

Sorensen
Asterion DC
Multiooutput 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 \$5 billion. The Company has over 15,000 colleagues working at more than 150 manufacturing facilities and more than 100 sales and service centers in the United States and around the world.

Trademarks

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

Notice of Copyright

M330517-01, Asterion DC Multioutput Series Programming Manual © 2021, AMETEK Programmable Power, Inc. All rights reserved.

Exclusion for Documentation

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

- (a) MAKES NO WARRANTY AS TO THE ACCURACY, SUFFICIENCY OR SUITABILITY OF ANY TECHNICAL OR OTHER INFORMATION PROVIDED IN ITS MANUALS OR OTHER DOCUMENTATION.
- (b) ASSUMES NO RESPONSIBILITY OR LIABILITY FOR LOSSES, DAMAGES, COSTS OR EXPENSES, WHETHER SPECIAL, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL, WHICH MIGHT ARISE OUT OF THE USE OF SUCH INFORMATION. THE USE OF ANY SUCH INFORMATION WILL BE ENTIRELY AT THE USER'S RISK, AND
- (c) REMINDS YOU THAT IF THIS MANUAL IS IN ANY LANGUAGE OTHER THAN ENGLISH, ALTHOUGH STEPS HAVE BEEN TAKEN TO MAINTAIN THE ACCURACY OF THE TRANSLATION, THE ACCURACY CANNOT BE GUARANTEED. APPROVED AMETEK CONTENT IS CONTAINED WITH THE ENGLISH LANGUAGE VERSION, WHICH IS POSTED AT WWW.PROGRAMMABLEPOWER.COM.

Date and Revision

30 Dec 21, Rev D

Part Number

M330517-01



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.

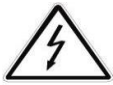
In normal operation, the operator does not have access to hazardous voltages within the chassis. However, depending on the user's application configuration, **HIGH VOLTAGES HAZARDOUS TO HUMAN SAFETY** may be normally generated on the output terminals. The customer/user must ensure that the output power 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.

This page intentionally left blank.

SAFETY SYMBOLS



WARNING
Risk of Electrical Shock



CAUTION
Refer to Accompanying Documents



Off (Supply)



Direct Current (DC)



Standby (Supply)



Alternating Current (AC)



On (Supply)



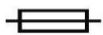
Three-Phase Alternating Current



Protective Conductor Terminal



Earth (Ground) Terminal



Fuse



Chassis Ground

This page intentionally left blank.

Product Family: Asterion DC Multioutput 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.powerandtest.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.....	1-1
1.4	Ethernet Interface.....	1-2
1.5	IEEE 488.2 GPIB Interface.....	1-2
2	FEATURES, FUNCTIONS AND SPECIFICATIONS	
	2-4	
2.1	Introduction	2-4
2.2	Features and Programmable Functions	2-4
2.3	Specific Features in Ethernet interface	2-5
2.4	Readback Functions	2-5
2.5	Specifications.....	2-5
2.5.1	Programming Resolution	2-5
2.5.2	Programming Accuracy	2-5
2.5.3	Readback Resolution	2-6
2.5.4	Readback Accuracy.....	2-6
2.6	Minimum System Requirements	2-6
2.6.1	PC Connection.....	2-6
2.6.2	Internet or LAN Connection	2-6
3	LOCAL/REMOTE SELECTION	3-8
3.1	Introduction	3-8
3.2	Remote mode	3-8
4	RS232 CONFIGURATIONS AND REMOTE	
	PROGRAMMING	4-10
4.1	Rear Panel.....	4-10
4.2	Remote Programming Via RS232.....	4-11
4.2.1	RS232 Connector Pinout.....	4-11
4.2.2	RS232 Setup Procedure.....	4-11

