.

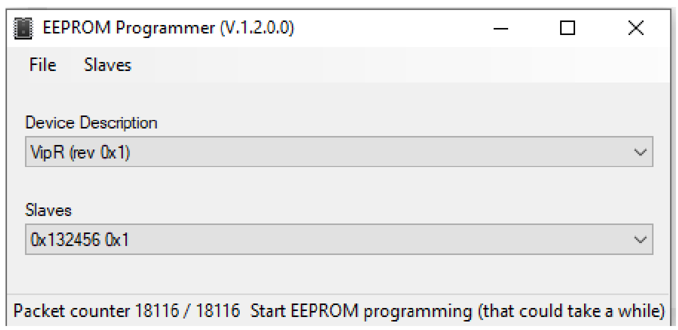


Figure 7-14: EEPROM Programming

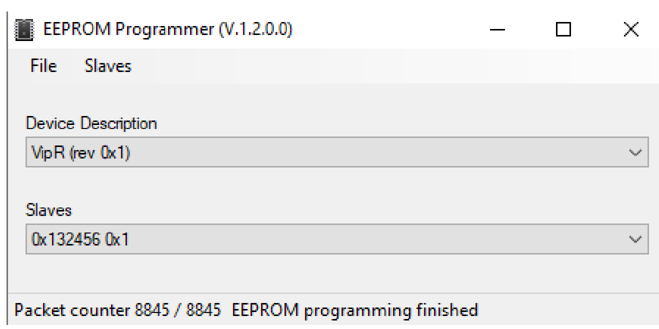


Figure 7-15: EEPROM Programming Completed

7.3.4 TwinCAT XAE Shell

1. Open TwinCAT XAE Shell tool.
2. Select File → Open → Project/Solution option and select the TwinCAT-ViPR.sln (XML Project).

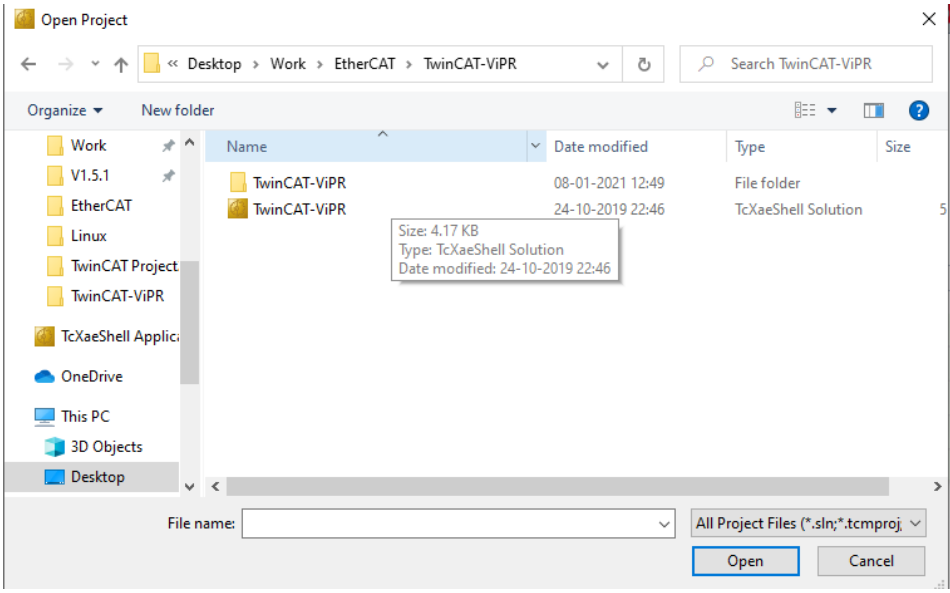


Figure 7-16: XML project selection

3. Once project is opened, Select opened TwinCAT project. After turning on the power to the unit, select I/O → Devices → Devices[n] → Box (ViPR). EtherCAT Device will be Scanned. Incase if Device is not scanned properly refer to section 4 and repeat the steps.

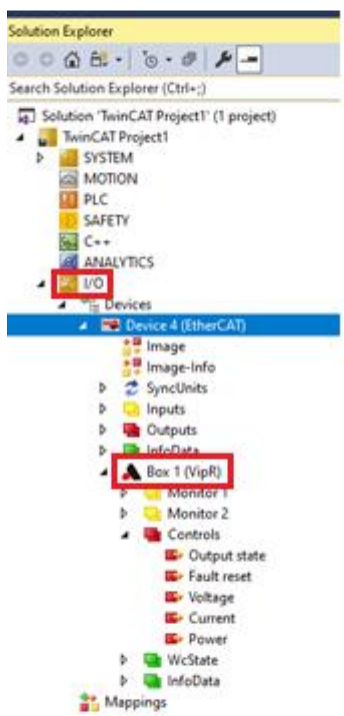


Figure 7-17: Scanned EtherCAT Devices

4. If it is first time for the mapping to be done. Double click on standard inputs variable and choose the relevant field and map it. For example, voltage monitor should correspond to voltage measure, refer to Figure 7-18.

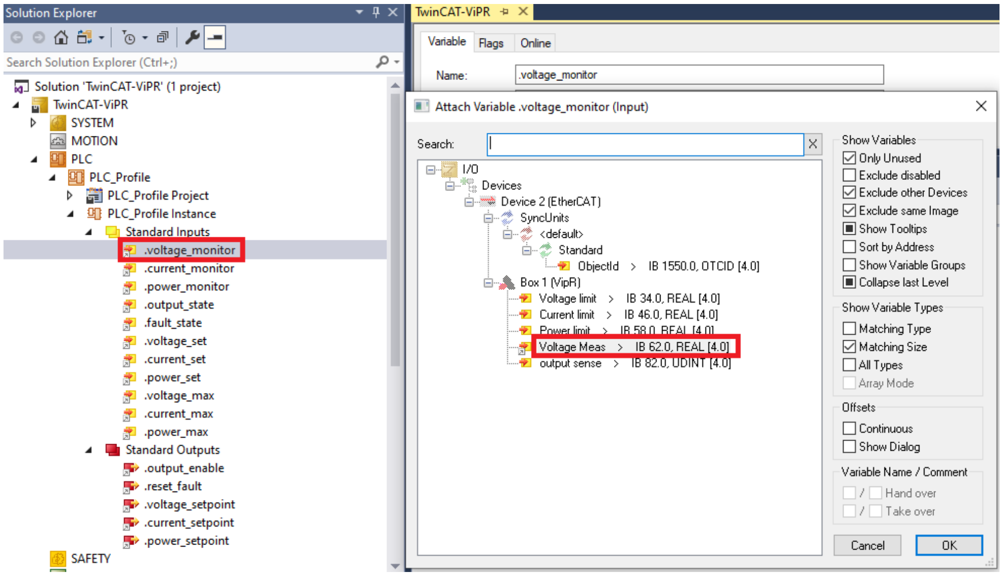


Figure 7-18: Mapping of variables

5. In Menu select TwinCAT → Activate Configuration option.

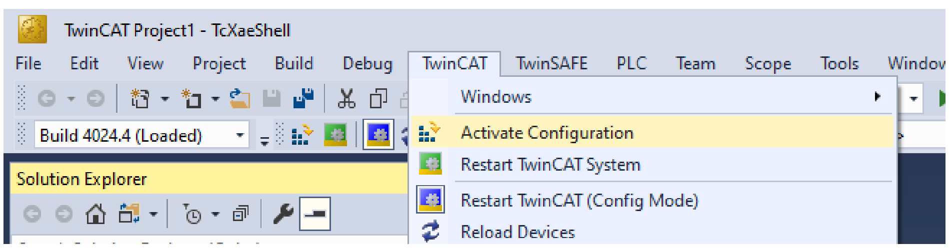


Figure 7-19: Activate Configuraiton

6. Select OK option.

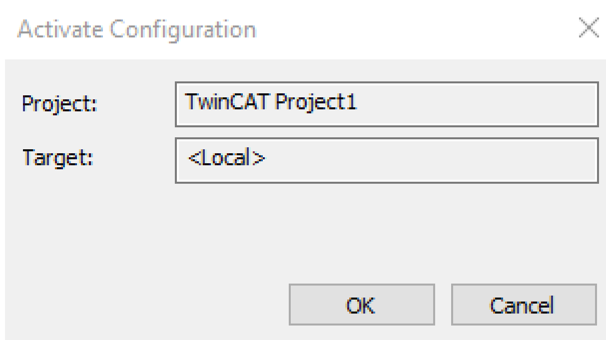


Figure 7-20: Activating Configuration Window

7. Select Continue option.

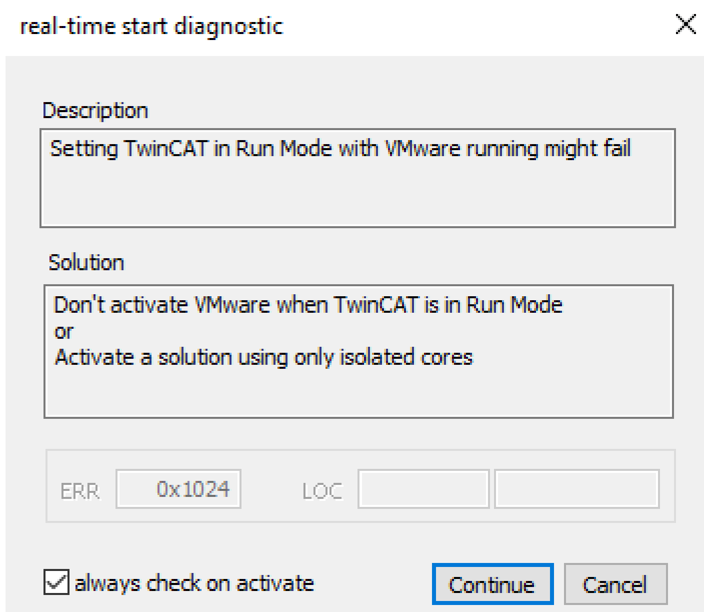


Figure 7-21: Real-time Start Diagnostic Window

8. Select OK option.

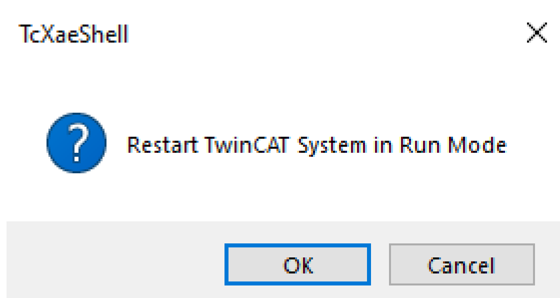


Figure 7-22: Restart TwinCAT in Run Mode

9. Always check if ADS port is set to 801. Navigate to PLC Profile and right-click to access the drop-down menu, select “Change ADS Port”.

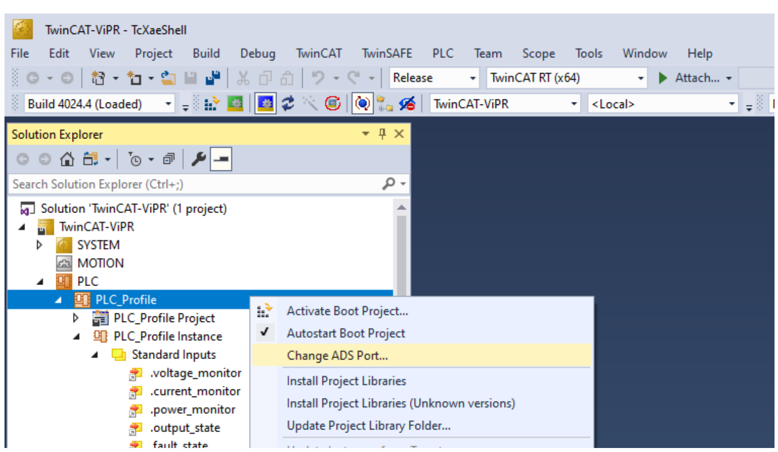


Figure 7-23: Navigation to ADS port

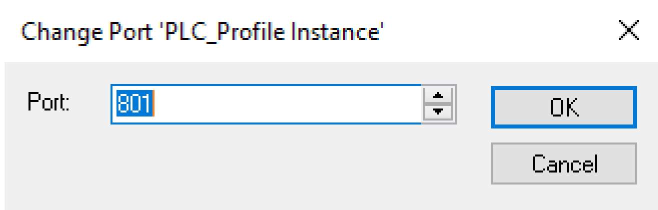


Figure 7-24: Verifying ADS port

10. Select option PLC → login.
11. Incase if 801 port doesn't exist you will receive Dialog box asking "801 port doesn't exist do you want to create and proceed" Select option "Yes"
12. Select option PLC → Start.
13. Select ViPR project → I/O → Devices → Device [n] → Box (ViPR). Select Monitor options to monitor the output of the unit and verify if it is updating correctly.

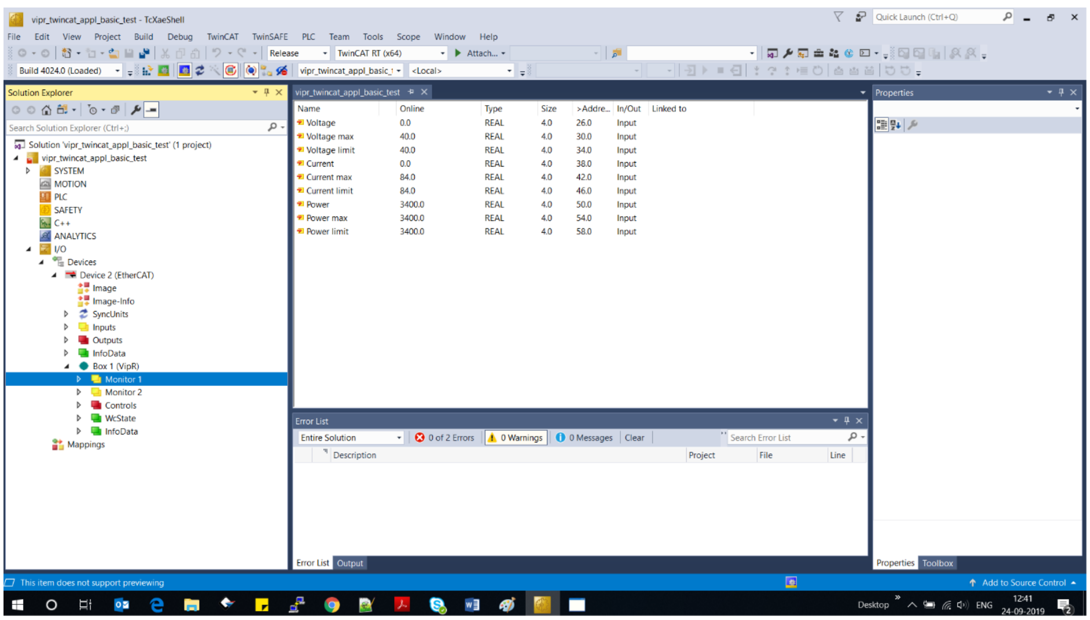


Figure 7-25: Monitor screen

14. Start ViPR Unit Monitor, select EtherCAT option and connect.

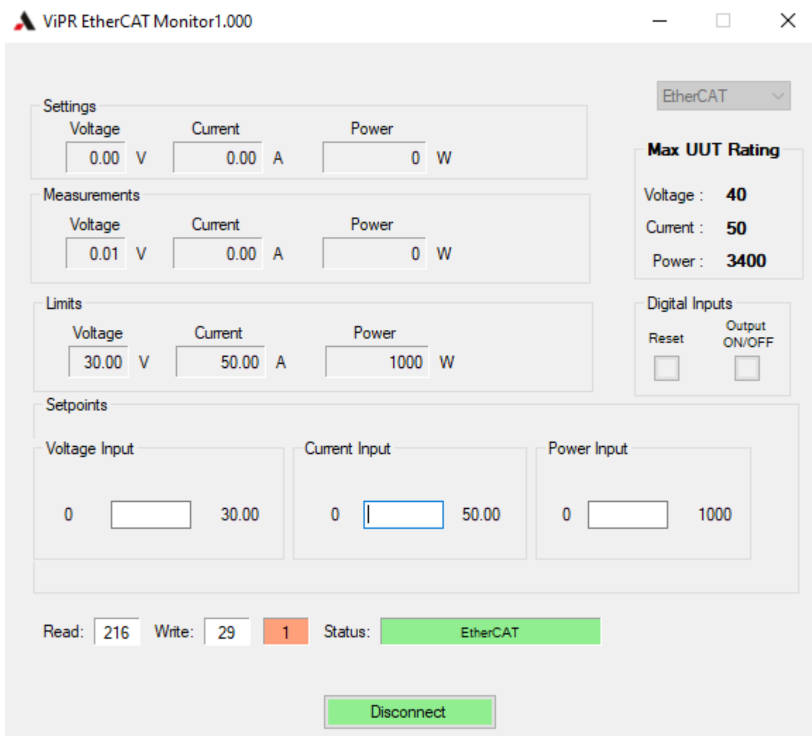


Figure 7-26: ViPR Unit Monitor

8
IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING

8.1 REAR PANEL

This section provides illustrations of the Asterion DC power supply's rear panel layout, which differs among the Asterion DC models. For example, see Figure 8-1. Regardless of the layout, the component functions are common across all models, and those that are pertinent to the IEEE488.2 GPIB(Optional) options are described here.

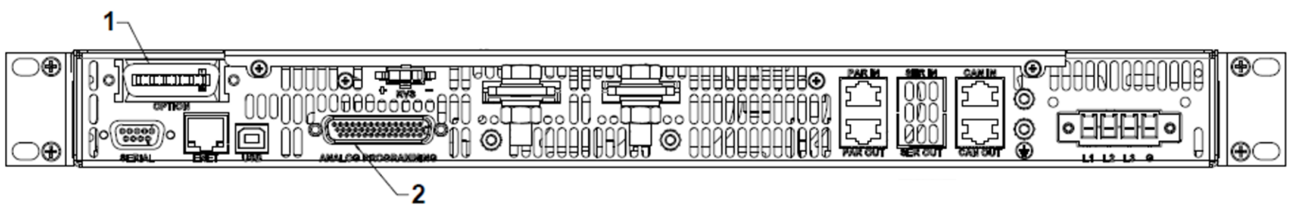


Figure 8-1. Rear Panel – IEEE 488.2 GPIB Interface

1- IEEE 488.2 GPIB Connector
2- Remote Analog Programming External User Control Interface connector

8.2 REMOTE PROGRAMMING VIA IEEE488.2 GPIB CONNECTOR

8.2.1 Address Selection

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

Asterion DC power supply's GPIB address can be configured by using the front panel menu, Refer to Figure 8-2. To navigate to GPIB screen, go to Home → Control Interface → GPIB.

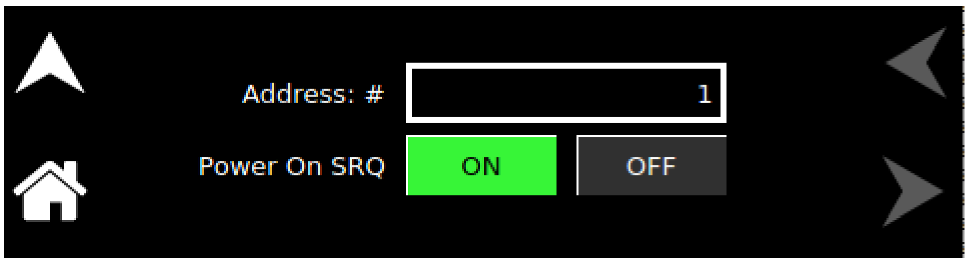


Figure 8-2. GPIB Screen

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

PON SRQ can be enable using the front panel menu, see Figure 8-2. When PON SRQ is enabled, a GPIB service request will be sent by the power supply to the computer controller on Power ON.

8.2.3 Shield Ground

Connects GPIB cable shield to chassis ground.

8.2.4 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 8-2.
3. Connect GPIB cable from the controlling computer to the Asterion DC power supply's.
   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.

This page intentionally left blank.

9
REMOTE ANALOG PROGRAMMING AND EXTERNAL USER CONTROL INTERFACE CONNECTOR

9.1 INTRODUCTION

This section provides illustrations of the Asterion DC power supply's rear panel layout, which differs among the Asterion DC models. For example, Refer to Figure 9-1. Regardless of the layout, the component functions are common across all models and those that are pertinent to the Remote Analog Programming and External user control Interface Connector is described here. The analog programming and external user interface control functionality can be exercised from either of the communication interfaces (RS232, USB, Ethernet, IEEE 488.2 GPIB and EtherCAT).

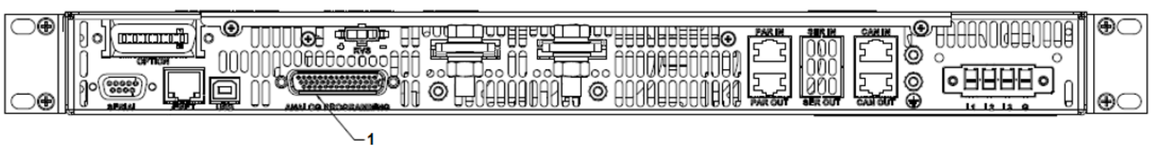


Figure 9-1. Rear Panel of Asterion DC 1U – 44 pin Analog Programming Connector

1: Remote Analog Programming and External User Control Interface Connector

9.2 REMOTE ANALOG PROGRAMMING CONNECTOR DETAILS

A 44-pin Subminiature-D connector (refer to Figure 9-2) located at the rear panel provides remote analog programming and external user interface control signals to increase the user's operating control of the supply. The mating receptacle is NORCOMP 180-044-102L001 with 44 Pin male terminals. The connector terminals accommodate wire sizes from #24 - #30.