5 USB CONFIGURATIONS AND REMOTE PROGRAMMING 5-17

5.1 Rear Panel..... 5-17

5.2 USB Setup Procedure 5-18

5.2.1 Port Number Identification 5-18

5.2.2 Communication test using serial interface program 5-19

6 ETHERNET CONFIGURATION AND REMOTE PROGRAMMING 6-22

6.1 Introduction 6-22

6.2 Rear Panel..... 6-22

6.3 Ethernet/LAN Configuration 6-23

6.3.1 Ethernet Configuration Factory Defaults 6-23

6.4 Ethernet Setup Procedure 6-23

6.4.1 Network Setup Using DHCP Server 6-24

6.4.2 Network Setup Using Auto-IP (Direct Connection between Asterion power Supply and PC using Cross cable)..... 6-26

6.4.3 Network Setup using Static IP 6-26

6.4.4 Raw Socket Interface 6-28

6.4.5 VXI-11 Protocol..... 6-28

6.4.6 Web Server..... 6-29

6.5 Ethernet Web Pages, Overview 6-29

6.5.1 Home 6-30

6.5.2 IP Configuration 6-31

6.5.3 Interactive Control..... 6-33

6.5.4 LXI Identification 6-34

7 IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING 7-36

7.1 Rear Panel..... 7-36

7.2 Remote Programming Via IEEE 488.2 GPIB connector 7-37

7.2.1 Address Selection..... 7-37

7.2.2 Power-On GPIB Service Request (PON SRQ) Selection 7-37

7.2.3 Shield Ground..... 7-37

7.2.4 IEEE 488.2 GPIB Setup Procedure 7-37

8	REMOTE ANALOG PROGRAMMING INTERFACES.....	8-39
8.1	Introduction	8-39
8.2	Remote Analog Programming Connector details	8-39
8.3	Remote ANALOG Programming.....	8-41
8.3.1	REMOTE PROGRAMMING BY RESISTANCE	8-41
8.3.2	Remote Programming BY VOLTAGE	8-43
8.4	Remote MONITOR OUTPUTS	8-45
9	REMOTE EXTERNAL USER CONTROL INTERFACE.....	9-47
9.1	Introduction	9-47
9.2	Remote External User Control Interface Connector details	9-47
9.2.1	Fault common to all channels	9-52
9.2.2	Table 9-2: List of Faults in Asterion DC Multioutput Series Power Supply Trigger-In & Trigger Out	9-53
9.2.3	Remote Output On/Off Control	9-54
10	RS232/ETHERNET/ IEEE 488.2 GPIB REGISTER COMMAND OPERATION	10-57
10.1	Introduction	10-57
10.2	Register Definitions	10-57
10.2.1	Protection Condition and Protection Event Status Register	10-57
10.2.2	Standard Event Status Register (ESR)	10-61
10.2.3	SCPI Status Byte	10-62
10.2.4	Error/Event Queue.....	10-63
10.2.5	Serial Poll Operation.....	10-64
10.2.6	Self-Test	10-64
10.2.1	Fault specific to channels	10-66
10.3	SCPI Conformance Information	10-69
10.3.1	Parameter Definitions	10-69
10.3.2	Conventions.....	10-70
10.3.3	Queries	10-70
10.4	IEEE 488.2 Common Command Subsystem.....	10-70
10.5	SOURCE SCPI Command Subsystem	10-73
10.5.1	SOURCE SCPI Command Summary.....	10-73
10.5.2	SOURCE SCPI Command Reference	10-75
10.5.3	RAMP FUNCTION.....	10-79

10.6	MEASURE SCPI Command Subsystem	10-80
10.6.1	MEASURE SCPI Command Summary	10-80
10.6.2	MEASURE SCPI Command Reference	10-80
10.7	OUTPUT SCPI Command Subsystem	10-81
10.7.1	OUTPUT SCPI Command Summary	10-81
10.7.2	OUTPUT SCPI Command Reference	10-81
10.8	SYSTEM SCPI Command Subsystem	10-83
10.8.1	SYSTEM SCPI Command Summary	10-83
10.8.2	SYSTEM SCPI Command Reference	10-84
10.9	HTRIGGER SCPI Command Subsystem	10-87
10.9.1	HTRIGGER SCPI Command Summary	10-87
10.9.2	HTRIGGER SCPI Command Reference	10-87
10.10	TRIGGER SCPI Command Subsystem	10-87
10.10.1	TRIGGER SCPI Command Summary	10-87
10.10.2	TRIGGER SCPI Command Reference	10-87
10.11	CALIBRATION SCPI Command Subsystem	10-89
10.11.1	CALIBRATION SCPI Command Summary	10-89
10.11.2	CALIBRATION SCPI Command Reference	10-92
10.12	STATUS SCPI Command Subsystem	10-97
10.12.1	STATUS SCPI Command Summary	10-97
10.12.2	STATUS SCPI Command Reference	10-97
10.13	Examples of Using the SCPI Commands	10-98
10.13.1	VI Mode Example	10-98
10.13.2	OVP Setup Example	10-99
10.13.3	Trigger Example	10-100
10.13.4	Ramp I Example	10-100
10.13.5	Ramp V Example 2	10-101
10.13.6	Power On INIT Example	10-101
11	OUTPUT ON/OFF DELAY	11-104
11.1	Introduction	11-104
11.2	Applications under OUTPUT ON/OFF DELAY	11-104
12	SEQUENCING FUNCTION	12-108
12.1	Introduction	12-108
12.2	Program SCPI Command Subsystem	12-108
12.2.1	Program SCPI Command Summary	12-109
12.2.2	Program SCPI Command Reference	12-109
12.2.3	Sequence Creation and Execution Examples	12-113