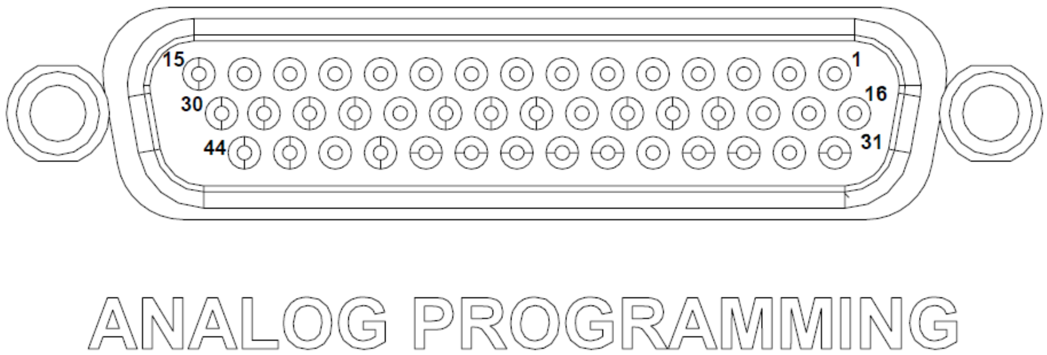


Figure 9-2 Remote Analog Programming and External User Control Interface Connector Pin-Out

Pin	Signal	Type	Description
1	ISO HV ON/OFF	ISO HV INPUT	Isolated remote-control input for output on/off with an applied AC/DC voltage source. A positive (+) 6-120 VDC or an AC input of 12- 240 VAC will enable (turn-on) the output of the supply. This control input is optically isolated from the output power negative terminal of the power supply (up to 500 VDC). Signal return: Pin 16 (ISO_RTN), Circuit is electrically isolated from the output power negative terminal
2	Not used	N/A	Not used
3	VPRG_ISOUR	ANALOG INPUT	Remote control input for voltage programming using a resistance connected between pin 3 and signal return. Current Source of 1 mA is internally connected to this pin to enable resistance programming. Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 kΩ to user selectable maximum range (2 kΩ to 10 kΩ) for 0 to full scale rated Output **Note:** Do not exceed resistance of maximum 20 kΩ
4	VPRG_VSOUR/ VPRG_4-20mA	ANALOG INPUT	Remote control input for voltage programming using a voltage source connected between pin 4 and signal return. Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 V to user selectable maximum range (2 V to 10 V) for 0 to full scale rated Output Note: Do not exceed an input of 25 VDC

			This remote-control input can be configured to take 4- 20 mA current input for output voltage programming. Uses same signal return pin 18 for 4- 20 mA input type also. Range: 4 mA to 20 mA for 0 to full scale rated Output Input impedance: 500 Ω, typical
5	OVPPRG_VSOUR	ANALOG INPUT	Remote control input for overvoltage programming using a voltage source connected between pin 5 and signal return. Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0-10VDC = 0-110% of the full-scale Output Voltage. Note: Do not exceed an input of 12 VDC
6	VMON	ANALOG OUTPUT	Monitor signal for output voltage Signal return: Pin 22, Circuit is electrically connected to the output power negative terminal Range: 0 V to 10 V corresponds to 0-100% full-scale output. Minimum recommended Load: 100 kΩ, typical Maximum Load: 20 kΩ
7	DIO_IN2	DIGITAL INPUT	User digital input, function to be assigned by user. Signal return: Pin 12, Circuit is electrically connected to the output power negative terminal Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.
8	TRIGGER_IN	DIGITAL INPUT	User digital Input signal, TTL active-high; provides external hardware triggering of voltage and current ramp functions. Signal return: Pin 12, Circuit is electrically connected to the output power negative terminal Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.
9	DIO_IN1	DIGITAL INPUT	User digital input, function to be assigned by user. Signal return: Pin 12, Circuit is electrically connected to the output power negative terminal Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.

10	REV_RY_EN	DIGITAL INPUT	User digital input, enables high output on pin 38. Signal return: Pin 12, Circuit is electrically connected to the output power negative terminal Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.
11	OUT_RY_EN	DIGITAL INPUT	User digital input, enables high output on pin 37. Signal return: Pin 12, Circuit is electrically connected to the output power negative terminal Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.
12	DIN_RTN*	RETURN FOR DIGITAL INPUTS	Return/GND. Internally, pins 12, 14, 15,18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
13	Remote ON/OFF / ESTOP	ON/OFF control INPUT	Control input for Output ON/OFF. Switch/Relay contact closure or direct short-circuit from pin 13 to signal return (pin 14) will enable (turn-on) the output of the supply Internally ties to 5V through a 2.49kΩ resistor Signal return: Pin 14, Circuit is electrically connected to the output power negative terminal
14	Remote ON/OFF_RTN*	RETURN FOR REMOTE ON/OFF	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential.
15	USER_POWER_RTN*	RETURN FOR USER POWER INPUT	Return/GND. Internally, pins 12, 14, 15 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential.
16	ISO_RTN / RTN_HV	RETURN FOR REMOTE ISO ON/OFF	Dedicated Return for pins 1 and 31 (ISO HV ON/OFF and ISOTTLON/OFF).
17	Not used	N/A	Not used
18	EXT_PRG_RTN*	RETURN FOR ANALOG INPUTS	Return/GND. Internally, pins 12, 14, 15 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
19	IPRG_VSOUR / IPRG_4-20mA_SOUR	ANALOG INPUT	Remote control input for current programming using a voltage source connected between pin 19 and signal return pin 18.

			Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 V to user selectable maximum range (2 V to 10 V) for 0 to full scale rated Output Note: Do not exceed an input of 12 VDC This remote-control input can be configured to take 4- 20 mA current input for current programming. Uses same signal return pin 18 for 4- 20 mA input type also. Fixed range from 4 mA to 20 mA for 0 to full scale rated Output
20	IPRG_RES / IPRG_ISOUR	ANALOG INPUT	Remote control input for current programming using a resistance connected between pin 20 and signal return. Current source of 1 mA for remote current programming using a resistance connected to signal return Pin 18 Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 kΩ to user selectable maximum range (2 kΩ to 10 kΩ) for 0 to full scale rated Output Note: Do not exceed resistance of maximum 20 kΩ
21	IMON	ANALOG OUTPUT	Monitor signal for output current Signal return: Pin 22, Circuit is electrically connected to the output power negative terminal Range: 0 V to 10 V corresponds to 0-100% full-scale output. Minimum recommended Load: 100 kΩ, typical Maximum Load: 20 kΩ
22	MON_RTN*	RETURN FOR ANALOG OUTPUTS	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal
23	Not used	N/A	Not used
24	TRIGGER_OUT	DIGITAL OUTPUT	User digital output signal, active-low; synchronization pulse of 10 ms when a change in the output occurs. Output high state either min 4.5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher. Signal return: Pin 28, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.

25	DIO_OUT1	DIGITAL OUTPUT	User digital output. Output low for CV and high for CC. Output high state either min 4.5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher Signal return: Pin 28, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.
26	FAULT	DIGITAL OUTPUT	High state indicates fault. Output high state either min 4.5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher Signal return: Pin 28, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.
27	DIO_OUT2	DIGITAL OUTPUT	Output low for CV or CC and high for CP. Output high state either min 4.5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher Signal return: Pin 28, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.
28	DOUT_RTN*	RETURN FOR DIGITAL OUTPUTS	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
29	AUX5_RTN*	RETURN FOR AUXILIARY 5V POWER OUTPUT	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
30	USER_POWER	USER POWER INPUT	(Optional) - User can connect between 5V and 24V to control digital output voltage. If left unconnected, digital outputs would have 4.5V as high state. Signal return: Pin 15, Circuit is electrically connected to the output power negative terminal
31	ISOTTL_ON/OFF	ISO TTL ON/OFF CONTROL INPUT	Isolated remote control input for output on/off with a logic signal: a logic-high, 3.3V to 24V signal will enable (turn-on) the output of the supply, and a logic-low signal disables (turns off) the output. This control input is optically isolated from the output power negative terminal of the power supply (up to 600 VDC). Signal return: Pin 16 (ISO RTN), Circuit is electrically isolated from the output power negative terminal
32	Not used	N/A	Not used

33	PPRG_VSOUR / PPRG_4-20mA_SOUR	ANALOG INPUT	Remote control input for power programming using a voltage source connected between pin 33 and signal return (pin 18). Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 V to user selectable maximum range (2 V to 10 V) for 0 to full scale rated Output Note: Do not exceed an input of 25 VDC This remote-control input can be configured to take 4- 20 mA current input for power programming. Uses same signal return pin 18 for 4- 20 mA input type also. Fixed range from 4 mA to 20 mA for 0 to full scale rated Output
34	PPROG_RES / PPRG_ISOUR	ANALOG INPUT	Remote control input for power programming using a resistance connected between pin 34 and signal return (pin 18). Current source of 1 mA for remote power programming using a resistance connected to signal return Pin 18. Signal return: Pin 18, Circuit is electrically connected to the output power negative terminal Range: 0 kΩ to user selectable maximum range (2 kΩ to 10 kΩ) for 0 to full scale rated Output Note: Do not exceed resistance of maximum 20 kΩ
35	PMON	ANALOG OUTPUT	Monitor signal for output power Signal return: Pin 22, Circuit is electrically connected to the output power negative terminal Range: 0 V to 10 V corresponds to 0-100% full-scale output. Minimum recommended Load: 100 kΩ, typical Maximum Load: 20 kΩ
36	DOUT_RTN*	RETURN FOR DIGITAL OUTPUTS	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
37	DO3 / OUT_RY_ON	DIGITAL OUTPUT	User digital output, cause to be assigned by user or state of pin 11. Output high state either min 5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher. Signal return: Pin 36, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.

38	DO4 / REV_RY_ON	DIGITAL OUTPUT	User digital output, cause to be assigned by user or state of pin 10. Output high state either min 5V or voltage on pin 30 (USER_POWER) minus 1V, whichever is higher. Signal return: Pin 36, Circuit is electrically connected to the output power negative terminal Maximum current of 0.5A.
39	AUX5_EN	DIGITAL INPUT	User digital input, enables output on pin 43. Apply a high to enable output on pin 43 (AUX5_OUT). Signal return: Pin 40, Circuit is electrically connected to the output power negative terminal Up to 24V capable, 0.3V max low, 2.7V min high.
40	AUXEN_RTN*	RETURN FOR AUXILIARY ENABLE DIGITAL INPUTS	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
41	AUX15_RTN*	RETURN FOR AUXILIARY 15V POWER OUTPUT	Return/GND. Internally, pins 12, 14, 15, 18, 22, 28, 29, 36, 40, and 41 are kept at the same DC potential. Circuit is electrically connected to the output power negative terminal.
42	AUX15_OUT	AUXILIARY USER POWER 15V OUTPUT	15V user power output, 1A max current. Signal return: Pin 41, Circuit is electrically connected to the output power negative terminal **(15V_AUX not available on 1U chassis models when not equipped with optional isolated analog interface.)**
43	AUX5_OUT	AUXILIARY USER POWER 5V OUTPUT	5V user power output, 1A max current. Signal return: Pin 29, Circuit is electrically connected to the output power negative terminal
44	AUX15_EN	DIGITAL INPUT	User digital input, enables output on pin 42 (AUX15_OUT). Apply a high to enable output on pin 42. Signal return: Pin 40, Circuit is electrically connected to the output power negative terminal **(15V_AUX not available on 1U chassis models when not equipped with optional isolated analog interface.)**

* With the option, Remote Isolated Analog Interface control, the control signal return is isolated from the output power negative terminal.

Table 9-1. Analog Programming Connector, Designations and Functions

9.3 REMOTE ANALOG PROGRAMMING

Remote programming is used for applications that require the output parameters be programmed (controlled) from a remote instrument. An external resistance, voltage or 4-20mA can be used as a programming source. External Source of programming (Resistance, Voltage or 4-20mA) is common for Output Voltage, Current and Power when remote programming mode is selected as External. When using remote programming, a shielded, twisted-pair cable is recommended to prevent noise interference to programming signals.

9.3.1 REMOTE PROGRAMMING BY RESISTANCE

Remote Analog reference source of the Output Voltage, Current and Power can be selected as resistance by sending the following SCPI Command:

SOUR:ANALOG:PROG:SOUR CURR

9.3.1.1 REMOTE CURRENT PROGRAMMING BY RESISTANCE

Analog Reference source is selected as Resistance and programming mode of output current as external by sending following SCPI commands.

SOUR:ANALOG:PROG:SOUR CURR

SOUR:CURR:PROG EXT

The resistance-programming default coefficient for output current is (100% rated output current) / 5 kΩ, with input at Pin 20 (IPRG_ISOUR) and return to Pin 18 (RTN_PRG) Refer to Figure 9-3. An internal current source, factory-set at 1 mA, from Pin 20. This pin is utilized to drive the resistance. This produces a transfer function for output current, as follows:

Iout = R * (100% rated output current) / 5 kΩ), with R in kohms.

Full Scale current programming resistance can be modified from default 5kOhms to any other value, from 2 kOhm to 10 kOhm.

For example, to set 10 kOhms as full scale for Current programming send below command.

SOUR:CURR:PROG:FSCR 10

Then the transfer function for output current, as follows:

Iout = R * (100% rated output current) / FSC kΩ), with R in kohms.

If multiple switches or relays are used to select resistors to program different current levels, make-before-break contacts are recommended.

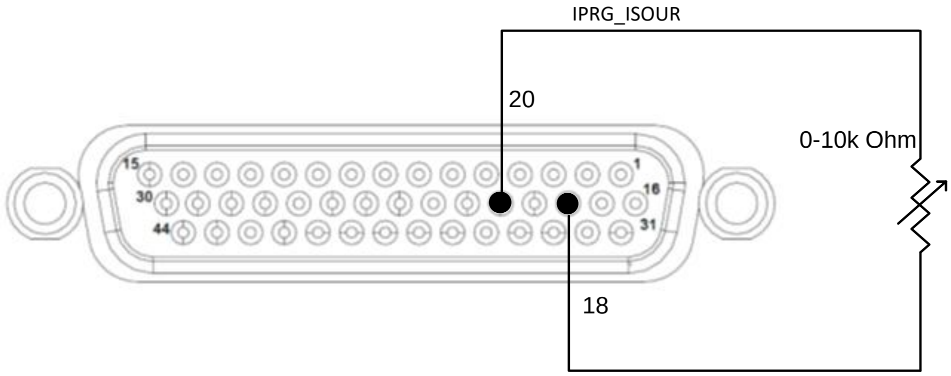


Figure 9-3 Remote Current Programming Using Resistance

9.3.1.2 REMOTE VOLTAGE PROGRAMMING BY RESISTANCE

Analog Reference source is selected as Resistance and programming mode of output Voltage as external by sending following SCPI command.

SOUR:ANALOG:PROG:SOUR CURR

SOUR:VOLT:PROG EXT

The resistance-programming default coefficient for output voltage is (100% rated output voltage) / 5 kΩ, with input at Pin 3 (VPRG_ISOUR) and return to Pin 18 (RTN_PRG. Refer to Figure 9-4. An internal current source, factory-set at 1 mA, from Pin 3 is utilized to drive the resistance. This produces a transfer function for output voltage, as follows:

Vout = R * (100% rated output voltage) / 5 kΩ), with R in kohms.

Full Scale voltage programming resistance can be modified from default 5kOhms to any other value, from 2 kOhm to 10 kOhm for Asterion DC 1U and 2U. For example, to set 10 kOhms as full scale for Voltage programming send below command

SOUR:VOLT:PROG:FSCR 10

Then the transfer function for output voltage, as follows:
Vout = R * (100% rated output voltage) / FSC kΩ), with R in kohms.

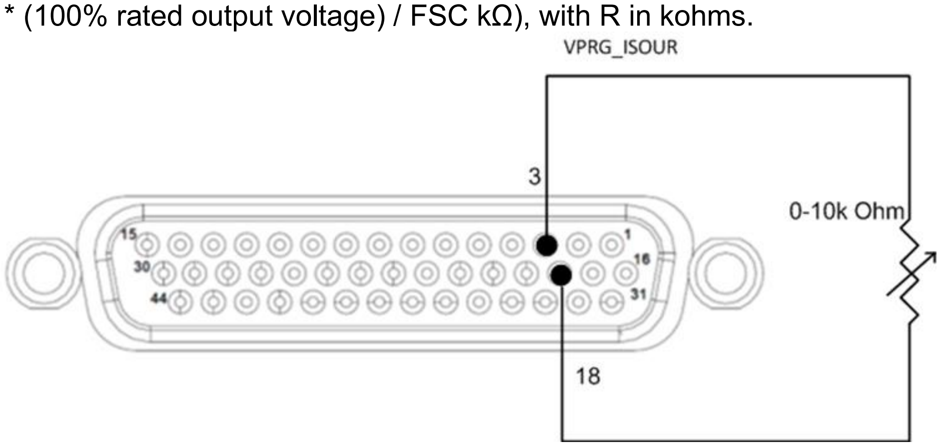


Figure 9-4 Remote Voltage Programming Using Resistance

9.3.1.3 REMOTE POWER PROGRAMMING BY RESISTANCE

Analog Reference source is selected as Resistance and programming mode of output Power as external by sending following SCPI command.

SOUR:ANALOG:PROG:SOUR CURR

SOUR:POW:PROG EXT

The resistance-programming default coefficient for output power is (100% rated output power) / 5 kΩ, with input at Pin 34 (PPRG_ISOUR) and return to Pin 18 (RTN_PRG). Refer to Figure 9-5. An internal current source, factory-set at 1 mA, from Pin 34 is utilized to drive the resistance. This produces a transfer function for output power, as follows:

Pout = R * (100% rated output power) / 5 kΩ), with R in kohms.

Full Scale power programming resistance can be modified from default 5kOhms to any other value, from 2 kOhm to 10 kOhm. For example, to set 10 kOhms as full scale for Power programming send below command

SOUR:POW:PROG:FSCR 10

Then the transfer function for output power, as follows:

Pout = R * (100% rated output power) / FSC kΩ), with R in kohms.

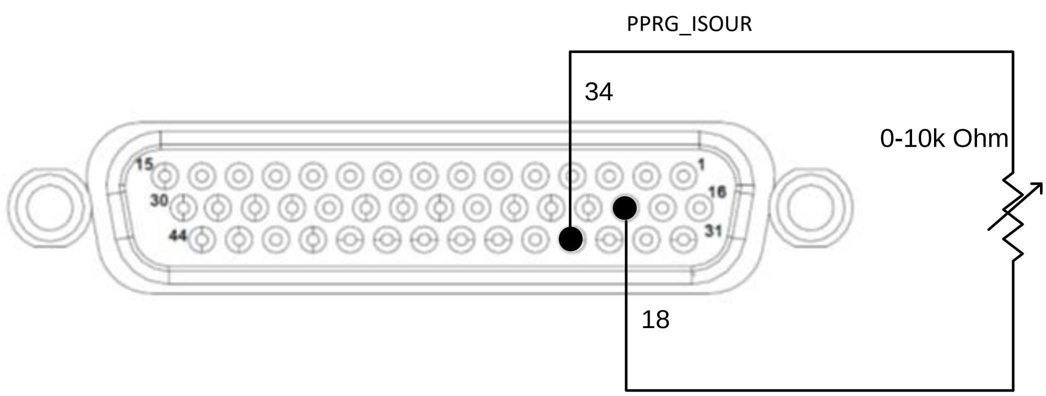


Figure 9-5 Remote Power Programming Using Resistance

9.3.2 Remote Programming BY VOLTAGE

Remote Analog reference source of the Output Voltage, Current and Power can be selected as Voltage by sending the following SCPI Command:

SOUR:ANALOG:PROG:SOUR VOLT

9.3.2.1 REMOTE CURRENT PROGRAMMING BY VOLTAGE SOURCE

Analog Reference source is selected as Voltage and programming mode of output

Current as external by sending following SCPI command.

SOUR:ANALOG:PROG:SOUR VOLT

SOUR:CURR:PROG EXT

The DC voltage source is connected between Pin 19 (IPRG_VSOUR) and the return Pin 18 (RTN_PRG), refer to Figure 9-6. Default FSC voltage value is 10V, where 10V corresponds to 100% output current. The corresponding voltage-programming coefficients for output current are (100% rated output current) / FSC VDC. This produces transfer functions for output current, as follows:

Iout = Vdc * (100% rated output current) / 10 VDC), with Vdc in volts, or

Iout = Vdc * (100% rated output current) / FSC VDC), with Vdc in volts.

The Full-Scale voltage value can be modified to any voltage between 2V to 10V. For example, to set 5V as full scale for Current programming send below command:

SOUR:CURR:PROG:FSC 5

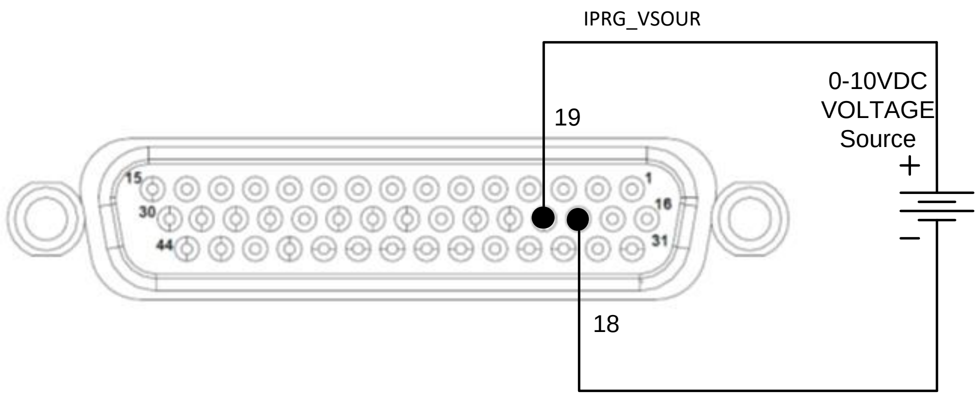


Figure 9-6 Remote Current Programming Using 0-5 VDC or 0-10 VDC Source

9.3.2.2 REMOTE VOLTAGE PROGRAMMING BY VOLTAGE SOURCE

Analog Reference source is selected as Voltage and programming mode of Output Voltage as external by sending following SCPI command

SOUR:ANALOG:PROG:SOUR VOLT

SOUR:VOLT:PROG EXT

The DC voltage source is connected between Pin 4 (VPRG_VSOUR) and the return Pin 18 (RTN_PRG, refer to Figure 9-7. Default FSC voltage value is 10V, where 10V corresponds to 100% output voltage. The corresponding voltage-programming coefficients for output voltage are (100% rated output voltage) / FSC VDC. This produces transfer functions for output voltage, as follows:

Vout = Vdc * (100% rated output voltage) / 10 VDC), with Vdc in volts, or

Vout = Vdc * (100% rated output voltage) / FSC VDC), with Vdc in volts.

The Full-Scale voltage value can be modified to any voltage between 2V to 10V. For example, to set 5V as full scale for Voltage programming send below command

SOUR:VOLT:PROG:FSCR 5

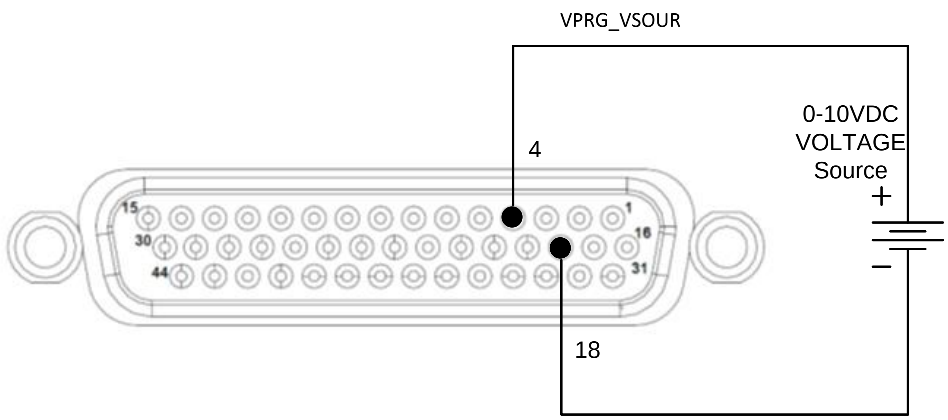


Figure 9-7 Remote Voltage Programming Using 0-10 VDC Source

9.3.2.3 REMOTE POWER PROGRAMMING BY VOLTAGE SOURCE

Analog Reference source is selected as Voltage and programming mode of Output Power as external by sending following SCPI command

SOUR:ANALOG:PROG:SOUR VOLT

SOUR:POW:PROG EXT

The DC voltage source is connected between Pin 33 (PPRG_VSOUR) and the return Pin 18 (RTN_PRG). Refer to Figure 9-8. Default FSC voltage value is 10V, where 10V corresponds to 100% output power. The corresponding voltage-programming coefficients for output power are (100% rated output power) / FSC VDC. This produces transfer functions for output power, as follows:

Pout = Vdc * (100% rated output power) / 10 VDC), with Vdc in volts, or

Pout = Vdc * (100% rated output power) / FSC VDC), with Vdc in volts.

The Full-Scale voltage value can be modified to any voltage between 2V to 10V from below SCPI command. For example, to set 5V as full scale for Power programming send below command

Ex - SOUR:POW:PROG:FSC 5

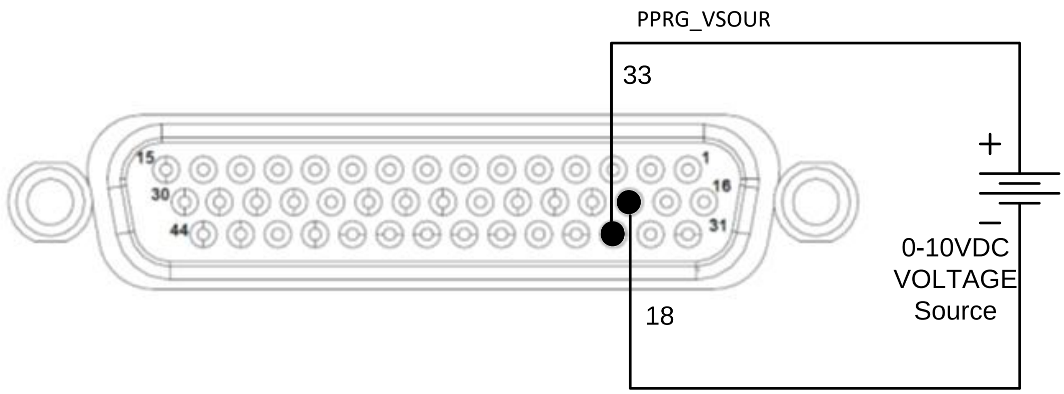


Figure 9-8 Remote Power Programming Using 0-10 VDC Source

9.3.3 Remote Programming BY 4-20mA SOURCE

Remote Analog reference source of the Output Voltage, Current and Power can be selected as 4-20 mA by sending the following SCPI Command:

SOUR:ANALOG:PROG:SOUR MA420

9.3.3.1 REMOTE CURRENT PROGRAMMING BY 4-20MA SOURCE

Analog Reference source is selected as 4-20mA and programming mode of Output Current as external by sending following SCPI command

SOUR:ANALOG:PROG:SOUR MA420

SOUR:CURR:PROG EXT

A 4-20mA current source is connected between Pin 19 (IPRG_VSOUR) and the return

Pin 18 (RTN_PRG), refer to Figure 9-9.

The transfer function for the output current will be as follows:

Iout = ($I_{4\text{-}20mA}$ – 4) (100% rated output current) / 16, with $I_{4\text{-}20mA}$ in mA

Which produces 0A output current at 4mA and 100% rated output current at 20mA.

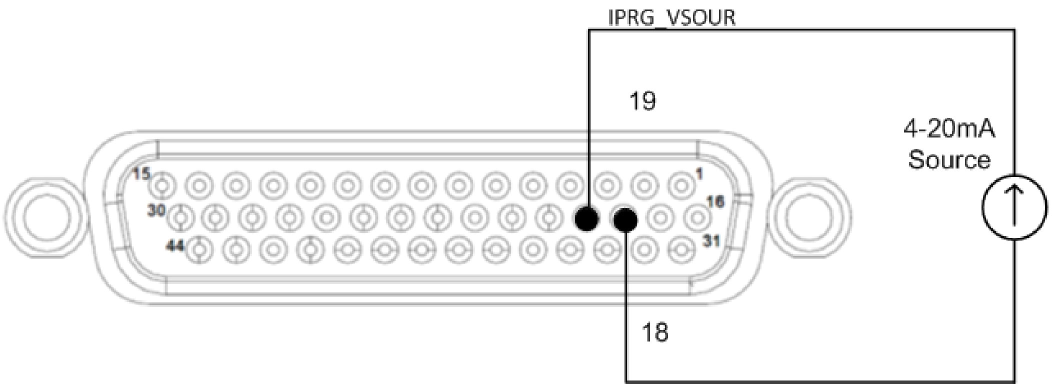


Figure 9-9 Remote Current Programming Using 4-20mA Source

9.3.3.2 REMOTE VOLTAGE PROGRAMMING BY 4-20MA SOURCE

Analog Reference source is selected as 4-20mA and programming mode of Output Voltage as external by sending following SCPI command

SOUR:ANALOG:PROG:SOUR MA420

SOUR:VOLT:PROG EXT

A 4-20mA current source is connected between Pin 4 (VPRG_VSOUR) and the return Pin 18 (RTN_PRG, refer to Figure 9-10.

The transfer function for the output voltage will be as follows:

Vout = ($I_{4\text{-}20mA}$ – 4) (100% rated output voltage) / 16, with $I_{4\text{-}20mA}$ in mA

Which produces 0V output voltage at 4mA and 100% rated output voltage at 20mA.

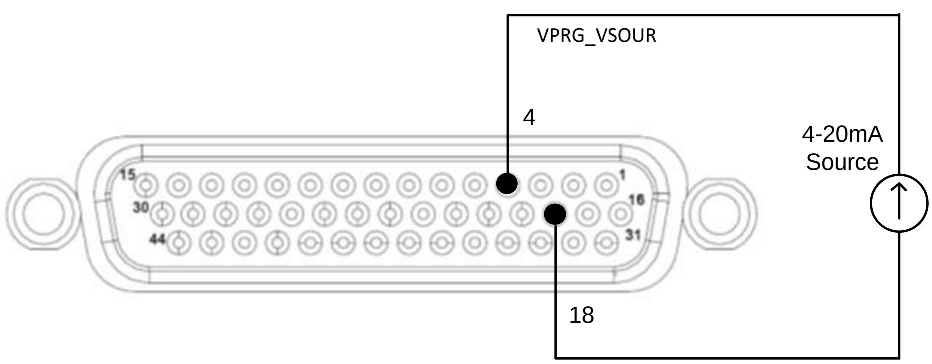


Figure 9-10 Remote Voltage Programming Using 4-20mA Source

9.3.3.3 REMOTE POWER PROGRAMMING BY 4-20MA SOURCE

Analog Reference source is selected as 4-20mA and programming mode of Output Power as external by sending following SCPI command

SOUR:ANALOG:PROG:SOUR MA420

SOUR:POW:PROG EXT

A 4-20mA current source is connected between Pin 33 (PPRG_VSOUR) and the return Pin 18 (RTN_PRG)., refer to Figure 9-11.

The transfer function for the output power will be as follows:

Pout = ($I_{4\text{-}20mA}$ – 4) (100% rated output power) / 16, with $I_{4\text{-}20mA}$ in mA

Which sets 0W output power at 4mA and 100% rated power voltage at 20mA.

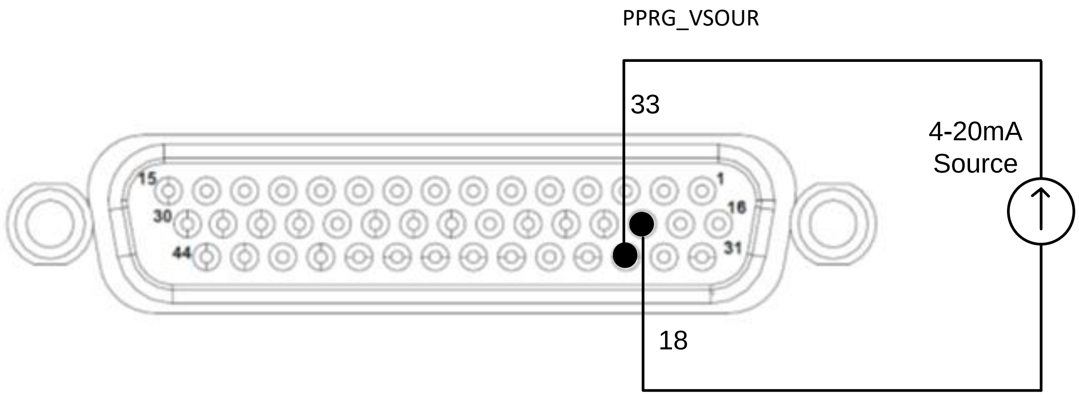


Figure 9-11 Remote Power Programming Using 4-20mA Source

9.3.4 Remote Overvoltage Programming

A remote DC voltage source can be connected externally between Pins 5 (OVPRG_VSOUR) and Pin 18 (RTN_PRG) to set the output overvoltage trip level. A 0-10 VDC signal equals 0-110% of rated output voltage. See Figure 9-12 for connection requirements. This full-scale programming voltage source range can be set to any value between 2V to 10VDC from front panel. Only External Voltage source can be used for Remote Overvoltage Programming. Refer to below set of SCPI commands.

SOUR:VOLT:PROT:PROG EXT

SOUR:VOLT:PROT:PROG:FSC 5

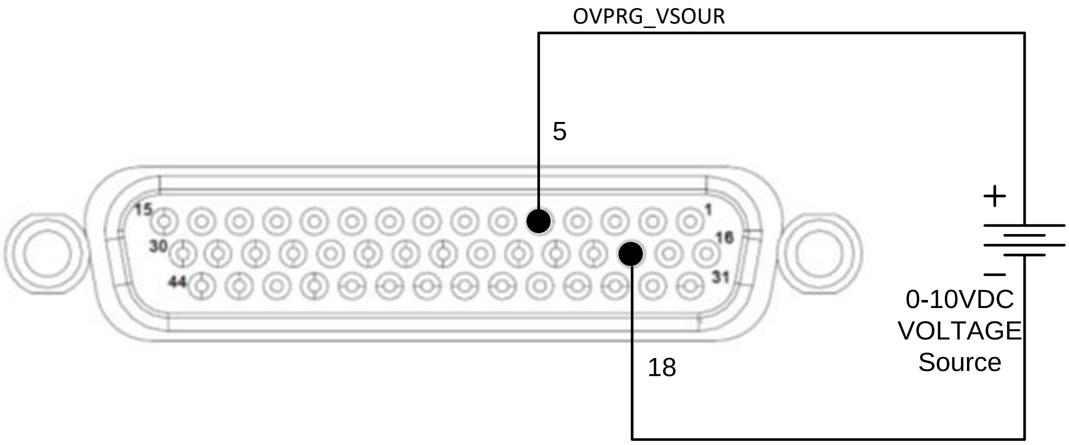


Figure 9-12 Remote Overvoltage Programming Using DC Voltage Source

9.4 REMOTE MONITOR OUTPUTS

DC Asterion provides independent analog voltage monitor signals for the Output Voltage, Current and Power.

1. Voltage Monitor Output is in Pin 6 (VMON) and return Pin 22 (RTN).
2. Current Monitor Output is in Pin 21 (IMON) and return Pin 22 (RTN).
3. Power Monitor Output is in Pin 35 (PMON) and return Pin 22 (RTN).
4. Full Scale Monitor voltage would correspond to rated output of the power supply. Default Full Scale Monitor Output is 10V, the values be changed from 0V to 10 V by issuing SCPI commands. For example, to set to 5V as full scale for voltage monitoring send below command:
   SOUR:VOLT:MON:FSC 5
5. For the Current monitor to change the full-scale value to 5V, following SCPI command to be used.
   SOUR:CURR:MON:FSC 5
6. Similarly, for the Power monitor to change the full-scale value to 5V, following SCPI command to be used.
   SOUR:POW:MON:FSC 5

9.5 REMOTE DIGITAL INPUT AND OUTPUT

9.5.1 Auxiliary power output

Asterion DC power supply has the option of two auxiliary power outputs 5 V and 15V. These auxiliary power outputs can be controlled from the power supply remotely or by giving appropriate signals on the digital input enable pins provided for the same.

1. To enable the 5V output remotely following SCPI command to be used:

   OUTP:AUX:5V ON

2. To enable the 15V output remotely following SCPI command to be used:

   OUTP:AUX:15V ON

3. To Enable the 5V output from remote Digital Input:

   Apply a high between pin 39 (AUX5_EN) and pin 40 (AUXEN_RTN) to enable output of 5V DC across pin 43 (AUX5_OUT) and pin 29 (AUX5_RTN).

   Voltage Rating: Maximum 24 V, Minimum -5V

   Low state 0.3 V max, High State 2.7 V min.

4. To Enable the 15V output from remote Digital Input:

   Apply a high between pin 44 (AUX15_EN) and pin 40 (AUXEN_RTN) to enable output of 15V DC across pin 42 (AUX15_OUT) and return pin is 41 (AUX15_RTN).

   Voltage Rating: Maximum 24 V, Minimum -5V

   Low state 0.3 V max, High State 2.7 V min.

9.5.2 Digital outputs

DC Asterion power supply provides four digital outputs:

Digital Output 1: Provides Low state at Constant Voltage (CV) mode of operation and High state at Constant Current (CC) operation. Output signal Pin is 25 (DIO_OUT1) and return pin is 28 or 36 (DOUT_RTN).

Digital Output 2: Provides High state at Constant Power (CP) mode of operation otherwise remains at low state. Output signal Pin is 27 (DIO_OUT2) and return pin is 28 or 36 (DOUT_RTN).

Digital Output 1 and Output 2 states reflects the query response of following SCPI command.

SOUR:CURR:MODE? ------ Returns the mode of operation of the power supply.

0 - Constant Voltage operation,
1 - Constant Current operation and
2 - Constant Power operation

Command	Response	Digital Output-1	Digital Output-2
SOUR:CURR:MODE?	0	Low State	Low State
	1	High State	Low State
	2	Low State	High State

Digital Output 3: Output Signal, pin 37 (DO3 / OUT_RY_ON) and return pin 28 or 36 (DOUT_RTN) can be made high by using high signal on the Digital input pin 11 (OUT_RY_EN) and return pin 12 (DIN_RTN) or by issuing following SCPI Command.

OUTP:ISOL ON

Digital Output 4: Output Signal, pin 38 (DO4 / REV_RY_ON) and return pin 28 or 36 (DOUT_RTN) can be made high by using high signal on the Digital input

pin 10 (REV_RY_EN) and return pin 12 (DIN_RTN) or by issuing following SCPI Command
OUTP:POL ON

For the all the Digital Output, Low state and High state signal level is as follows.

Low state: 0.3 V max

High state: Minimum 3V or Voltage at Pin 30 (User Power), whichever is higher. Maximum Current is 0.5 A

User Power: 5 V to 24 V (Pin 30), Return Pin for User Power is Pin 36.

Connection diagram to use the signals to control external Isolation and Polarity relays from the DC-Asterion power supply is shown in Figure 9-13.

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.

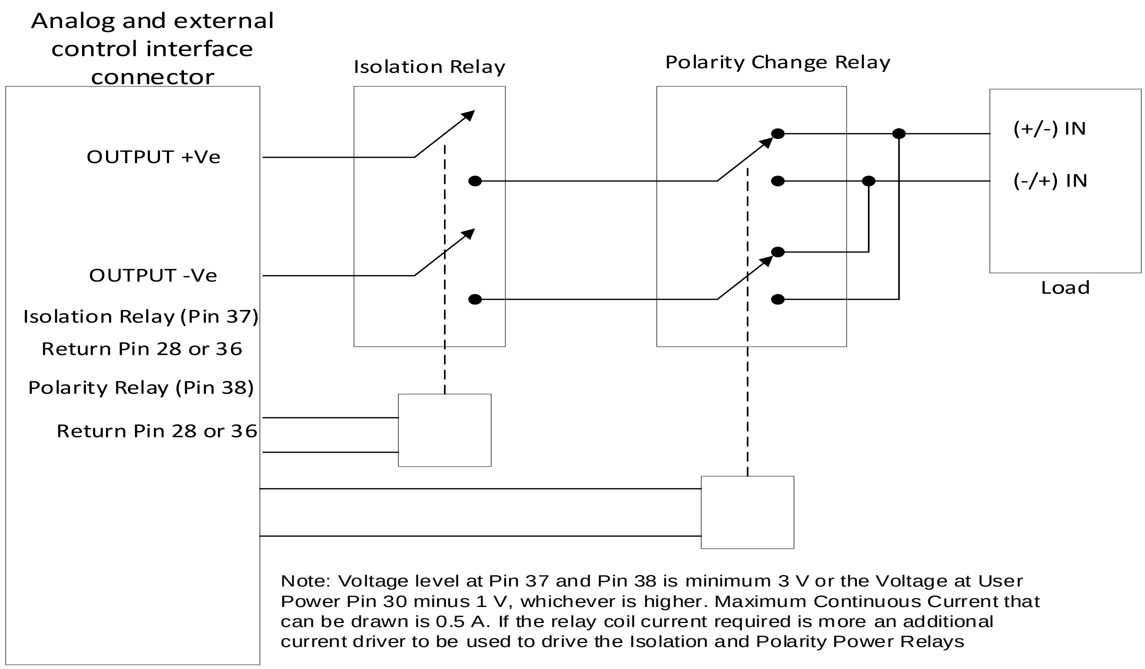


Figure 9-13 External Isolation and Polarity Relay interface to Control interface

9.5.3 Digital Inputs

Asterion DC 1U and 2U power supplies provide four digital inputs:

Digital Input 1: This input is general purpose and cause can be assigned by the user. Input Signal is Pin 9, Return Pin 12.

Following SCPI command returns 1 or 0.

SOUR:DIGINP:DIN1?

Digital Input 2: This input is general purpose and cause can be assigned by the user. Input Signal is Pin 7, Return Pin 12.

Following SCPI command returns 1 or 0.

SOUR:DIGINP:DIN2?

Digital Input 3 (OUT_RY_EN / DIN3): This input is general purpose and cause can be assigned by the user. Other functionality is that applying high input on this signal would enable high output on pin 37, Digital Output 3 (OUT_RY_ON). Input signal pin is 11, Return Pin 12.

Following SCPI command returns 1 or 0.

SOUR:DIGINP:DIN3?

Digital Input 4 (REV_RY_EN / DIN4): This input is general purpose and cause can be assigned by the user. High input on this signal would enable high output on pin 38, Digital Output 3 (OUT_RY_ON). Input signal pin is 10, Return Pin 12.

Following SCPI command returns1 or 0.

SOUR:DIGINP:DIN4?