13	CALIBRATION.....	13-126
13.1	Introduction	13-126
13.2	Setup for Calibration.....	13-127
13.3	Voltage Programming Calibration.....	13-128
13.4	Voltage Programming Calibration – REMOTE SENSE.....	13-130
13.5	Current Programming Calibration.....	13-134
13.6	Analog Program Adjustment	13-136
13.6.1	Voltage programming – 10V range voltage source calibration	13-136
13.6.2	Voltage programming – 5V range voltage source calibration	13-137
13.6.3	Voltage programming – current source 10k ohms range (Resistance Programming) calibration	13-139
13.6.4	Voltage programming – current source 5k ohms range (Resistance Programming) calibration	13-140
13.6.5	Current programming – 10V range voltage source calibration.....	13-141
13.6.6	Current programming – 5V range voltage source calibration.....	13-142
13.6.7	Current programming – 10k ohms range current source (Resistance Programming) calibration	13-143
13.6.8	Current programming – 5k ohms range current source (Resistance Programming) calibration	13-144
13.7	REMOTE MONITOR OUTPUTS CALIBRATION.....	13-145
13.7.1	Voltage Monitor Output Calibration	13-145
13.7.2	Current Monitor Output Calibration.....	13-146
14	SCPI STATUS IMPLEMENTATION	14-148
14.1	SCPI Status Byte Implementation	14-148

TABLE

Table 4-1: RS-232C Interface Connector Type 4-11

Table 4-2: RS-232C Interface Connector Pinout 4-11

Table 4-3: Configuration requirements for RS232 4-12

Table 4-4: RS-232C Interface Connector Pinout 4-12

Table 8-1: Remote Analog Programming Connector pin out details 8-41

Table 9-1: Remote External User Interface connector pin out details 9-51

9.2.2 Table 9-2: List of Faults in Asterion DC Multioutput Series Power Supply
Trigger-In & Trigger Out..... 9-53

Table 10-1: Protection Condition and Event Status Registers 10-59

Table 10-2: Standard Event Status Register 10-61

Table 10-3: SCPI Status Byte 10-62

Table 10-4: SCPI Error Codes 10-64

Table 10-5: Self-Test Error Codes 10-65

Table 10-6: Faults 10-66

Table 10-7: Faults Specific to Channels..... 10-68

FIGURE

Figure 3-1: Local/Remote Screen 3-8

Figure 3-2: Home Screen..... 3-8

Figure 4-1: Rear Panel – RS232 Interface 4-10

Figure 4-2: RS-232C Interface Connector 4-11

Figure 4-3: RS232 Communications Cable Pinout 4-12

Figure 4-4: RS232 Screen (Configure)..... 4-13

Figure 4-5: Port Setting in Tera Term..... 4-13

Figure 4-6: Baud rate Setting in Tera Term 4-14

Figure 4-7: Terminal option in Tera Term..... 4-14

Figure 4-8: Terminal Setup in Tera Term 4-15

Figure 5-1: Rear Panel – Type-B USB Interface 5-17

Figure 5-2: Port identification 5-18

Figure 5-3: Port identification in Tera Term 5-19

Figure 5-4: Baud rate selection in Tera Term 5-19

Figure 5-5: Terminal selection in Tera Term 5-20

Figure 5-6: Terminal setup in Tera Term..... 5-20

Figure 6-1: Rear Panel - Ethernet Interface 6-22

Figure 6-2: LAN Screen (Configure)..... 6-24

Figure 6-3: LAN Screen 6-24

Figure 6-4: LAN Screen (Settings) 6-25

Figure 6-5: RS232 Screen (Settings) 6-25

Figure 6-6: LAN configure (Settings)..... 6-27

Figure 6-7: LAN Screen (Configure Static IP)..... 6-27

Figure 6-8: LAN Screen (Subnet Mask) 6-27

Figure 6-9: LAN Screen (Gateway Address) 6-27

Figure 6-10: Asterion DC Multioutput Series power supply Banner and Tab 6-29

Figure 6-11: Login Window 6-29

Figure 6-12: Asterion DC Multioutput Series power supply Home Page..... 6-30

Figure 6-13: Asterion DC Multioutput Series power supply Device identify screen..... 6-31

Figure 6-14: IP Configuration Page..... 6-31

Figure 6-15: Interactive Control Page 6-33

Figure 6-16: LXI Identification Page 6-34

Figure 7-1: Rear Panel – IEEE 488.2 GPIB Interface..... 7-36

Figure 7-2: GPIB Screen..... 7-37

Figure 8-1: Rear Panel - Analog Programming Connector 8-39

Figure 8-2: Remote Analog Programming Interface Connector Pin-Out..... 8-40

Figure 8-3: Remote Current Programming Using Resistance..... 8-42

Figure 8-4: Remote Power Programming Using Resistance 8-43

Figure 8-5: Remote Current Programming Using 0-5 VDC or 0-10 VDC Source 8-44

Figure 8-6: Remote Voltage Programming Using 0-10 VDC Source 8-45

Figure 9-1: Rear Panel – External User Control Interface 9-47

Figure 9-2: External User Control Interface Connector Pin-Out..... 9-48

Figure 9-3: Remote Output ON/OFF by contact closure channel 1 9-54

Figure 9-4: Remote Output ON/OFF by contact closure channel 2 9-55

Figure 9-5: Remote Output ON/OFF by contact closure channel 3 9-55

Figure 10-1: Protection Condition and Protection Event Register..... 10-60

Figure 10-2: Standard Event Status and Standard Event Status Enable Register 10-61

Figure 10-3: SCPI Status Byte and Service Request Enable Register 10-63

Figure 12-1: Power Supply Output for Example 1 12-115

Figure 12-2: Power Supply Output for Example 2 12-116

Figure 12-3: Power Supply Output for Example 3 12-117

Figure 12-4: End-of-Sequence Pause for Example 4 12-118

Figure 12-5: Power Supply Output Example 5 12-122

Figure 12-6: Power Supply Output Example 6 12-123

Figure 12-7: Power Supply Output for Example 7 12-124

Figure 14-1. SCPI Status Implementation Diagram..... 14-149

This page intentionally left blank.

1 OVERVIEW

1.1 Introduction

This manual provides instructions for remote programming control and monitoring for Asterion DC Multioutput Series power supplies. For easy navigation to the applicable instructions, this manual separates RS232, USB, IEEE 488.2 GPIB and Ethernet setup instructions. The instructions then converge where they are common to all three interface options. The Asterion DC Multioutput Series power supplies provides default interface of RS232, USB and Ethernet. IEEE 488.2 GPIB is an optional interface. Use this programming manual in conjunction with your Asterion DC Multioutput Series power supplies operational manual.

1.2 RS232 Interface

If you are using the RS232 interface, go to:

- 2 for Features, Functions and Specifications
- 3 for Local/Remote Selection
- 4 for RS232 Configurations and Remote Programming
- 8 for Remote Analog Programming
- 9 External User Control Interface Connectors
- 10 for SCPI Commands and Definitions
- 11 Output ON/OFF Delay
- 12 Sequence Function
- 13 for Calibration Procedures
- 14 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 Local/Remote Selection
- 5 for USB Configurations and Remote Programming

- 8 for Remote Analog Programming
- 9 External User Control Interface Connectors
- 10 for SCPI Commands and Definitions
- 11 Output ON/OFF Delay
- 12 Sequence Function
- 13 for Calibration Procedures
- 14 for SCPI Status Implementation

1.4 Ethernet Interface

If you are using the Ethernet interface, go to:

- 2 for Features, Functions and Specifications
- 3 for Local/Remote Selection
- 6 for Ethernet Configuration and Remote Programming
- 8 for Remote Analog Programming
- 9 External User Control Interface Connectors
- 10 for SCPI Commands and Definitions
- 11 Output ON/OFF Delay
- 12 Sequence Function
- 13 for Calibration Procedures
- 14 for SCPI Status Implementation

1.5 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 Local/Remote Selection
- 7 for IEEE 488.2 GPIB Configuration and Remote Programming
- 8 for Remote Analog Programming
- 9 External User Control Interface Connectors
- 10 for SCPI Commands and Definitions
- 11 Output ON/OFF Delay
- 12 Sequence Function
- 13 for Calibration Procedures
- 14 for SCPI Status Implementation

This page intentionally left blank.