Input high and low state levels are as described below.

1-High State 2.7 V min, 24 V max

0-Low state 0.3 V max, -5V min

9.5.4 Fault

High state in this pin indicates the Fault in the power supply. Fault Signal Pin is 26 (FAULT) and the return pin is 28 (DOUT_RTN). Fault Signal Pin becomes High whenever there is a fault recorded in the Protection Condition Register. An exception to this Constant Voltage Operation (Bit 0), Constant Current Operation (Bit 1) and Constant Power Operation (Bit 2), as they indicate the operating status of the power supply.

Low state: 0.3 V max

High state: Minimum 3V or Voltage at Pin 30 (User Power), whichever is higher. Maximum Current is 0.5 A

User Power: 5 V to 24 V (Pin 30), Return Pin for User Power is Pin 36

Following are the list of faults that can be generated:

Bit	Hex Value	Description	Explanation	Actions

3	0x08	Overvoltage Protection Fault	Output voltage exceeded Over Voltage trip limit.	• Check sufficient margin between set voltage and OVP limit • Check if OVP is set from external Analog source and sufficient margin between set voltage and OVP limit
4	0x10	Over Temperature Fault	Power supply internal hardware temperature exceeded limit	• Check and make sure there is no Air blockage in the front and rear of the power supply. Make sure Ambient temperature and the Operating Power limit with input line condition is as per the Specifications. If fault persists, Contact factory
5	0x20	External Shutdown	External Shutdown activated from External Analog Programming Connector	• Make sure Pin 13 and Pin 14 are connected for disabling External Shutdown
6	0x40	Foldback Mode Operation	Trip activated due to foldback operation setting	• Check the foldback operation setting from OUTP:PROT:FOLD? Query and modify as required
7	0x80	Remote Programming Error	Power supply remote programming command is incorrect	• Verify the SCPI command sent from remote communication interface is as per programming manual. • Verify command sent are not violating any user limits or hardware parameter range specified in datasheet of power supply
8	0x100	Fan Fault	Fault from cooling Fan	• This condition is caused by Fan's not running properly. Reset the fault or try restarting the power supply and If fault persists, Contact factory.
9	0x200	Line Drop Fault	Input voltage to the power supply is not in operating range	• Check if input voltage to the supply is within specified range in datasheet
10	0x400	DC Module Fault	Internal DC module hardware fault	• Try restarting the power supply, if fault persists Contact factory

11	0x800	PFC Fault	Internal PFC module hardware fault	• Try restarting the power supply if fault persists Contact factory
12	0x1000	OCP Fault	Output overcurrent Fault	• By default, this fault is disabled and not accessible to user
13	0x2000	AUX Supply Fault	Auxiliary Supply to internal hardware fault	• Try restarting the power supply if fault persists Contact factory
14	0x4000	Line Status Changed	Input voltage changed from Low range to High range or Vice versa	• Make sure input voltage is stable at one voltage range. If there is change over in input voltage range power supply's output power limits will be reset based on the limits specified in datasheet
15	0x8000	Parallel Cable Fault	Chassis Parallel cables PAR IN/PAR OUT or CAN IN/CAN OUT cables are connected or disconnected when power supply is ON	• Reset the PAR IN/PAR OUT or CAN IN/CAN OUT cables as required and clear the faults
16	0x10000	Follower System Fault	One of the Follower units in Parallel/ Series chassis configuration NOT responding	• Check all follower units in Parallel/Series chassis configuration make sure they are powered ON • Power cycle all the units of Parallel/Series chassis configuration
17	0x20000	Not Used		
18	0x40000	Remote Sense Fault	Remote voltage sensing is out of range from power supply capacity or cable connected to RVS connector fault	• Check if the cable connected to RVS connected at rear side power supply is intact and polarity is correct. • Check the output cable voltage drop and make sure voltage drop across cable is not exceeding limit specified in datasheet
19	0x80000	Regulation Fault	Power supply not able to settle to any of the operating modes CC/CV or CP	• Try restarting the power supply if fault persists Contact factory
20	0x100000	Current Feedback Fault	Internal DC modules current	• Try restarting the power supply if fault persists Contact factory

			feedbacks are unbalanced	
21-31		Not Used (Always returns zero for these bits)		

Table 9-2. List of Faults in DC-Asterion Power Supply

9.5.5 Trigger-In & Trigger Out

DC Asterion power supply has Trigger-In and Trigger-Out functionality.

Trigger-In: Applying a high voltage signal on pin 8 (TRIGGER_IN) and return pin 12 (DIN_RTN) will trigger the ramp.

```
*CLS                              // clear the unit to its power-on default settings.
*RST                              // reset the unit.
SOUR:VOLT 20                      // Set the voltage to 20V
SOUR:CURR 20                      // Set the current to 20A
OUTP 1                            // Turn on the output
SOUR:VOLT:RAMP:HTRIG 25 5         //Give an external trigger i.e. an active high
                                  pulse of 10ms on the pin-8 (TRIGGER_IN) and pin-
                                  12 (DIN_RTN) to generate the Voltage Ramp. This
                                  will ramp the voltage from 20V to 25V in 5 secs
```

Low state 0.3 V max, High State 2.7 V min

Trigger-Out: Active low signal will be generated on the pin 24 (TRIGGER_OUT) and return pin 36 (DOUT_RTN) when a change in the output occurs.

9.6 REMOTE OUTPUT ON/OFF CONTROL

Remote output on/off control may be accomplished by contact closure, or through an opto-isolated interface with external voltage sources, AC/DC or TTL/CMOS.

9.6.1 Remote Output ON/OFF by Contact Closure

Application of a contact closure between pins 13 (Remote ON/OFF) and pin 14 (Remote ON/OFF_RTN) will enable the output (if Output Enable from front panel or SCPI is ON). See Figure 9-14 for connection requirements.

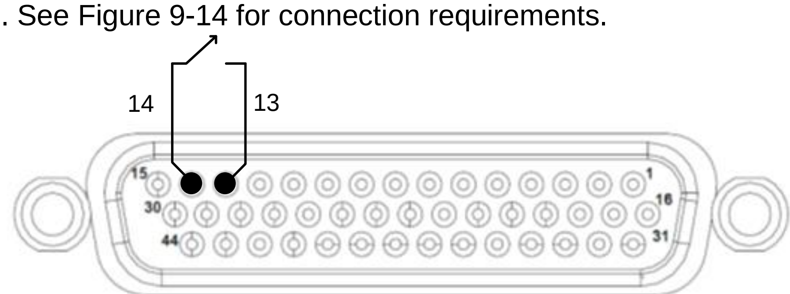


Figure 9-14 Remote Output On/Off Control by Contact Closure

9.6.2 Remote Output ON/OFF Control by External Source

Application of AC/DC voltage between Pins 1 (ISO HV ON/OFF) and pin 16 (ISO_RTN / RTN_HV), or TTL/CMOS voltage between pin 31 (ISOTTL_ON/OFF) and pin 16 (ISO_RTN / RTN_HV), will turn on the power supply. This interface is opto-isolated from circuit common. See Figure 9-15 and Figure ***9-16*** for connection requirements.

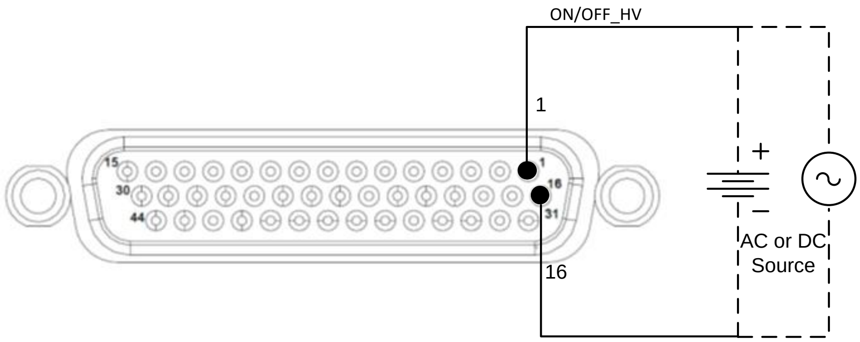


Figure 9-15 Remote Output On/Off Using Isolated AC or DC Source for Asterion DC 1U and 2U

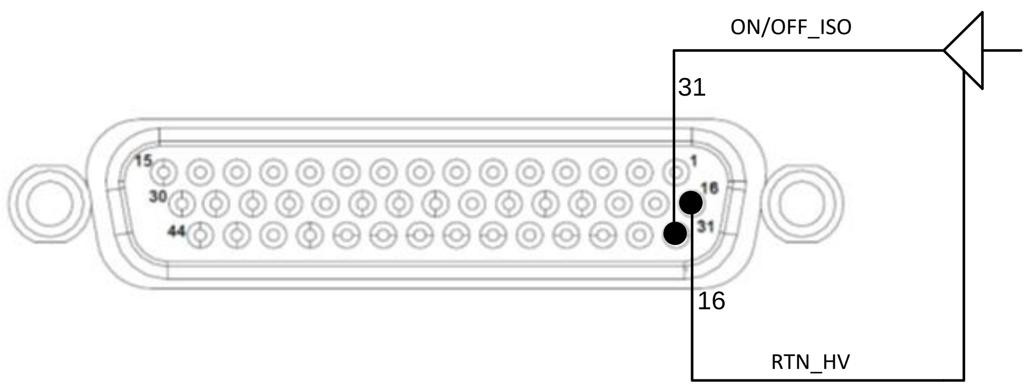


Figure 9-16 ***Remote Output On/Off Using Isolated TTL/CMOS Source for Asterion DC 1U and 2U***

10
RS232/ETHERNET/ IEEE 488.2 GPIB AND SCPI COMMAND OPERATION

10.1 INTRODUCTION

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

10.2 REGISTER DEFINITIONS

The Asterion DC Power Supply supports IEEE 488.2 GPIB, RS232, Ethernet, EtherCAT 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.

10.2.1 Protection Condition and Protection Event Status Register

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

10.2.1.1 STATUS PROTECTION CONDITION REGISTER

Read the Protection Condition Register by issuing the following command.

STATus:PROTection:CONDtion?

The above 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. The query value hexadecimal number of 32 bits, the definition corresponding to each bit is listed in **Error! Reference source not found.**Table 10-1 Bit value “1” indicates fault and “0” indicates normal condition.

For Example:
STAT:PROT:COND?
#H0000042 // Response for the above command

which means two bits are set: Constant voltage operation(0x02) and Fold back mode operation(0x40)

10.2.1.2 STATUS PROTECTION EVENT REGISTER OR FAULT REGISTER

Read the Protection Event Status Register by issuing the following command.
STATus:PROTection:EVENt?

Refer Table 10-1**Error! Reference source not found.** 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.

10.2.1.3 STATUS PROTECTION ENABLE REGISTER OR FAULT REGISTER

Set the Status Protection Enable Register with the following command.

STATus:PROTection:ENABle <mask>

Read the Status Protection Enable Register with the following query command.

STATus:PROTection:ENABle?

Protection Enable Register Used to select what fault events could set a bit in the Fault Register. Certain faults can occur even if they are not enabled in the Protection Enable Register. 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.

All the events in the protection condition register causes shutdown to the power supply, an exception to this rule involves the Constant Voltage Operation, Constant Current Operation, Constant Power Operation and Foldback Mode Operation bits. If these bits are not enabled in the protection enable register, then mode changes shall not cause a shutdown.

Figure 10-1 shows the implementation of the Protection Status Register, Protection Enable Register and Protection Event Register. 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 10.2.2 and Section 13 SCPI Status Implementation for more details.

Bit	Hex Value	Description
0	0x01	Constant Voltage Operation
1	0x02	Constant Current Operation
2	0x04	Constant Power Operation
3	0x08	Overvoltage Protection Fault
4	0x10	Over Temperature Fault
5	0x20	External Shutdown
6	0x40	Foldback Mode Operation
7	0x80	Remote Programming Error
8	0x100	Fan Fault
9	0x200	Line Drop Fault
10	0x400	DC Module Fault
11	0x800	PFC Fault
12	0x1000	OCP Fault
13	0x2000	AUX Supply Fault
14	0x4000	Line Status Changed
15	0x8000	Parallel Cable Fault
16	0x10000	Follower System Fault
17	0x20000	Not Used
18	0x40000	Remote Sense Fault
19	0x80000	Regulation Fault
20	0x100000	Current Feedback Fault
21-31	Not Used (Always returns zero for these bits)	

Table 10-1: Protection Condition and Event Status Registers

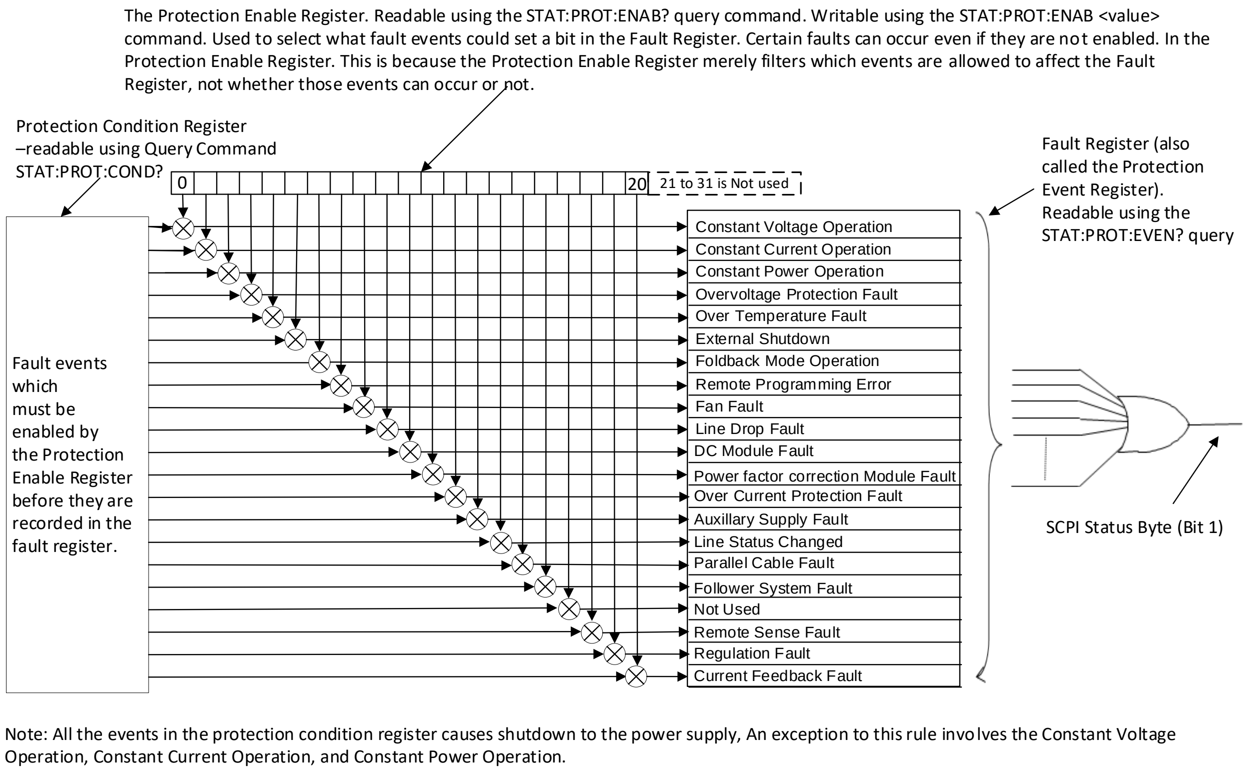


Figure 10-1. Protection Condition and Protection Event Register

10.2.2 Standard Event Status Register (ESR)

Read the Standard Event Status Register (ESR) by issuing the *ESR? command. See 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 10-2 and Section 13 SCPI Status Implementation for further information. Table 10-2 for 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

Table 10-2: Standard Event Status Register

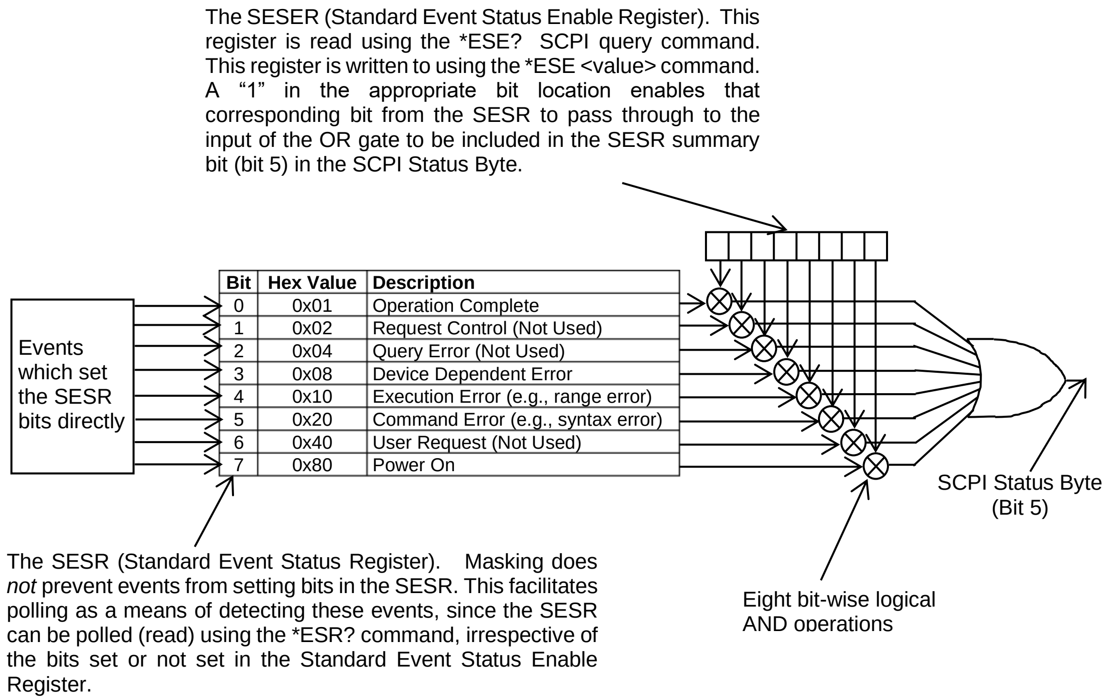


Figure 10-2. Standard Event Status and Standard Event Status Enable Register

10.2.3 SCPI Status Byte

The SCPI Status Byte registers the status of the instrument, in one of seven bits described in Table 10-3. 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 10-3.

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 10-3 and Section 13 SCPI Status Implementation for further information.

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 Summary Status (SS) 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 SS bit indicates that the device has at least one reason to request service. Even though the device sends the SS 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.

Table 10-3: SCPI Status Byte

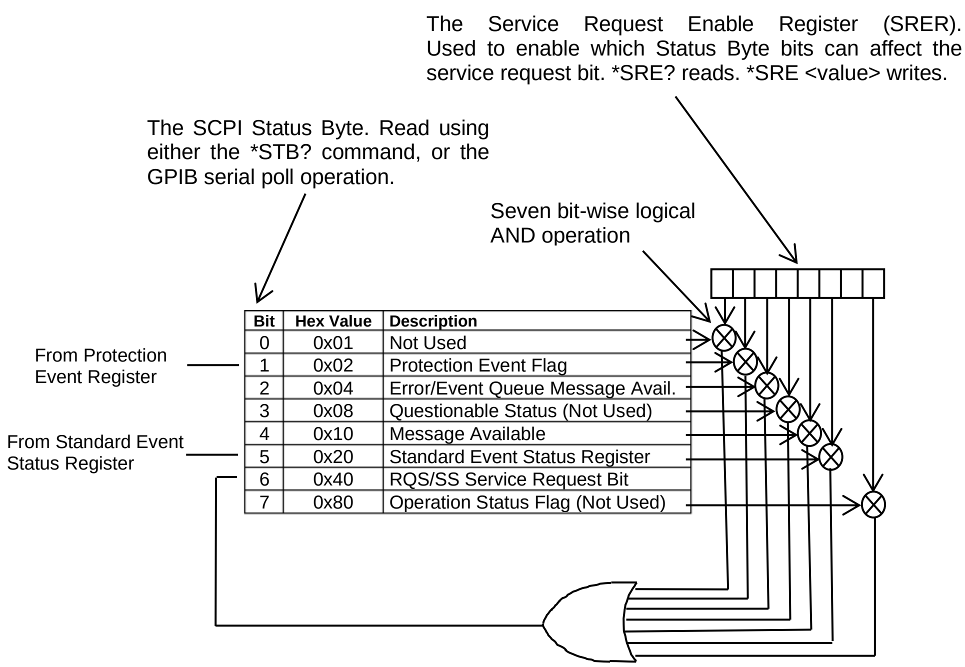


Figure 10-3. SCPI Status Byte and Service Request Enable Register

10.2.4 Error/Event Queue

The Asterion DC 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 Asterion DC 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.

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.
-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.

Error Code	Description
-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.

Table 10-4: SCPI Error Codes

10.2.5 Serial Poll Operation

Performing a serial poll will not modify the SCPI Status Byte other than to clear the RQS (bit 6) for a Asterion DC 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

10.2.6 Self-Test

The Asterion DC 1U/2U series power supply has power ON self-test functionality. Self-test is initiated whenever the power is recycled. If the power supply is faulty then the state will be reserved with the device. The Fault state could be identified by issuing the following command `*TST?`. This will return integer value of the fault state that occurs during self-test; refer Table 10-5. Self-Test could be performed at any interval by sending the following command `*TST.` If no errors are observed during the test, "`*TST?`" would return 0, this confirms that the self-test is passed. If the test fails `*TST?` would return integer value in the range of [0- 63].

Example: if *TST? returns "31"

Binary equivalent of 31(int) is 0001 1111

Bit	LSB **0**	**1**	**2**	**3**	**4**	**5**	**6**	MSB **7**
Faults	Hardware Fault	Interface communication Test 1 Fault	Interface communication Test 2 Fault	Front Panel Communication Fault	FLASH Memory Test Fault	EEPROM Memory Test Fault	Not Used	Not Used
Binary equivalent of integer	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**

Binary "1" represents that the respected fault state is high and Binary "0" represents that the respected fault state is low.

If hardware fault occurs, it would be displayed in front panel and indicated with fault

LED.

To identify hardware specific faults "`STAT:FAULT?`" command should be issued to the supply. Refer section 10.2.1.

The following fault state are supported by the Asterion DC 1U/2U series power supply. Fault states are in the range of [0 - 63].

Bit	Integer Value	Description
0	1	Hardware Fault
1	2	Interface communication Test 1 Fault
2	4	Interface communication Test 2 Fault
3	8	Front Panel communication Fault
4	16	Flash memory Test Fault
5	32	EEPROM Memory Test Fault
6	64	Not Used
7	128	Not Used

Table 10-5: Self-Test Error Codes

10.2.7 Fault Status

Power Supply faults status can be identified by sending the following command `STAT:FAULT?`. This would return Hex value of the fault state value. The following fault states are supported by the Asterion DC 1U/2U series power supply, refer Table 10-6.

Bit	Hex Value	Description
0	0x01	Current Feedback Fault
1	0x02	Local Sense Fault / Regulation Mode Fault
2	0x04	Remote Sense Fault
3	0x08	Follower System Fault
4	0x10	CV Fault
5	0x20	CC Fault
6	0x40	Output Tripped
7	0x80	Overvoltage Protection Fault
8	0x100	Over Temperature Fault
9	0x200	External Shutdown
10	0x400	Foldback Mode Operation
11	0x800	Remote Programming Error
12	0x1000	Fan Fault
13	0x2000	Line Drop Fault
14	0x4000	DC Module Fault
15	0x8000	PFC Fault
16	0x10000	OCP Fault
17	0x20000	AUX Supply Fault

18	0x40000	Line State changed
19	0x80000	Parallel cable fault
20	0x100000	Not used

Table 10-6: Faults

Example: if `STAT:FAULT?` Returns “C0”

Binary equivalent of 0xC0 is “1100 0000 0000 0000 0000”

Binary “0” represents fault state is low and “1” represents fault state is high.

Bit	0 LSB	1	2	3	4	5	6	7	8	9
Status	Current Feedback Fault	Local Sense fault/ Regulation Mode Fault	Remote Sense Fault	Follower system Fault	CV Fault	CC Fault	Output tripped	OVP Fault	OTP Fault	External Shutdown Fault
Binary Equivalent	0	0	0	0	0	0	1	1	0	0

10	11	12	13	14	15	16	17	18	19
Foldback Mode Operation Fault	Remote Programming Error	Fan Fault	Line Drop Fault	DC Module Fault	PFC Fault	OCP Fault	Auxiliary Supply Fault	Line State Changed	Parallel cable Fault
0	0	0	0	0	0	0	0	0	0

10.3 SCPI CONFORMANCE INFORMATION

The syntax of all SCPI commands implemented by the Asterion DC 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 Asterion DC power supplies are classified as SCPI approved commands (approved by the SCPI Consortium but not contained in the SCPI version to which the Asterion DC 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

10.3.1 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.

10.3.2 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

10.3.3 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>).

10.4 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 10.2.2 for valid values.

Command	Description
*ESE?	Returns the integer value of the Standard Event Status Enable Register. See section 10.2.2 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 10.2.2 for valid values. **Response:** <0+NR1>
*IDN?	Returns the device identification as an ASCII string. **Response:** <Manufacturer>, <model>, <serial number>, <SC firmware version>, AIB firmware version>,<FPC firmware version> SC-System Controller AIB- Analog Interface Board FPC- Front Panel Controller **Example:** AMETEK, AST60-83D1C-E01,0622A00111,1.000,1.00,1.00
*OPC	Enables the Operation Complete bit of the Standard Event Status Register to be set when all pending operations are complete. See section 10.2.2.
*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.
*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 10.2.2 on Status Byte for valid values.
*SRE?	Returns the integer value of the Service Request Enable Register. See section 10.2.2 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 Summary Status (SS) instead of RQS. The SS 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 10.2.2 on Status Byte for valid values. Values range from 0-255. **Response:** <0+NR1>
*TST	Performs the self-test for the power supply and reserves the self-test status with device. For more information refer section 10.2.6.
*TST?	Returns the integer value of self-test status (*TST). Value ranges from 0-63. Response: <0+NR1>

10.5 SOURCE SCPI COMMAND SUBSYSTEM

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

10.5.1 SOURCE SCPI Command Summary

```
SOURce
     :ANALOG
          :PROGram
               :SOURce <string>
               :SOURce?
     :CURRent <NRf>
     :CURRent?
          [:LEVel] <NRf>
          [:LEVel]?
               [:IMMediate] <NRf>
               [:IMMediate]?
                    [:AMPLitude] <NRf>
          :MODE?
          [:AMPLitude]?
          :LIMit<NRf>
          :LIMit?
               [:AMPLitude] <NRf>
               [:AMPLitude]?
          :RAMP <NRf> <NRf>
          :RAMP?
               :ABORt
               :ALL?
               :HTRIGgered <NRf> <NRf>
               :HTRIGgered?
               :TRIGgered <NRf> <NRf>
               :TRIGgered?
          :TRIGgered
          :TRIGgered?
               :CLEar
               :AMPLitude <NRf>
               :AMPLitude?
          :PROGram <string>
          :PROGram?
               :FSC <NRf>
               :FSC?
               :FSCR <NRf>
               :FSCR?
          :MONitor
               :FSC <NRf>
               :FSC?

SOURce
     :VOLTage
     :VOLTage?
          [:LEVel] <NRf>
          [:LEVel]?
```

```
                    [:IMMediate] <NRf>
                    [:IMMediate?]
                         [:AMPLitude] <NRf>
                         [:AMPLitude]?
               :LIMit <NRf>
               :LIMit?
                    [:AMPLitude] <NRf>
                    [:AMPLitude]?
               :PROTection <NRf>
               :PROTection?
                    [:LEVel] <NRf>
                    [:LEVel]?
                    :TRIPped?
                    :PROGram <string>
                    :PROGram?
                    :FSC <NRf>
                    :FSC?
                    :STATe?
                    :CLEar
               :RAMP <NRf> <NRf>
               :RAMP?
                    :ABORt
                    :ALL?
                    :HTRIGgered <NRf> <NRf>
                    :HTRIGgered?
                    :TRIGgered <NRf> <NRf>
               :TRIGgered?
               :TRIGgered <NRf>
               :TRIGgered?
                    :CLEar
                    :AMPLitude <NRf>
                    :AMPLitude?
               :PROGram <string>
               :PROGram?
                    :FSC <NRf>
                    :FSC?
                    :FSCR <NRf>
                    :FSCR?
               :MONitor
                    :FSC <NRF>
                    :FSC?

SOURce
     :POWer <NRf>
     :POWer?
          [:LEVel] <NRf>
          [:LEVel?]
               [:IMMediate] <NRf>
               [:IMMediate?]
                    [:AMPLitude] <NRf>
                    [:AMPLItude?]
          :MAX?
          :LIMit <NRf>
```

```
            :LIMit?

            :PROGram <string>
            :PROGram?
                 :FSC <NRf>
                 :FSC?
                 :FSCR <NRf>
                 :FSCR?
            :MONitor
                 :FSC <NRF>
                 :FSC?
SOURce
     :DIGINP
            :DIN1?
            :DIN2?
            :DIN3?
            :DIN4?
```

10.5.2 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:ANALOG:PROGram:SOURce <string>	Changes the source for EXTernal analog programming Mode. Valid arguments are VOLT, CURR and MA420. VOLT – Voltage source (0-10V range) CURR – Resistive load on constant 1mA source (0-10kOhm). 1mA current source is in built in power supply. MA420 – Standard 4-20mA source	N
SOURce:ANALOG:PROGram:SOURce?	Returns the setting of External analog programming source.	N
SOURce:CURRent <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURce:CURRent?	Returns the output current in amps or in milliamps	C
SOURce:CURRent[:LEVel] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURce:CURRent[:LEVel]?	Returns the output current in amps or in milliamps.	C
SOURce:CURRent[:LEVel][:IMMediate] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURce:CURRent[:LEVel][:IMMediate]?	Returns the output current in amps or in milliamps	C
SOURce:CURRent[:LEVel][:IMMediate] [:AMPLitude] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURce:CURRent[:LEVel][:IMMediate] [:AMPLitude]?	Returns the output current in amps or in milliamps	C
SOURce:CURRent:MODE?	Returns the mode of operation of the power supply.	N

Command	Description	SCPI
	0 - Constant voltage operation, 1 - Constant current operation and 2 - constant power operation	
SOURce:CURRent:LIMit<NRf>	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
SOURce:CURRent:LIMit?	Returns the upper soft limit on the programmed output current for the supply.	C
SOURce:CURRent:LIMit[:AMPLitude] <NRf>	Sets an upper soft limit on the programmed output current for the supply.	C
SOURce:CURRent:LIMit[:AMPLitude]?	Returns the upper soft limit on the programmed output current for the supply.	C
SOURce:CURRent: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
SOURce:CURRent:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
SOURce:CURRent:RAMP:ABORt	Aborts ramping and clears trigger mode.	N
SOURce:CURRent:RAMP:ALL?	Returns the ramping status.	N
SOURce:CURRent:RAMP:HTRIGgered <NRf> <NRf>	Sets the value of the output current to ramp to be implemented when the hardware trigger is received.	N
SOURce:CURRent:RAMP:HTRIGgered?	Returns the value of the output current to ramp to be implemented when the hardware trigger is received.	N
SOURce:CURRent:RAMP: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.	C
SOURce:CURRent:RAMP: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
SOURce:CURRent:TRIGgered <NRf>	Sets the output current to the values <NRf> when TRIGger:TYPe 2 or 3 is sent	C
SOURce:CURRent:TRIGgered?	Returns the current level that will be set upon receipt of the trigger.	C
SOURce:CURRent:TRIGgered[:AMPLitude] <NRf>	Sets the output current to the values <NRf> when TRIGger:TYPe 2 or 3 is sent	C
SOURce:CURRent:TRIGgered[:AMPLitude]?	Returns the current level that will be set upon receipt of the trigger.	C
SOURce:CURRent:TRIGgered:CLEar	Clears the value stored by the SOURce:CURRent:TRIGger:AMPLitude command.	C
SOURce:CURRent:PROGram <string>	Changes the Current programming mode of the supply. Valid arguments are: INT/0 (Internal Digital Current programming) EXT/1 (External analog Current programming) ADD/2 (Add Internal and External Current programming values)	N

Command	Description	SCPI
SOURce:CURRent:PROGram?	Returns the setting of Current programming mode	N
SOURce:CURRent:PROGram:FSC <NRf>	Sets the Full Scale voltage, at which Full scale Current will be programmed in external Current programming Mode with voltage as programming source. Valid Range is from 2 to 10V.	N
SOURce:CURRent:PROGram:FSC?	Returns the Full-scale Voltage, at which Full scale Current will be programmed.	N
SOURce:CURRent:PROGram:FSCR <NRf>	Sets the Full Scale resistance, at which Full scale Current will be programmed in external Current programming Mode with Current(Resistance) as programming source. Valid Range is from 2 to 10kOhm.	N
SOURce:CURRent:PROGram:FSCR?	Returns the Full-scale Resistance, at which Full scale Current will be programmed.	N
SOURce:CURRent:MONitor:FSC <NRf>	Sets Full Scale voltage on Current monitor pin (IMON), when power supply is producing full scale output current.	N
SOURce:CURRent:MONitor:FSC?	Returns the full scale voltage set for Current monitor pin (IMON)	N
SOURce:VOLTage <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce:VOLTage?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce:VOLTage[:LEVel] <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce:VOLTage[:LEVel]?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce:VOLTage[:LEVel][:IMMediate] <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce:VOLTage[:LEVel][:IMMediate]?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce:VOLTage[:LEVel][:IMMediate] [:AMPLitude] <NRf>	Sets the output voltage of the supply in amps (default) or in milliamps.	C
SOURce:VOLTage[:LEVel][:IMMediate] [:AMPLitude?]	Returns the output voltage of the supply in amps or in milliamps.	C
SOURce:VOLTage:LIMit <NRf>	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
SOURce:VOLTage:LIMit?	Returns the upper soft limit set on the programmed output voltage.	C
SOURce:VOLTage:LIMit[:AMPLitude] <NRf>	Sets the upper soft limit on the programmed output voltage.	C
SOURce:VOLTage:LIMit[:AMPLitude]?	Returns the upper soft limit on the programmed output voltage.	C
SOURce:VOLTage:PROTection <NRf>	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C
SOURce:VOLTage:PROTection?	Returns the set overvoltage protection trip point in volts (default) or in millivolts.	C
SOURce:VOLTage:PROTection[:LEVel] <NRf>	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C

Command	Description	SCPI
SOURce:VOLTage:PROTection[:LEVel]?	Returns the set overvoltage protection trip point in volts or in millivolts.	C
SOURce:VOLTage:PROTection:TRIPped?	Returns 1 (TRIPPED) or 0 (UNTRIPPED) state of the overvoltage protection circuit.	C
SOURce:VOLTage:PROTection:STATe?	Returns the state 1 (ON) or 0 (OFF) If the overvoltage protection is enabled.	C
SOURce:VOLTage:PROTection:CLEar	Clears the overvoltage protection circuit.	C
SOURce:VOLTage:PROTection:PROGram<string>	Changes the Over Voltage Programming mode of the supply. Valid arguments are: INT/0 (Internal Digital Over voltage programming) EXT/1 (External analog Over voltage programming)	N
SOURce:VOLTage:PROTection:PROGram?	Returns the setting of Over voltage Programming mode	N
SOURce:VOLTage:PROTection:PROGram:FSC <NRf>	Sets the Full-Scale voltage, at which Full Scale Over Voltage trip point will be programmed in external Over voltage Programming Mode. Valid voltage Range is from 2 to 10V.	N
SOURce:VOLTage:PROTection:PROGram:FSC?	Returns the Full-scale Voltage, at which Full Scale Over voltage trip point will be programmed.	N
SOURce:VOLTage: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 10.5.3.	N
SOURce:VOLTage:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
SOURce:VOLTage:RAMP:ABORt	Aborts ramping and clears trigger mode.	N
SOURce:VOLTage:ALL?	Returns the ramping status.	N
SOURce:VOLTage:RAMP:HTRIGgered <NRf><NRf>	Sets the value of the output voltage ramp to be implemented when the hardware trigger is received.	N
SOURce:VOLTage:RAMP:HTRIGgered?	Returns the value of the output voltage ramp to be implemented when the hardware trigger is received.	N

Command	Description	SCPI
SOURce:VOLTage:RAMP: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
SOURce:VOLTage:RAMP:TRIGgered?	Returns the output voltage to ramp	N
SOURce:VOLTage:TRIGgered <NRf>	Sets the output voltage to the values stored by sending the command TRIGger:TYPe 1 or 3.	C
SOURce:VOLTage:TRIGgered?	Returns the voltage level that will be set upon receipt of the trigger.	C
SOURce:VOLTage:TRIGgered:CLEar	Clears the value stored by the SOURce:VOLTage:TRIGger:AMPLitude command.	C
SOURce:VOLTage:TRIGgered:[AMPLitude] <NRf>	Sets the output voltage to the values stored by sending the command TRIGger:TYPe 1 or 3.	C
SOURce:VOLTage:TRIGgered:[AMPLitude]?	Returns the stored value of the output current to be set when the SOURce:VOLTage:TRIGGered command is sent.	C
SOURce:VOLTage:PROGram <string>	Changes the Voltage programming mode of the supply. Valid arguments are: INT/0 (Internal Digital Voltage programming) EXT/1 (External analog Voltage programming) ADD/2 (Add Internal and External Voltage programming values)	N
SOURce:VOLTage:PROGram?	Returns the setting of Voltage programming mode	N
SOURce:VOLTage:PROGram:FSC <NRf>	Sets the Full Scale voltage, at which Full scale Voltage will be programmed in external Voltage programming Mode with voltage as programming source. Valid Range is from 2 to 10V.	N
SOURce:VOLTage:PROGram:FSC?	Returns the Full-scale Voltage, at which Full scale Voltage will be programmed.	N
SOURce:VOLTage:PROGram:FSCR <NRf>	Sets the Full Scale resistance, at which Full scale Voltage will be programmed in external Voltage programming Mode with Current(Resistance) as programming source. Valid Range is from 2 to 10kOhm.	N
SOURce:VOLTage:PROGram:FSCR?	Returns the Full-scale Resistance, at which Full scale Voltage will be programmed.	N
SOURce:VOLTage:MONitor:FSC <NRf>	Sets Full Scale voltage on voltage monitor pin(VMON), when power supply is producing full scale output voltage.	
SOURce:VOLTage:MONitor:FSC?	Returns the full scale voltage set for Voltage monitor pin (VMON)	
SOURce:POWer <NRf>	Sets the maximum power limit	C
SOURce:POWer?	Returns the set power.	C
SOURce:POWer:MAXimum?	Returns the Maximum power that can be programmed with given hardware configuration and Input voltage conditions.	N
SOURce:POWer:LIMit <NRf>	Sets an upper soft limit on the programmed output power for the supply. The soft limit prevents the supply from being inadvertently programmed above the soft limit, thus providing	N

Command	Description	SCPI
	a method for protecting the load against damaging power.	
SOURce:POWer:LIMit?	Returns the upper soft limit on the programmed output power for the supply.	N
SOUR:POWer:PROGram <string>	Changes the Power programming mode of the supply. Valid arguments are: INT/0 (Internal Digital Power programming) EXT/1 (External analog Power programming) ADD/2 (Add Internal and External Power programming values)	N
SOUR:POWer:PROGram?	Returns the setting of Power programming mode	N
SOUR:POWer:PROGram:FSC <NRf>	Sets the Full-Scale voltage, at which Full Scale Power will be programmed in external Power Programming Mode with voltage as programming source. Valid Range is from 2 to 10V.	N
SOUR:POWer:PROGram:FSC?	Returns the Full-scale Voltage, at which Full Scale Power will be programmed.	N
SOUR:POWer:PROGram:FSCR <NRf>	Sets the Full-Scale resistance, at which Full Scale Power will be programmed in external Power Programming Mode with Current(Resistance) as programming source. Valid Range is from 2 to 10kOhm.	N
SOUR:POWer:PROGram:FSCR?	Returns the Full-scale Resistance, at which Full Scale Power will be programmed.	N
SOUR:POWer:MONitor:FSC <NRf>	Sets Full Scale voltage on power monitor pin (PMON), when power supply is producing full scale output power.	N
SOUR:POWer:MONitor:FSC?	Returns the full scale voltage set for Power monitor pin (PMON)	N
SOURce:DIGINP:DIN1?	Returns the status of digital input 1 at the Remote Analog Programming connector	N
SOURce:DIGINP:DIN2?	Returns the status of digital input 2 at the Remote Analog Programming connector	N
SOURce:DIGINP:DIN3?	Returns the status of digital input 3 at the Remote Analog Programming connector (OUT_RY_EN / DIN3)	N
SOURce:DIGINP:DIN4?	Returns the status of digital input 4 at the Remote Analog Programming connector (REV_RY_EN / DIN4)	N

10.5.3 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.

10.5.3.1 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.

10.5.3.2 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.

10.5.3.3 CURRENT RAMPING

Requires an appropriate resistive load.

10.5.4 MEASURE SCPI Command Subsystem

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

10.5.5 MEASURE SCPI Command Summary

```
MEASure
      :CURRent
      :CURRent?
            :AVErage <NR1>
            :AVErage?
            :PROGram?
      :POWer?
            :PROGram?
      :VOLTage?8
      :VOLTage
            :AVErage <NR1>
            :AVErage?
            :PROGram?
            :PROTection
                  :PROGram?
```

10.5.6 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:CURRent?	Returns the floating point value of the DC output current in amps.	C
MEASure:CURRent:AVErage <NR1>	Enter a value of 1 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 1 (factory default) provides the fastest response time in the readings, but less rejection of noise.	N
MEASure:CURRent:AVErage?	Returns the number 1 to 9 to indicate the number of readings to average together when taking a current reading.	N
MEASure:CURRent:PROGram?	Returns the programmed output current from external Analog current programming feature.	
MEASure:POWer?	Returns the floating point value of the measured output power in watts.	
MEASure:PROGram?	Returns the programmed output power from external Analog power programming feature.	
MEASure:VOLTage?	Returns the floating point value of the DC output voltage in volts.	C
MEASure:VOLTage:AVErage <NR1>	Enter a value of 1 to 9 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	N

Command	Description	SCPI
	default) provides the fastest response time in the readings, but less rejection of noise.	
`MEASure:VOLTage:AVErage?`	Returns the number 1 to 9 to indicate the last set number of readings to average together when taking a voltage reading.	N
`MEASure:VOLTage:PROGram?`	Returns the programmed output voltage from external Analog voltage programming feature.	
`MEASure:VOLTage:PROTection :PROGram?`	Returns the programmed Over voltage trip point from external Analog over voltage programming feature.	