2

FEATURES, FUNCTIONS AND SPECIFICATIONS

2.1 Introduction

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

2.2 Features and Programmable Functions

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

- Output voltage and current
- User limits for voltage and current
- Overvoltage protection
- Programming and readback of voltage and current
- Programmable overvoltage protection with reset
- Output Modes of Operation: Constant-Voltage (CV), Constant-Current (CC), Constant Power (CP) or Foldback mode (CV/CC/CP), with reset.
- Output enable/disable
- Maskable fault interrupt
- Hold and trigger
- Full calibration
- SCPI compliant command set
- Soft calibration
- Remote Inhibit
- Voltage Ramp and Current Ramp functions
- Sequencing Function

- Programmable ON/OFF delay between the channels for enabling the Output.
- Rear panel Remote Analog Programming and External user control interface (Common to all interfaces)
- Front panel control selection (Local/Remote)

2.3 Specific Features in Ethernet interface

- Ethernet/LAN connectivity, 10/100base-T compatible
- Built-in Web Server for direct control using Web Browser
- Field-upgradable firmware via Ethernet

2.4 Readback Functions

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

- Actual measured voltage and current
- Voltage and current settings
- 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

2.5 Specifications

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

2.5.1 Programming Resolution

Voltage Output:	0.012% of full scale
Current Output:	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

2.5.2 Programming Accuracy

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

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

2.5.3 Readback Resolution

Voltage Output:	$\pm 0.012\%$ of full scale
Current Output:	$\pm 0.012\%$ of full scale
Power Output:	$\pm 0.012\%$ of full scale

2.5.4 Readback Accuracy

Voltage Output:	$\pm (0.1\%$ of full-scale output voltage)
Current Output:	$\pm (0.2\%$ of full-scale output current) *
Power Output:	$\pm (0.3\%$ of rated output power)

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

2.6 Minimum System Requirements

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

2.6.1 PC Connection

To operate your Asterion DC Multioutput Series 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 10 (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

2.6.2 Internet or LAN Connection

To operate your Asterion DC Multioutput Series power supply connected to the Internet or a LAN you will need:

- Pentium-based laptop or desktop computer running Microsoft Windows 10 (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

This page intentionally left blank.

LOCAL/REMOTE SELECTION

3.1 Introduction

This section describes the operation of Local or Remote modes available in the front panel. Remote mode allows the power supply operation to be controlled only through digital communication interfaces. Whereas Local mode allows the power supply operations controlled through front panel.

3.2 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).

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

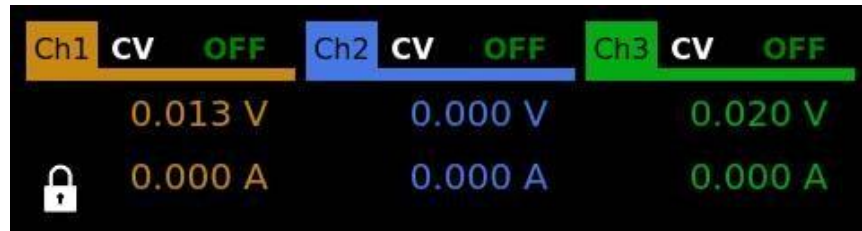


Figure 3-1: Local/Remote Screen

- Upon pressing the Lock symbol or by sending the command **SYST:LOCAL 1**, Asterion DC Multioutput Series power supply front panel screen will navigate to Home Screen menu. (Refer to Figure 3-2)



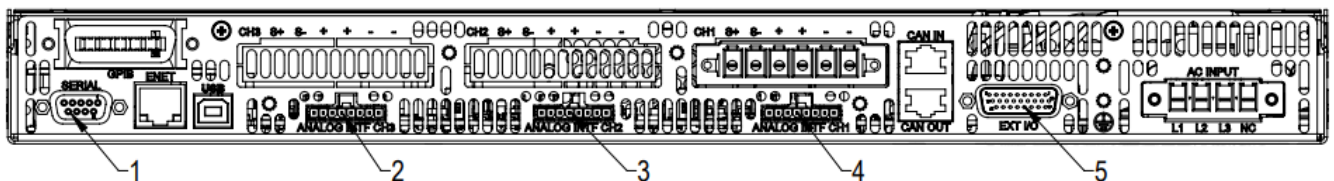
Figure 3-2: Home Screen

This page intentionally left blank.