10.6 OUTPUT SCPI COMMAND SUBSYSTEM

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

10.6.1 OUTPUT SCPI Command Summary

```
OUTPut
     :PROTection
          :DELay <NRf>
          :DELay?
          :FOLD <0|1|2>
          :FOLD?
     :SENSe <boolean>
     :SENSe?
          :DEFault <Boolean>
          :DEFault?
     :STATe <boolean>
     :STATe?
     :TRIPped?
     :AUX
          :5V <ON|OFF>
          :5V?
          :15V <ON|OFF>
          :15V?
     :ISOLation <Boolean>
     :ISOLation?
     :POLarity <string>
     :POLarity?
```

10.6.2 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:PROTection: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
OUTPut:PROTection:DELay?	Returns the time delay to be executed by the supply.	N
OUTPut:PROTection:FOLD <0|1|2|3>	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 constant power mode), or 2 (program down to zero upon entering constant-current mode or constant power mode). 3 (program down to zero upon entering constant-current mode or constant-voltage mode)	N
OUTPut:PROTection: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 or constant power mode. 2 = will program down to zero upon entering constant current mode or constant power mode 3 = program down to zero upon entering constant-current mode or constant-voltage mode	N
OUTPut:PROTection:SENSe <boolean>	Sets the output voltage sense signal setting. Valid arguments are 1/REMOTE or 0/LOCAL. When REMOTE option is selected, voltage sense signal must be connected at RVS connector at the rear side of power supply.	N
OUTPut:PROTection:SENSe?	Returns the setting of the output voltage sense signal.	N
OUTPut:PROTection:SENSe:DEFault <Boolean>	Sets the Power ON setting for output voltage sense signal LOCAL or REMOTE. Valid arguments are 1/REMOTE or 0/LOCAL.	N
OUTPut:PROTection:SENSe:DEFault?	Returns the Power ON setting of the output voltage sense signal.	N
OUTPut: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
OUTPut:STATe?	Returns the state of the output: 1 = ON 0 = OFF	C
OUTPut:TRIPped?	Returns the integer value 1 (TRIPPED) or 0 (UNTRIPPED) state of the output.	N
OUTPut:AUX	Subsystem for controlling Auxiliary supply	N

<table>
<tr><th>Command</th><th>Description</th><th>SCPI</th></tr>
<tr><td>OUTPutAUX:5V <ON|OFF></td><td>Turns ON or OFF, 5V auxiliary supply at the Remote Analog Programming connector</td><td>N</td></tr>
<tr><td>OUTPutAUX:5V?</td><td>Returns the status of 5V auxiliary supply at the Remote Analog Programming connector</td><td>N</td></tr>
<tr><td>OUTPutAUX:15V <ON|OFF></td><td>Turns ON or OFF, 15V auxiliary supply at the Remote Analog Programming connector</td><td>N</td></tr>
<tr><td>OUTPutAUX:15V?</td><td>Returns the status of 15V auxiliary supply at the Remote Analog Programming connector</td><td>N</td></tr>
<tr><td>OUTPut:ISOLation <Boolean></td><td>Sets the rear panel isolation relay control signal ON or OFF. Valid arguments are 1/ON or 0/OFF.</td><td>N</td></tr>
<tr><td>OUTPut:ISOLation?</td><td>Returns the state of the rear panel isolation relay control signal: 1 = ON 0 = OFF</td><td>N</td></tr>
<tr><td>OUTPut:POLarity<NORM/0/OFF|INV/1/O></td><td>Changes the state of the polarity relay signal. This command requires that the isolation relay be open beforehand</td><td>C</td></tr>
<tr><td>OUTPut:POLarity?</td><td>Returns the state of the polarity relay: <NORM|INV></td><td>C</td></tr>
</table>

10.7 SYSTEM SCPI COMMAND SUBSYSTEM

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

10.7.1 SYSTEM SCPI Command Summary

```
SYSTem
     :ERRor?
     :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?
:ENUM
      :RST
      :COUNT?
```

10.7.2 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: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
SYSTem: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.	N
SYSTem:LOCAL?	Returns ON or 1 if in local mode. Returns OFF or 0 if in remote mode.	N
SYST:NET:AUTOIP <boolean>	Sets the network Auto IP mode in the Primary configuration without affecting the Secondary configuration.. 0 = disable AutoIP; 1 = enable AutoIP	N
SYST:NET:AUTOIP?	Returns 1 if AutoIP is enabled in the Primary configuration. Returns 0 if AutoIP is disabled in the Primary configuration.	N
SYST:NET:DESC <string>	Set the network Description, a 36 character alphanumeric string	N
SYST:NET:DESC?	Returns the network Description.	N
SYST:NET:DHCPMODE <boolean>	Sets the network DHCP Mode in the Primary configuration without affecting the Secondary configuration. 0 = disable DHCP; 1 = enable DHCP	N
SYST:NET:DHCPMODE?	Returns 1 if DHCP Mode is enabled in the Primary configuration. Returns 0 if DHCP mode is disabled in the Primary configuration.	N
SYST:NET: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
SYST:NET:DNS?	Returns the network DNS address for the device.	N
SYST:NET: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
SYST:NET:GATE?	Returns the network gateway IP address for the device.	N
SYST:NET:HOST <string>	Set the network Host Name, a 15-character (maximum) alphanumeric string. (Must be limited to 15 characters for LXI compliance)	N
SYST:NET:HOST?	Returns the network Host Name	N

Command	Description	SCPI
SYST:NET: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
SYST:NET: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
SYST:NET:LANLED:BLINK <string>	ON changes front panel screen to device identify. Refer to Figure 6-11 OFF changes to dashboard screen.	N
SYST:NET:MAC?	Returns the network MAC address. xx:xx:xx:xx:xx:xx (Hexadecimal digit pairs)	N
SYST:NET: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
SYST:NET:MASK?	Returns the network Subnet Mask for the device.	N
SYST:NET: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
SYST:NET:PORT <NRf>	Set the network TCP/IP socket listening port. Valid values are 1025 to 65535.	N
SYST:NET:PORT?	Returns the network TCP/IP socket listening port.	N
SYST:NET: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
SYST:NET:TERM?	Returns the string terminators to be used by the device.	N
SYST:ENUM:RST	Re-assigns the chassis address	N
SYST:ENUM:COUNT?	Returns the number of chassis in connection for parallel or multi-chassis communication.	N

10.8 HTRIGGER SCPI COMMAND SUBSYSTEM

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.

10.8.1 HTRIGGER SCPI Command Summary

```
HTRIGger
      :ABORt
```

10.8.2 HTRIGGER SCPI Command Reference

Command	Description	SCPI
`HTRIGger: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

10.9 TRIGGER SCPI COMMAND SUBSYSTEM

This section describes the programming soft trigger function.

10.9.1 TRIGGER SCPI Command Summary

```
TRIGger
      :ABORt
      :RAMP
      :TYPE <1|2|3>
```

10.9.2 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:ABORt`	Stops the execution of a currently running trigger function, and clears all settings of voltage and current.	N

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

`TRIGger: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

10.10 CALIBRATION SCPI COMMAND SUBSYSTEM

Note: *See section 12 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.

10.10.1 CALIBRATION SCPI Command Summary

```
CALibrate
     :INITial
          :CURRent <NRf>
          :CURRent?
               :LIMit <NRf>
               :LIMit?
               :PROGram <string>
               :PROGram?
                    :FSC <NRf>
                    :FSC?
                    :FSCR <NRf>
                    :FSCR?
               :MONitor
                    :FSC <NRf>
                    :FSC?
          :VOLTage <NRf>
          :VOLTage?
               [:AMPLitude] <NRf>
               [:AMPLitude?]
               :PROTection <NRf>
               :PROTection?
               :LIMit <NRf>
               :LIMit?
               :PROGram <string>
               :PROGram?
                    :SOURce <string>
                    :SOURce?
                    :FSC <NRf>
                    :FSC?
                    :FSCR <NRf>
                    :FSCR?
               :MONitor
                    :FSC <NRf>
                    :FSC?
```

```
CALibrate
      :INITial
            :POWer <NRf>
            :POWer?
                  :LIMit <NRf>
                  :LIMit?
                  :PROGram <string>
                  :PROGram?
                        :FSC <NRf>
                        :FSC?
                        :FSCR <NRf>
                        :FSCR?
                  :MONitor
                        :FSC <NRf>
                        :FSC?
            :MEASure:CURRent:AVErage <NR1>
            :MEASure:CURRent:AVErage?
            :MEASure:VOLTage:AVErage <NR1>
            :MEASure:VOLTage:AVErage?
            :OUTPut:PROTection:FOLD <NR1>
            :OUTPut:PROTection:FOLD?

CALibrate
      :MEASure
            :CURRent
                  :CALCulate
                  :GAIN <NRf>
                  :GAIN?
                  :OFFSet <NRf>
                  :OFFSet?
            :VOLTage
                  :CALCulate
                  :GAIN <NRf>
                  :GAIN?
                  :OFFSet <NRf>
                  :OFFSet?
      :MODel
            :LASTCALDATE <string>
            :LASTCALDATE?
            :NEXTCALDATE <string>
            :NEXTCALDATE?
            :POWERON <string>
            :POWERON?
            :RESET <string>
            :RESET?
```

```
CALibrate
      :OUTPut
            :CURRent
                  :CALCulate
                  :DAC <NR1>
                  :FIVEPOINT <1|2|3|4|5> <NRf>
                  :FIVEPOINT?
                  :GAIN <NRf>
                  :GAIN?
                  :OFFSet <NRf>
                  :OFFSet?
                  :EXTV
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXTI
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXT420MA
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :MON
                        :OFSET <NRf>
                        :FSC <NRf>
            :VOLTage
                  :CALCulate
                  :DAC <NR1>
                  :FIVEPOINT <1|2|3|4|5> <NRf>
                  :FIVEPOINT?
                  :GAIN <NRf>
                  :GAIN?
                  :OFFSet <NRf>
                  :OFFSet?
                  :EXTV
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXTI
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXT420MA
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :PROTection
                        :DAC <NR1>
                  :MON
                        :OFSET <NRf>
                        :FSC <NRf>
```

```
CALibrate
      :OUTPut
            :POWer
                  :EXTV
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXTI
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :EXT420MA
                        :OFFSET <NRf>
                        :FSC <NRf>
                  :MON
                        :OFSET <NRf>
                        :FSC <NRf>
            :OVERVOLT
                  :EXTV
                        :OFFSET <NRf>
                        :FSC <NRf>
      :UNLock <string>
      :STORe
      :LOCK
```

10.10.2 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:INITial	This command subsection is used to store next power up values or Initialize calibration in specified cases	N
CALibrate:INITial:CURRent <NRf>	Sets the power-on default value of current.	N
CALibrate:INITial:CURRent?	Returns the default value of power-on current.	N
CALibrate:INITial:CURRent:LIMit <NRf>	Sets the power-on default value of current limit.	N
CALibrate:INITial:CURRent:LIMit?	Returns the default value of power-on current limit.	N
CALibrate:INITial:CURRent:PROGram <string>	Sets the power-on default setting for output current programming. Valid arguments - INT, EXT or ADD	N
CALibrate:INITial:CURRent:PROGram:FSC <NRf>	Sets the power-on default voltage value for full scale current from external analog programming.	N
CALibrate:INITial:CURRent:PROGram:FSC?	Returns the power-on default full scale voltage value.	N
CALibrate:INITial:CURRent:PROGram:FSCR <NRf>	Sets the power-on default resistance value for full scale current from external analog programming.	N

Command	Description	SCPI
CALibrate:INITial:CURRent:PROGram:FSCR?	Returns the power-on default full scale resistance value.	N
CALibrate:INITial:CURRent:PROGram?	Returns the default setting of power-on output current programming.	N
CALibrate:INITial:CURRent:MONitor	Initializes the current monitor (IMON) signal calibration	N
CALibrate:INITial:CURRent:MONitor:FSC <NRf>	Sets the power-on default voltage on IMON signal for full scale output current.	N
CALibrate:INITial:CURRent:MONitor:FSC?	Returns the power-on default full scale voltage value on IMON signal.	N
CALibrate:INITial:VOLTage <NRf>	Sets the power-on default voltage.	N
CALibrate:INITial:VOLTage?	Returns the default value of power-on voltage	N
CALibrate:INITial:VOLTage [:AMPLitude] <NRf>	Sets the power-on default voltage.	N
CALibrate:INITial:VOLTage [:AMPLitude]?	Returns the power-on default voltage.	N
CALibrate:INITial:VOLTage:PROTection <NRf>	Sets the power-on default value of the overvoltage protection.	N
CALibrate:INITial:VOLTage:PROTection?	Returns the default value of the power-on overvoltage protection.	N
CALibrate:INITial:VOLTage:LIMit <NRf>	Sets the power-on default value of voltage limit.	N
CALibrate:INITial:VOLTage:LIMit?	Returns the default value of power-on current limit.	N
CALibrate:INITial:VOLTage:PROGram?	Returns the default setting of power-on output voltage programming.	N
CALibrate:INITial:VOLTage:PROGram	Program subsystem	N
CALibrate:INITial:VOLTage:PROGram:SOURce <string>	Sets the power-on default source for external analog programming for Voltage Current and Power. Valid arguments- VOLT, CURR or MA420	N
CALibrate:INITial:VOLTage:PROGram:SOURce?	Returns the default power-on source for external analog programming.	N
CALibrate:INITial:VOLTage:PROGram:FSC <NRf>	Sets the power-on default voltage value for full scale voltage from external analog programming.	N
CALibrate:INITial:VOLTage:PROGram:FSC?	Returns the power-on default full scale voltage value.	N
CALibrate:INITial:VOLTage:PROGram:FSCR <NRf>	Sets the power-on default resistance value for full scale voltage from external analog programming.	N
CALibrate:INITial:VOLTage:PROGram:FSCR?	Returns the power-on default full scale resistance value.	N
CALibrate:INITial:VOLTage:MONitor	Initializes the voltage monitor (VMON) signal calibration	N
CALibrate:INITial:VOLTage:MONitor:FSC <NRf>	Sets the power-on default voltage on VMON signal for full scale output voltage.	N
CALibrate:INITial:VOLTage:MONitor:FSC?	Returns the power-on default full scale voltage value on VMON signal.	N

Command	Description	SCPI
CALibrate:INITial:POWer <NRf>	Sets the power-on default value of power.	N
CALibrate:INITial:POWer?	Returns the default value of power-on power.	N
CALibrate:INITial:POWer:LIMit <NRf>	Sets the power-on default value of power limit.	N
CALibrate:INITial:POWer:LIMit?	Returns the default value of power-on power limit.	N
CALibrate:INITial:POWer:PROGram <string>	Sets the power-on default setting for output power programming. Valid arguments - INT, EXT or ADD	N
CALibrate:INITial:POWer:PROGram?	Returns the default setting of power-on output power programming.	N
CALibrate:INITial:POWer:PROGram:FSC <NRf>	Sets the power-on default voltage value for full scale power from external analog programming.	N
CALibrate:INITial:POWer:PROGram:FSC?	Returns the power-on default full scale voltage value.	N
CALibrate:INITial:POWer:PROGram:FSCR <NRf>	Sets the power-on default resistance value for full scale power from external analog programming.	N
CALibrate:INITial:POWer:PROGram:FSCR?	Returns the power-on default full scale resistance value.	N
CALibrate:INITial:POWer:MONitor	Initializes the power monitor (PMON) signal calibration	N
CALibrate:INITial:POWer:MONitor:FSC <NRf>	Sets the power-on default voltage on PMON signal for full scale output power.	N
CALibrate:INITial:POWer:MONitor:FSC?	Returns the power-on default full scale voltage value on PMON signal.	N
CALibrate:INITial: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
CALibrate:INITial:MEASure:CURRent:AVErage?	Returns the number 3 to 9 to indicate the number of readings to average together when taking a current reading.	N
CALibrate:INITial:MEASure:VOLTage:AVErage <NR1>	Sets the number of readings to average together when returning the voltage value with the MEAS:VOLT? command to reduce noise in the readback readings. Enter a value of 1 to 10, with the value of 1 (factory default) providing the fastest response time in the readings, but less rejection of noise.	N

Command	Description	SCPI
CALibrate:INITial:MEASure:VOLTage:AVErage?	Returns the number 1 to 10 to indicate the number of readings to average together when taking a current reading.	N
CALibrate:INITial:OUTPut:PROTection:FOLD <NR1>	Sets the power-on default foldback protection setting. Valid arguments are same as for OUTP:PROT:FOLD	N
CALibrate:INITial:OUTPut:PROTection:FOLD?	Returns the power-on default setting of foldback protection	N
CALibrate:MEASure	Calibrate Measure subsystem	N
CALibrate:MEASure:CURRent	Calibrate Measure Current subsystem	N
CALibrate:MEASure:CURRent:CALCulate	Calculates the value of the gain and offset for current measurements.	N
CALibrate:MEASure:CURRent:GAIN <NRf>	Sets the value of the gain for current measurements.	N
CALibrate:MEASure:CURRent:GAIN?	Returns the value of the gain for current measurements.	N
CALibrate:MEASure:CURRent:OFFSet <NRf>	Sets the value of the offset for current measurements.	N
CALibrate:MEASure:CURRent:OFFSet?	Returns the value of the offset for current measurements.	N
CALibrate:MEASure:VOLTage	Calibrate Measure Voltage subsystem	N
CALibrate:MEASure:VOLTage:CALCulate	Calculates the value of the gain and offset for voltage measurements.	N
CALibrate:MEASure:VOLTage:GAIN <NRf>	Sets the value of the gain for voltage measurements.	N
CALibrate:MEASure:VOLTage:GAIN?	Returns the value of the gain for voltage measurements.	N
CALibrate:MEASure:VOLTage:OFFSet <NRf>	Sets the value of the offset for voltage measurements.	N
CALibrate:MEASure:VOLTage:OFFSet?	Returns the value of the offset for the voltage measurements.	N
CALibrate:MODel	Calibrate Model subsystem	N
CALibrate:MODel:LASTCALDATE<string>	Sets the date last calibrated; format: MM DD YYYY (space after MM and DD required)	N
CALibrate:MODel:LASTCALDATE?	Returns the date last calibrated.	N
CALibrate:MODel:NEXTCALDATE<string>	Sets the date next calibration is required; format: MM DD YYYY (space after MM and DD required)	N
CALibrate:MODel:NEXTCALDATE?	Returns the date next calibration is required.	N

Command	Description	SCPI
CALibrate:MODel: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
CALibrate:MODel:POWERON?	Returns the status of the output enable condition at power on	N
CALibrate:MODel: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
CALibrate:MODel:RESET?	Returns the status of the default output enable condition when the *RST command is issued	N
CALibrate:OUTPut	Calibrate Output subsystem	N
CALibrate:OUTPut:CURRent	Calibrate Output Current subsystem	N
CALibrate:OUTPut:CURRent:CALCulate	Calculates the value of the gain and offset for output current.	N
CALibrate:OUTPut:CURRent:DAC <NR1>	Sets the output of the output current D/A converter.	N
CALibrate:OUTPut:CURRent:FIVEPOINT <1|2|3|4|5> <NRf>	Sets output current value for each calibration point (1-5)	N
CALibrate:OUTPut:CURRent:FIVEPOINT?	Returns the entered values for 5-point calibration.	N
CALibrate:OUTPut:CURRent:GAIN <NRf>	Sets the value of the gain for the output current.	N
CALibrate:OUTPut:CURRent:GAIN?	Returns the value of the gain for the output current.	N
CALibrate:OUTPut:CURRent:OFFSet <NRf>	Sets the value of the offset for the output current.	N
CALibrate:OUTPut:CURRent:OFFSet?	Returns the value of the offset for the output current.	N
CALibrate:OUTPut:CURRent:EXTV:OFFSET <NRf>	Sets the calibration Offset point for current programming from external voltage source	N
CALibrate:OUTPut:CURRent:EXTV:FSC <NRf>	Sets the calibration full-scale point for current programming from external voltage source	N
CALibrate:OUTPut:CURRent:EXTI:OFFSET <NRf>	Sets the calibration Offset point for current programming from external resistance source	N
CALibrate:OUTPut:CURRent:EXTI:FSC <NRf>	Sets the calibration full-scale point for current programming from external resistance source	N

Command	Description	SCPI
CALibrate:OUTPut:CURRent:EXT420MA:OFFSET <NRf>	Sets the calibration Offset point for current programming from external 4-20mA source	N
CALibrate:OUTPut:CURRent:EXT420MA:FSC <NRf>	Sets the calibration full-scale point for current programming from external 4-20mA source	N
CALibrate:OUTPut:CURRent:MON:OFFSET <NRf>	Sets the calibration Offset point for current monitor signal	N
CALibrate:OUTPut:CURRent:MON:FSC <NRf>	Sets the calibration full-scale point for current monitor signal	N
CALibrate:OUTPut:VOLTage	Calibrate Output Voltage subsystem	N
CALibrate:OUTPut:VOLTage:CALCulate	Calculates the value of the gain and offset for output voltage.	N
CALibrate:OUTPut:VOLTage:DAC <NR1>	Sets the output of the output voltage D/A converter.	N
CALibrate:OUTPut:VOLTage:FIVEPOINT <1|2|3|4|5> <NRf>	Sets output voltage value for each calibration point (1-5)	N
CALibrate:OUTPut:VOLTage:FIVEPOINT?	Returns the entered values for 5-point calibration.	N
CALibrate:OUTPut:VOLTage:GAIN <NRf>	Sets the value of the gain for the output voltage.	N
CALibrate:OUTPut:VOLTage:GAIN?	Returns the value of the gain for the output voltage.	N
CALibrate:OUTPut:VOLTage:OFFSet <NRf>	Sets the value of the offset for the output voltage.	N
CALibrate:OUTPut:VOLTage:OFFSet?	Returns the value of the offset for the output voltage.	N
CALibrate:OUTPut:VOLTage:EXTV:OFFSET <NRf>	Sets the calibration Offset point for voltage programming from external voltage source	N
CALibrate:OUTPut:VOLTage:EXTV:FSC <NRf>	Sets the calibration full-scale point for voltage programming from external voltage source	N
CALibrate:OUTPut:VOLTage:EXTI:OFFSET <NRf>	Sets the calibration Offset point for voltage programming from external resistance source	N
CALibrate:OUTPut:VOLTage:EXTI:FSC <NRf>	Sets the calibration full-scale point for voltage programming from external resistance source	N
CALibrate:OUTPut:VOLTage:EXT420MA:OFFSET <NRf>	Sets the calibration Offset point for voltage programming from external 4-20mA source	N
CALibrate:OUTPut:VOLTage:EXT420MA:FSC <NRf>	Sets the calibration full-scale point for voltage programming from external 4-20mA source	N
CALibrate:OUTPut: VOLTage:MON:OFFSET <NRf>	Sets the calibration Offset point for voltage monitor signal	N
CALibrate:OUTPut:VOLTage:MON:FSC <NRf>	Sets the calibration full-scale point for voltage monitor signal	N

Command	Description	SCPI
CALibrate:OUTPut:VOLTage:PROTection:DAC <NR1>	Sets the output of the output overvoltage protection D/A converter.	N
CALibrate:OUTPut:POWer:EXTV:OFFSET <NRf>	Sets the calibration Offset point for power programming from external voltage source	N
CALibrate:OUTPut:POWer:EXTV:FSC <NRf>	Sets the calibration full-scale point for power programming from external voltage source	N
CALibrate:OUTPut:POWer:EXTI:OFFSET <NRf>	Sets the calibration Offset point for power programming from external resistance source	N
CALibrate:OUTPut:POWer:EXTI:FSC <NRf>	Sets the calibration full-scale point for power programming from external resistance source	N
CALibrate:OUTPut:POWer:EXT420MA:OFFSET <NRf>	Sets the calibration Offset point for power programming from external 4-20mA source	N
CALibrate:OUTPut:POWer:EXT420MA:FSC <NRf>	Sets the calibration full-scale point for power programming from external 4-20mA source	N
CALibrate:OUTPut:POWer:MON:OFFSET <NRf>	Sets the calibration Offset point for power monitor signal	N
CALibrate:OUTPut:POWer:MON:FSC <NRf>	Sets the calibration full-scale point for power monitor signal	N
CALibrate:OUTPut:OVERVOLT:EXTV:OFFSET <NRf>	Sets the calibration Offset point for Over-voltage programming from external voltage source	N
CALibrate:OUTPut:OVERVOLT:EXTV:FSC <NRf>	Sets the calibration full-scale point for Over-voltage programming from external voltage source	N
CALibrate:UNLock <string>	Sets the non-volatile memory available to store calibration constants. The access string is "6867".	N
CALibrate:STORe	Stores the calibration constants in non-volatile memory.	N
CALibrate:LOCK	Disables access to the non-volatile memory. Prevents attempts to store calibration values. (Issue after CAL:UNLock and CAL:STORe commands)	N

10.12 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.*

10.12.1 STATUS SCPI Command Summary

```
STATus
      :PROTection
            :CONDition?
            :ENABle <NR1>
            :ENABle?
            :EVENt?
```

10.12.2 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:PROTection	Status Protection subsystem.	C
STATus:PROTection :CONDition?	Returns the integer value of the Protection Condition Register. Used to read the status of the power hardware. See section 10.2.1 for a detailed table of the various bits that make up this register.	C
STATus:PROTection :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
STATus:PROTection :ENABle?	Returns the value of the current mask of the Protection Event Register.	C
STATus:PROTection :EVENt?	Returns the integer value of the Protection Event Register.	C

10.13 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).

10.13.1 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).
`OUTP:STAT 1`	//program the output state of the power supply as enabled (1).
`OUTP:STAT?`	// confirm the output state of the power supply (response: 1).
`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).

10.13.2 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.
`CAL1:UNLOCK “6867”`	// Unlock overvoltage simulation.
`*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 4.0`	// program output voltage to 4.0 VDC - cause OVP trip! // confirm that unit issued a GPIB service request (use a serial poll).

10.13.3 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.

10.13.4 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.
	// upon trigger command.
`TRIG:RAMP`	// starts ramp execution.
`TRIG:ABORT`	// turn off trigger mode.

10.13.5 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.
```

10.13.6 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.
// 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.
```

11 SEQUENCING FUNCTION

11.1 INTRODUCTION

This section describes the features and operation of Sequencing Function. Sequencing function allows the user to set up the power supply to automatically run a series of voltage, current and power mode operations. This is especially useful for setting up the supply to test to compliance standards or unburdening the test computer in automated testing applications. Through RS-232, USB, IEEE-488.2 GPIB, Ethernet or EtherCAT, an external computer can trigger the sequences. Up to 50 sequences may be stored, with each sequence containing up to 20 individual steps. With the ability to string sequences together and an extensive list of step functions such as ramping, looping, go to and subroutine calls, the user can define a nearly infinite variety of test sequences.

11.2 PROGRAM SCPI COMMAND SUBSYSTEM

This section describes the operation of the Digital Interfaces GPIB, USB, RS232, Ethernet and EtherCAT by using SCPI Command set for sequencing function. The command set comprises of a tree summary of the PROGram commands and then provides a tabular description that facilitate remote control of the power supply.

11.2.1 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?
```

11.2.2 Program SCPI Command Reference

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

Command	Description	SCPI
PROGram	Program subsystem	C
PROGram: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
PROGram:CATalog?	Lists all the defined sequence names. The format for the response to the query is a list of comma-separated test names.	C
PROGram:[SELected]	Selected section of program subsystem	C
PROGram:[SELected]: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
PROGram:[SELected]:ARM?	A return of 1 means the sequence is armed. A return of 0 means it is not.	N

PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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

PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram:DELete	Program Delete subsystem	C

PROGram:DELete: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
PROGram:DELete:ALL	This command causes all defined sequences to be deleted from ram and non-volatile memory.	C
PROGram:SAVe	The sub tree for the SAVe commands.	N
PROGram:SAVe: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
PROGram:SAVe:ALL	Saves all defines sequences to non-volatile memory.	N
PROGram: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
PROGram:EXECute VIMODE, <volts>, <amps>, <ovp>	Allows setting of active voltage, current, and ovp.	C
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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
PROGram: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>
<td>PROGram:STATe <state name></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>PROGram: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 processor copy. (Internally, the supply uses a two-processor architecture, where one processor processes the SCPI commands and the user interface, and another processor commands to control the power hardware.)

where state may be any of the following:

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

11.2.3 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: State: STOPPED

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

```
OUTP<n>:STAT  ON

PROG<n>:STATE  RUN
```

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

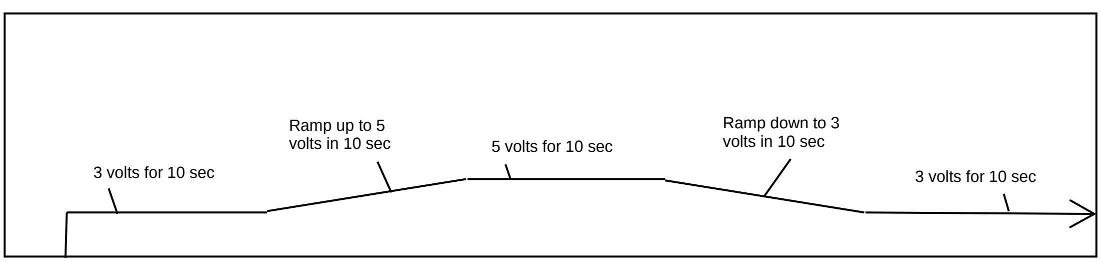


Figure 11-1: Power Supply Output for Example 1

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: State: 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:STATE  RUN
```

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

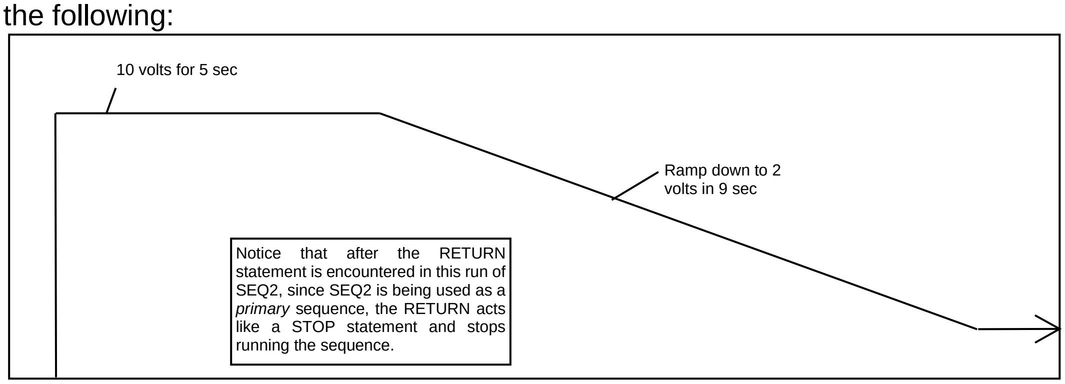


Figure 11-2: Power Supply Output for Example 2

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: State: STOPPED

```
OUTP:STAT  ON

PROG:STAT  RUN
```

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

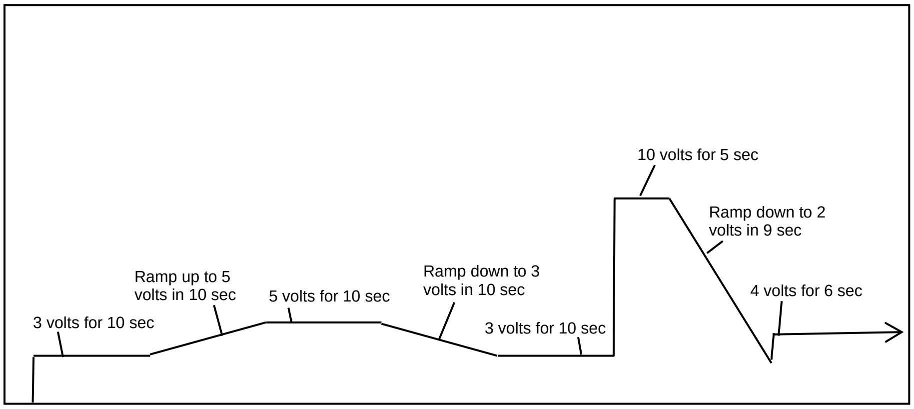


Figure 11-3: Power Supply Output for Example 3

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: State: STOPPED

```
OUTP:STAT  ON

PROG:STAT  RUN
```

Following is a diagram for Example 4.

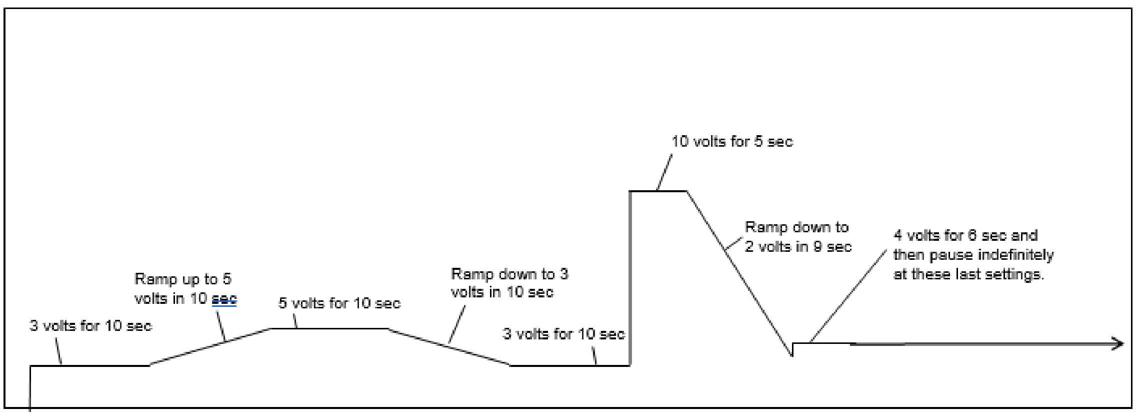


Figure 11-4: End-of-Sequence Pause for Example 4

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

`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: State: 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: State: 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:

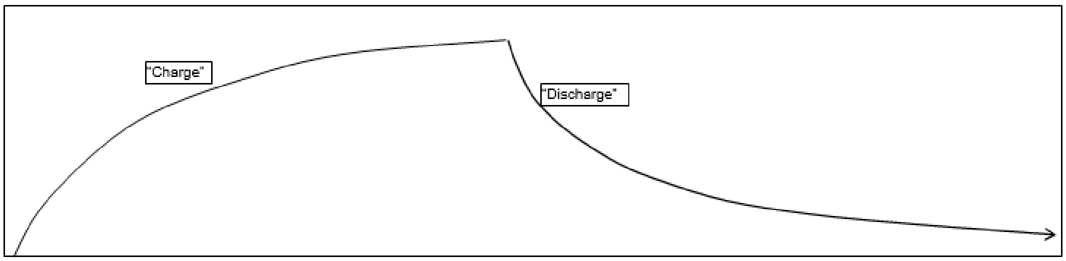


Figure 11-5: Power Supply Output Example 5

EXAMPLE 6

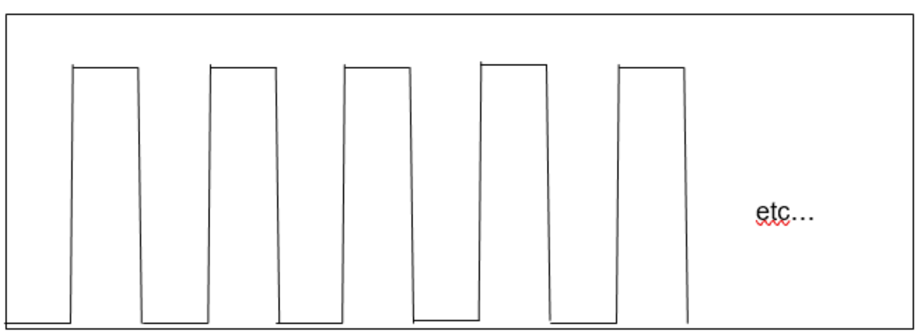
This example shows how to make an infinite loop for the creation of a continuous square wave:
Issue *RST to establish a known state.
PROG<n>:NAME “Square Wave”
PROG<n>:MALL DEFAULT
PROG<n>:DEF 1, VIMODE, 0, 5, 15, 0.5 //go to 0 volts, 5 amps, 15 volts ovp, for 0.5 seconds
PROG<n>:DEF 2, VIMODE, 10,5,15,0.5   //go to 10 volts, 5 amps, 15 volts ovp, for 0.5 seconds
PROG<n>:DEF 3,GOTO, “Square Wave”   //go to top of this sequence and loop indefinitely)
PROG<n>:STAT COMPLETE
PROG<n>:SAVE:SELPROG<n>:STAT?

Response: State: STOPPED

OUTP<n>:STAT ON

PROG<n>:STAT RUN

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



6.

Figure 11-6: Power Supply Output Example 6

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

```
PROG<n>: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<n>: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<n>:NAME "Pulse Train"

PROG<n>:MALL DEFAULT

PROG<n>:DEF 1, VIMODE, 0, 5, 15, 1

PROG<n>:DEF 2, RAMPTOV, 0, 4, 5, 15, 1

PROG<n>:DEF 3, LOOP, 10

PROG<n>:DEF 4, VIMODE, 4, 5, 15, 1

PROG<n>:DEF 5, VIMODE, 0, 5, 15, 1

PROG<n>:DEF 6, NEXT

PROG<n>:DEF 7, VIMODE, 4, 5, 15, 1

PROG<n>:DEF 8, RAMPTOV, 4, 0, 5, 15, 1

PROG<n>:DEF 9, STOP

PROG<n>:STAT  COMPLETE

PROG<n>:SAVE:SEL

PROG<n>:STAT?
```

Response: State: STOPPED

```
OUTP<n>:STAT ON

PROG<n>:STAT  RUN
```

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

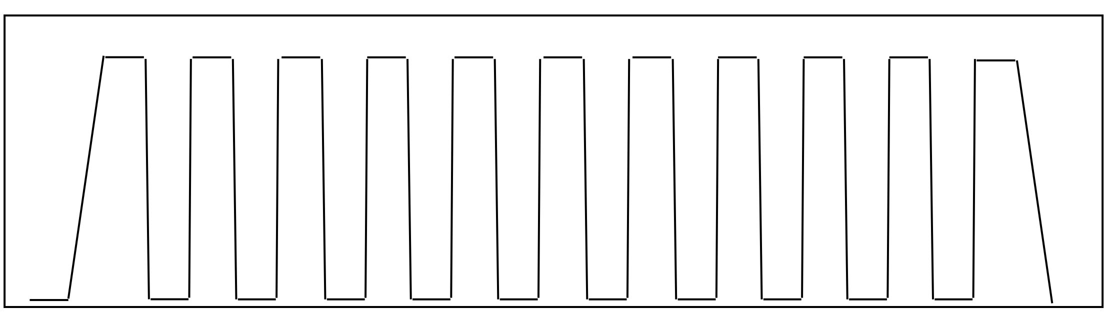


Figure 11-7: Power Supply Output for Example 7

12
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.

12.1 INTRODUCTION

The Asterion DC 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. The Asterion DC Power Supply is calibrated for output voltage programming and output current programming.

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 0.05% 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 10.10) should be issued once calibration procedures have been performed.

12.2 SETUP FOR CALIBRATION

This section provides calibration setup for RS232, Ethernet.

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 DVM to the UUT output terminals.
4. Perform one of the following:

   For **RS232**:

   Connect the RS232 controller to the rear panel of the UUT using the RS232 null modem cable (see Figure 12-1).

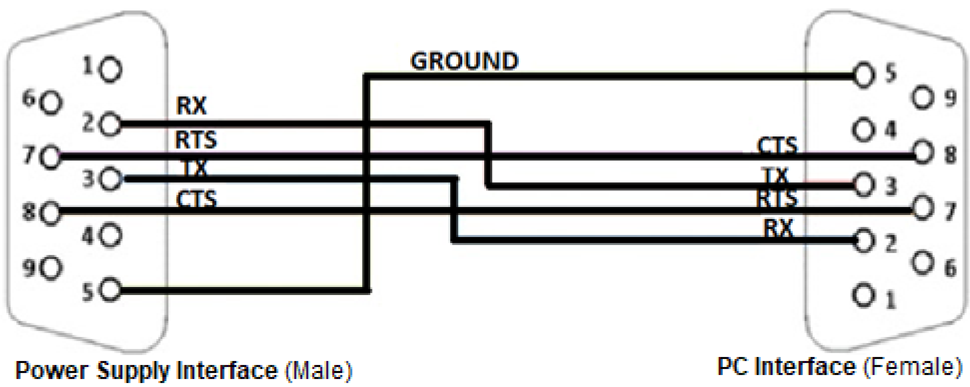


Figure 12-1: RS232 Communications Cable Pinout

   For **Ethernet**:

   Refer to Section 6 for Ethernet communication setup. Connect the RJ-45 Ethernet cable to the rear panel of the UUT.

5. 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.

12.3 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. Set the output voltage sensing to “Local”

   `OUTP:SENS LOCAL`

6. Set the output ON

   `OUTP:STAT ON`

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

   `CAL:OUTP:VOLT:DAC 3275`

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

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

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

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

    `CAL:OUTP:VOLT:DAC 19000`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

23. Send the command to calculate the calibration coefficients by the processor.

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

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

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

25. 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.

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

```
CAL:LOCK
```

12.4 VOLTAGE PROGRAMMING CALIBRATION – REMOTE SENSE

Preparation:

1. Power down the Unit Under Test (UUT) and remove the input power for safety.
2. Connect output terminals to the RVS connector on the rear side of the power supply. Connect positive terminal to positive of RVS connector and negative terminal to negative of RVS connector.
3. Connect the DVM to the UUT output terminals.
4. Apply input power and power up the UUT.
5. 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 current to full scale to prevent Constant-Current operation:

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

5. Set the output voltage sensing to “Remote”

```
OUTP:SENS REMOTE
```

6. Set the output ON

```
OUTP:STAT ON
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

23. Send the command to calculate the calibration coefficients by the processor.

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

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

```
CAL:UNLOCK “6867”
```

25. 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.

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

```
CAL:LOCK
```

12.5 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. Place the electronic load in Constant Resistance mode.
   R = (FSV * 80%) / FSC * (Present DAC value / MAX DAC value)
   If the UUT is an ASA Model:
   FSV = Maximum rated power / Current rating.
   If the UUT is an ASM Model:
   FSV = Full scale rated voltage.
5. Attach the voltmeter across the shunt.
6. Apply input power and power up the UUT.
7. 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. Send the command to calculate the calibration coefficients by the processor.

```
CAL:OUTP:CURR:CALC
```

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

```
CAL:UNLOCK “6867”
```

23. 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.

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

```
CAL:LOCK
```

12.6 ANALOG PROGRAM ADJUSTMENT

Analog programming ports can be calibrated with set of commands as described in below section. For detailed Pin description of the Analog programming connector and connection diagram, refer Section 9.2.

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. Locate the Analog Programming connector at the rear end of the power supply.
4. Connect a DC voltage source to PIN numbers required for the Calibration procedure that is to be performed or connect Resistor as a load for 1mA constant current source if the selected calibration procedure is on Resistance programming.

12.6.1 Voltage programming – 10V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 4 (VPRG_VSOUR / VPRG_4-20mA) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

4. Select the Analog programming source as Voltage source and 10V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:VOLT:PROG INT
SOUR:VOLT:PROG:FSC 10
```

5. Calibrate the Offset point.

```
CAL:OUTP:VOLT:EXTV:OFFS 0
```

6. Set the DC voltage at PIN4 (VPRG_VSOUR / VPRG_4-20mA to 10V +/- 0.003V. Calibrate the full scale point.

```
CAL:OUTP:VOLT:EXTV:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.2 Voltage programming – 5V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 4 (VPRG_VSOUR / VPRG_4-20mA) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

4. Select the Analog programming source as Voltage source and 5V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:VOLT:PROG INT
SOUR:VOLT:PROG:FSC 5
```

5. Calibrate the Offset point.

```
CAL:OUTP:VOLT:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 4 (VPRG_VSOUR / VPRG_4-20mA) to 5V +/- 0.003V. Calibrate the full scale point.

```
CAL:OUTP:VOLT:EXTV:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.3 Voltage programming – current source 10k ohms (Resistance Programming) calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero ohm across PIN 3 (VPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

4. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:VOLT:PROG INT
```

5. Calibrate the Offset point.

```
CAL:OUTP:VOLT:EXTI:OFFS 0
```

6. Connect 10k Ohm across PIN 3 (VPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector). Calibrate the full scale point.

```
CAL:OUTP:VOLT:EXTI:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.4 Voltage programming – current source 5k ohms (Resistance Programming) calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero ohm across PIN 3 (VPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector.

4. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:VOLT:PROG INT
```

5. Calibrate the Offset point.

```
CAL:OUTP:VOLT:EXTI:OFFS 0
```

6. Connect 5k Ohm across PIN 3 (VPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector). Calibrate the full scale point.

```
CAL:OUTP:VOLT:EXTI:FSC <UUT's Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK "6867"
CAL:STORE
CAL:LOCK
```

12.6.5 Voltage programming – 4-20mA source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

8. Connect 4mA source or 2VDC across PIN 4 (VPRG_VSOUR / VPRG_4-20mA) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

3. Select the Analog programming source as 4-20mA source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR MA420
SOUR:VOLT:PROG INT
```

4. Calibrate the Offset point.

```
CAL:OUTP:VOLT:EXT420MA:OFFS 0
```

5. Connect 20mA source or 10VDC across PIN 4 (VPRG_VSOUR / VPRG_4-20mA) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector. Calibrate the full scale point.

```
CAL:OUTP:VOLT:EXT420MA:FSC <UUT’s Full-scale volt>
```

6. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.6 Current programming – 10V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

4. Select the Analog programming source as Voltage source and 10V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:CURR:PROG INT
SOUR:CURR:PROG:FSC 10
```

5. Calibrate the Offset point.

```
CAL:OUTP:CURR:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) to 10V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:CURR:EXTV:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.7 Current programming – 5V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

8. Connect zero volt (0 +/- 0.003V) across PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

3. Select the Analog programming source as Voltage source and 5V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:CURR:PROG INT
SOUR:CURR:PROG:FSC 5
```

4. Calibrate the Offset point.

```
CAL:OUTP:CURR:EXTV:OFFS 0
```

5. Set the DC voltage at PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) to 5V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:CURR:EXTV:FSC <UUT’s Full-scale volt>
```

6. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.8 Current programming – 10k ohms current source (Resistance Programming) calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

9. Connect zero ohm across PIN 20 (IPRG_RES / IPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

3. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:CURR:PROG INT
```

4. Calibrate the Offset point.

```
CAL:OUTP:CURR:EXTI:OFFS 0
```

10. Connect 10k Ohm across PIN 20 (IPRG_RES / IPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector. Calibrate the full scale point.

```
CAL:OUTP:CURR:EXTI:FSC <UUT’s Full-scale volt>
```

5. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.9 Current programming – 5k ohms current source (Resistance Programming) calibration

STEP DESCRIPTION

6. Issue a *CLS command.

7. Issue a *RST command.

11. Connect zero ohm across PIN 20 (IPRG_RES / IPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

8. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:CURR:PROG INT
```

9. Calibrate the Offset point.

```
CAL:OUTP:CURR:EXTI:OFFS 0
```

12. Connect 5k Ohm across PIN 20 (IPRG_RES / IPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector.

10. . Calibrate the full scale point.

```
CAL:OUTP:CURR:EXTI:FSC <UUT’s Full-scale volt>
```

11. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
```

```
CAL:STORE
CAL:LOCK
```

12.6.10 Current programming – 4-20mA source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect 4mA source or 2VDC across PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

4. Select the Analog programming source as 4-20mA source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR MA420
SOUR:CURR:PROG INT
```

5. Calibrate the Offset point.

```
CAL:OUTP:CURR:EXT420MA:OFFS 0
```

6. Connect 20mA source or 10VDC across PIN 19 (IPRG_VSOUR / IPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming Connector. Calibrate the full scale point.

```
CAL:OUTP:CURR:EXT420MA:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.11 Power programming – 10V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

4. Select the Analog programming source as Voltage source and 10V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:POW:PROG INT
SOUR:POW:PROG:FSC 10
```

5. Calibrate the Offset point.

```
CAL:OUTP:POW:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) to 10V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:POW:EXTV:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.12 POWER programming – 5V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

4. Select the Analog programming source as Voltage source and 5V range. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR VOLT
SOUR:POW:PROG INT
SOUR:POW:PROG:FSC 5
```

5. Calibrate the Offset point.

```
CAL:OUTP:POW:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) to 5V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:POW:EXTV:FSC <UUT's Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK "6867"
CAL:STORE
CAL:LOCK
```

12.6.13 POWER programming – current source 10k ohms (Resistance Programming) calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero ohm across PIN 34 (PPROG_RES / PPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

4. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:POW:PROG INT
```

5. Calibrate the Offset point.

```
CAL:OUTP:POW:EXTI:OFFS 0
```

6. Connect 10k Ohm across PIN 34 (PPROG_RES / PPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming Connector. Calibrate the full scale point.

```
CAL:OUTP:POW:EXTI:FSC <UUT's Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK "6867"
CAL:STORE
CAL:LOCK
```

12.6.14 POWER programming – current source 5k ohms (Resistance Programming) calibration

NOTE: This feature is not available for Asterion DC Half Rack

STEP DESCRIPTION

8. Issue a *CLS command.

9. Issue a *RST command.

10. Connect zero ohm across PIN 34 (PPROG_RES / PPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

11. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR CURR
SOUR:POW:PROG INT
```

12. Calibrate the Offset point.

```
CAL:OUTP:POW:EXTI:OFFS 0
```

13. Connect 5k Ohm across PIN 34 (PPROG_RES / PPRG_ISOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming Connector. Calibrate the full scale point.

```
CAL:OUTP:POW:EXTI:FSC <UUT’s Full-scale volt>
```

14. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.6.15 POWER programming – 4-20mA source calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect 4mA source or 2VDC across PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector.

4. Select the Analog programming source as 4-20mA source. Also select programming mode as internal during calibration procedure.

```
SOUR:ANALOG:PROG:SOUR MA420
SOUR:POW:PROG INT
```

5. Calibrate the Offset point.

```
CAL:OUTP:POW:EXT420MA:OFFS 0
```

6. Connect 20mA source or 10VDC across PIN 33 (PPRG_VSOUR / PPRG_4-20mA_SOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming Connector. Calibrate the full-scale point.

```
CAL:OUTP:POW:EXT420MA:FSC <UUT's Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK "6867"
CAL:STORE
CAL:LOCK
```

12.6.16 OVER-Voltage programming – 10V range calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. Connect zero volt (0 +/- 0.003V) across PIN 5 (OVPPRG_VSOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).
4. Select the over-voltage Analog programming range as 10V. Also select programming mode as internal during calibration procedure.

```
SOUR:VOLT:PROT:PROG INT
SOUR:VOLT:PROT:PROG:FSC 10
SOUR:ANALOG:PROG:SOUR VOLT
```

5. Calibrate the Offset point.

```
CAL:OUTP:OVERVOLT:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 5 (OVPPRG_VSOUR) to 10V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:OVERVOLT:EXTV:FSC <UUT's Full-scale volt>
```

7. Unlock and store the Calibration.

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

```
CAL:STORE
CAL:LOCK
```

12.6.17 OVER-Voltage programming – 5V range calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect zero volt (0 +/- 0.003V) across PIN 5 (OVPPRG_VSOUR) and PIN 18 (EXT_PRG_RTN) of Analog Programming connector (44-pin connector).

4. Select the over-voltage Analog programming range as 5V. Also select programming mode as internal during calibration procedure.

```
SOUR:VOLT:PROT:PROG INT
SOUR:VOLT:PROT:PROG:FSC 5
```

5. Calibrate the Offset point.

```
CAL:OUTP:OVERVOLT:EXTV:OFFS 0
```

6. Set the DC voltage at PIN 5 (OVPPRG_VSOUR) to 5V +/-0.003V. Calibrate the full scale point.

```
CAL:OUTP:OVERVOLT:EXTV:FSC <UUT’s Full-scale volt>
```

7. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

12.7 REMOTE MONITOR OUTPUTS CALIBRATION

Remote Monitor output signals can be calibrated with set of commands as described in below section. For detailed Pin description of the Remote Monitor Outputs and connection diagram, Refer Section 9.4

12.7.1 Voltage Monitor Output Calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

8. Connect the DMM at PIN 6 (VMON) and PIN 22 (MON_RTN) of Analog Programming connector (44-pin connector) to measure voltage monitor pin

voltage. Set the load for approximately 25% of rated output current. Type the following commands.

```
SOUR:VOLT 0
SOUR:CURR <UUT's Full-scale current>
OUTP:STAT 1
CAL:INIT:VOLT:MON
```

3. Calibrate the Offset point.

```
CAL:OUTP:VOLT:MON:OFFS <voltage at PIN6 read from DMM>
```

4. Set the output voltage to 80% of UUT's Full-scale rated voltage.

```
SOUR:VOLT <0.8 times UUT's Full-scale voltage>
CAL:OUTP:VOLT:MON:FSC <voltage at PIN6 read from DMM>
```

5. Unlock and store the Calibration.

```
CAL:UNLOCK "6867"
CAL:STORE
CAL:LOCK
```

12.7.2 Current Monitor Output Calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect the DMM at PIN 21 (IMON) and PIN 22 (MON_RTN) of Analog Programming connector (44-pin connector) to measure current monitor pin voltage. Connect a short across the output. Type the following commands.

```
SOUR:VOLT <UUT's Full-scale voltage>
SOUR:CURR 0
OUTP:STAT 1
CAL:INIT:CURR:MON
```

4. Calibrate the Offset point.

```
CAL:OUTP:CURR:MON:OFFS <voltage at PIN21 read from DMM>
```

5. Set the output current to 80% of UUT's Full-scale rated current.

```
SOUR:CURR <0.8 times UUT's Full-scale current>
CAL:OUTP:CURR:MON:FSC <voltage at PIN21 read from DMM>
```

6. Unlock and store the Calibration.

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

```
CAL:STORE
CAL:LOCK
```

12.7.3 Power Monitor Output Calibration

STEP DESCRIPTION

1. Issue a *CLS command.

2. Issue a *RST command.

3. Connect the DMM at PIN 35 (PMON) and PIN 22 (MON_RTN) to measure voltage monitor pin voltage. Set the output load to 90% of the full scale. Type the following commands.

```
SOUR:VOLT <UUT’s Full-scale voltage>
SOUR:CURR <UUT’s Full-scale current>
SOUR:POW 0
OUTP:STAT 1
CAL:INIT:POW:MON
```

4. Calibrate the Offset point.

```
CAL:OUTP:POW:MON:OFFS <voltage at PIN35 read from DMM>
```

5. Set the output power to 80% of UUT’s Full-scale rated power.

```
SOUR:POW <0.8 times UUT’s Full-scale power>
CAL:OUTP:POW:MON:FSC <voltage at PIN35 read from DMM>
```

6. Unlock and store the Calibration.

```
CAL:UNLOCK “6867”
CAL:STORE
CAL:LOCK
```

13
SCPI STATUS IMPLEMENTATION

13.1 SCPI STATUS BYTE IMPLEMENTATION

SCPI status byte implementation for the Asterion DC power supply is described in ***Figure 13-1***.

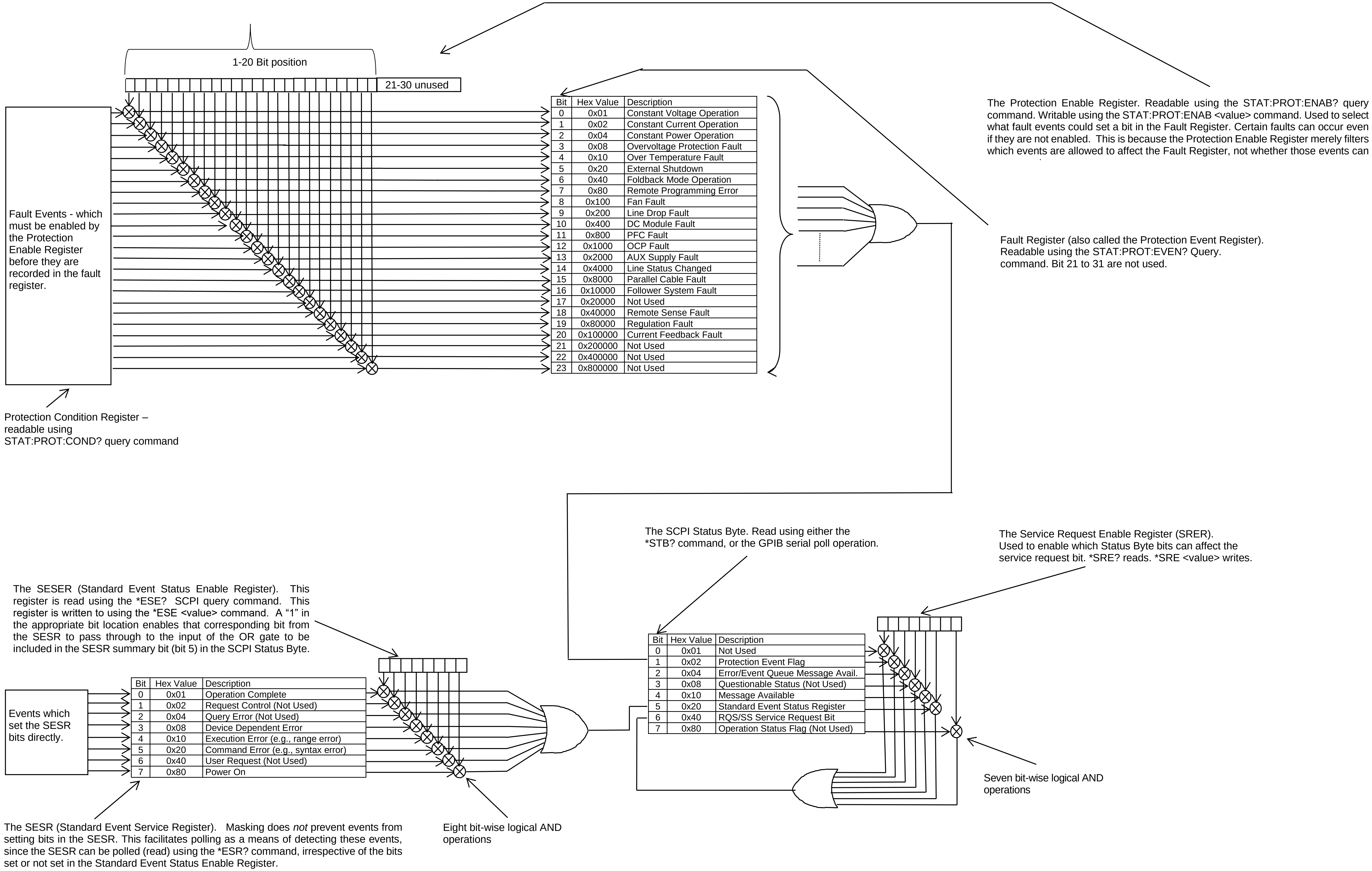


Figure 13-1. SCPI Status Implementation Diagram

INDEX

C

CALibrate, 9-27, 9-28
Calibration
  Setup for Calibration, 10-39
  Voltage Programming Calibration, 10-39
  Warning, 10-38
Calibration Procedrues
  Introduction, *10-38*
  Voltage Programming, Ethernet, 10-39, 10-42
Calibration Procedures, 10-38
  Analog Program Adjustment, *10-46*
  Current Programming, Ethernet, 10-44
  Setup, *10-39*
CALIBRATION SCPI Command Reference, 9-29
Configuration
  Ethernet/Lan, 6-6
  RS232 Setup Procedure, 4-3
  Setup Procedure, Ethernet, 6-8
Configuration and Remote Programming
  Ethernet, 6-6, 7-20
Configurations
  IEEE 488.2 GPIB Setup Procedures, 7-21
  Setup Procedure, DHCP, 6-8
Conventions
  [ ], 9-9

E

Error/Event Queue, 9-7
  SCPI Error Codes, 9-7
Ethernet Interface, 1-2
Examples, Using SCPI Commands, 9-34
  OVP Setup, *9-35*
  Power On INIT, *9-37*
  Ramp I, *9-36*
  Ramp V, 2, *9-37*
  Trigger, *9-36*
  VI Mode, *9-34*
External User Control Signal Connector
  Warning, 8-20

F

Features
  IEEE 488.2 GPIB, 2-2
Features of Ethernet interface, 2-2
Features of RS232 interface, 2-1
Features, Functions and Specifications, 2-1
Functions, **2-3**
  Programmable, **2-2**
  Readback, 2-3

G

Global Channel 0, 7-21

H

HTRIGGER SCPI Command Reference, *9-26*

I

IEEE 488.2 GPIB Configurations and Remote Programming, 5-1
IEEE 488.2 GPIB Interface, 1-2
IEEE-488.2 Common Command Subsystem, 9-10
  *CLS, 9-10

*ESE, 9-10
*ESR?, 9-10
*IDN?, 9-10
*OPC, 9-10
*RST, 9-10
*SRE, 9-10
*STB?, 9-11
Interactive Control, 6-17

L

LXI Identification, 6-18

M

MEASURE SCPI Command Reference, 9-20
MEASURE SCPI Command Subsystem, 9-20

N

Network Setup Procedure
DHCP, 6-8

O

Operation
Remote Output On/Off Control, 8-24
Remote Overvoltage Protection Programming, 8-17
Remote Voltage Programming, 8-13, 8-15
OUTPUT SCPI Command Reference, 9-22
OUTPUT SCPI Command Subsystem, 9-21
Overview, 1-1

P

Parameter Definitions, 9-9
boolean, 9-9
NR1, 9-9
NRf, 9-9
string, 9-9
Programming Accuracy
IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Programming Resolution
IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Programming/Communication via Ethernet, 6-12
Raw Socket Interface, 6-12
Web Server, 6-13

Q

Queries, 9-9

R

Ramp Funciton, *9-19*
Ramping
Description, 9-19, 9-26
Readback Accuracy
IEEE 488.2 GPIB/RS232/Ethernet, 2-4
Readback Resolution
IEEE 488.2 GPIB/RS232/Ethernet, 2-4
Rear Panel, 4-1
Ethernet Layout, 6-7
Rear Panel Configuration Switch
Address Selection, GPIB, 5-2, 7-21
Power-On GPIB Service Request (PON SRQ) Selection, 5-3, 7-21
Remote Current Programming
Resistance Control, 8-10
Voltage Source Control, 8-13, 8-16, 8-17
Remote Output On/Off Control, 8-24
Contact Closure Control, 8-24
External Source Control, 8-25
Remote Overvoltage Protection Programming, 8-17
Remote Programming via RS-232, 4-1
Remote Voltage Programming, 8-13, 8-15
Resistance Control, 8-11, 8-12
Voltage Source Control, 8-13, 8-14
Remote/Local Selection, 3-1
RS232 Configurations and Remote Programming, 4-1
RS232 Interface, 1-1, 1-2
RS232/Ethernet/IEEE 488.2 GPIB and SCPI Command Operation, 9-1

S

SCPI Command Operation
CALibrate, 9-27, 9-28
STATus, 9-34
SCPI Commands
Application Examples, *9-34*
CALIBRATION, 9-27
HTRIGGER, *9-26*
Measure, *9-20*
Output, *9-21*
Source, *9-11*
Status, *9-34*
System, *9-23*
SCPI Conformance, 9-8
SCPI Conventions, 9-9
SCPI Parameter Definitions, *9-9*
SCPI STATUS IMPLEMENTATION, 11-57
SCPI Status Byte, 9-5
Serial Poll Operation, 9-8
Setup Procedure, 6-8
SGX Banner and Tab, 6-13
Shield Ground, GPIB, 7-21
Source SCPI Command Reference, *9-14*

Specifications
IEEE 488.2 GPIB/RS232/Ethernet, 2-3
Specifications, IEEE 288.2 GPIB, RS232 and Ethernet, 2-3
Standard Event Status Register (ESR), 9-4
STATus, 9-34
STATUS SCPI Command Reference, 9-34
STATUS SCPI Command Subsystem, 9-34
System Requirements, Ethernet Minimum, 2-4
SYSTEM SCPI Command Reference, 9-24
SYSTEM SCPI Command Subsystem, 9-23

T

TRIGGER SCPI Command Reference, *9-26*
TRIGGER SCPI Command Subsystem, 9-26

W

Web Server
Configuration page, 6-15
Home page, 6-14
Login window, 6-13