RS232 CONFIGURATIONS AND REMOTE PROGRAMMING

4.1 Rear Panel

This section provides illustration of the Asterion DC Multioutput Series power supply's rear panel layout. Refer to Figure 4-1, regardless of the layout, the component functions are common across all models and those that are pertinent to the RS232 options are described here.



- 1- RS232C 9-contact receptacle (female) Subminiature-D
- 2- Remote Analog Programming Connector Channel 3
- 3- Remote Analog Programming Connector Channel 2
- 4- Remote Analog Programming Connector Channel 1
- 5- 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)

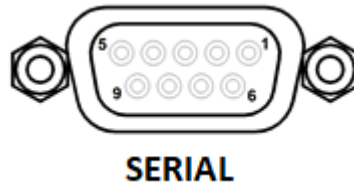


Figure 4-2: RS-232C Interface Connector

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	<ul style="list-style-type: none"> 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) 	Not Selectable
Outgoing Termination Character(s)	CR LF (Carriage Return and Line Feed): HEX, 0x0d 0x0a (DEC, 13 10)	Not Selectable

Table 4-3: Configuration requirements for RS232

1. Build an RS232 communications cable as per the pinout description illustrated in Figure 4-3 and Table 4-4 (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 4-4: RS-232C Interface Connector Pinout

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

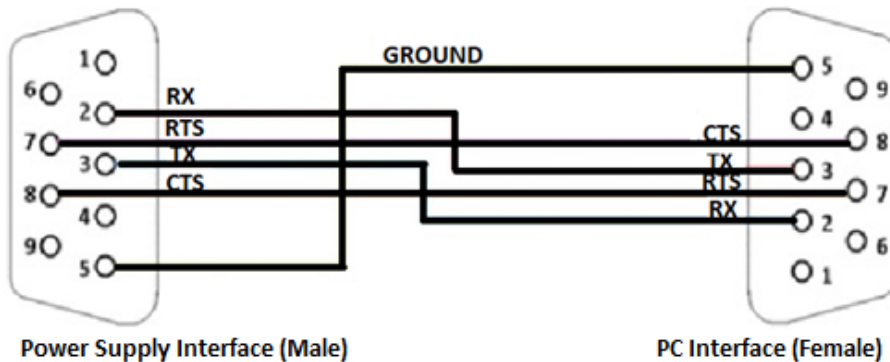
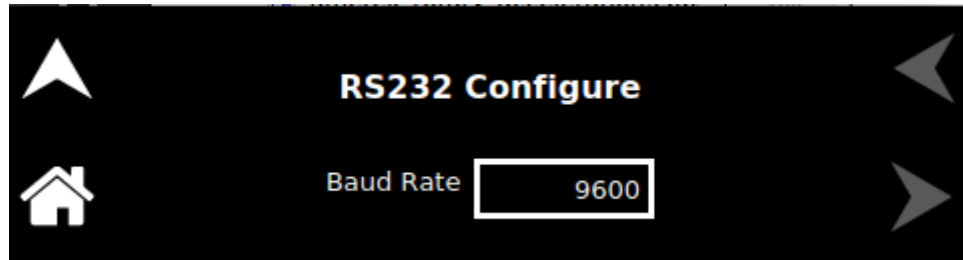


Figure 4-3: RS232 Communications Cable Pinout

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.

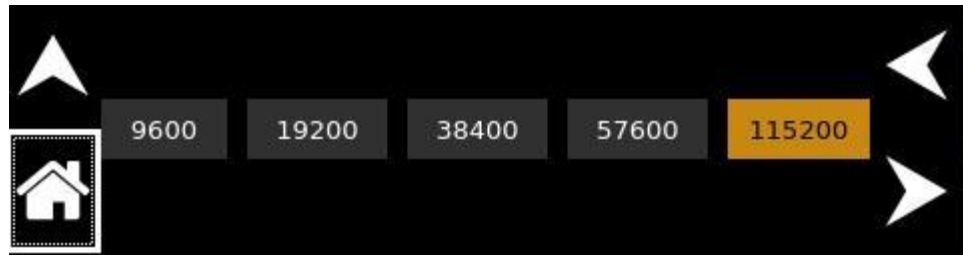


Figure 4-4: RS232 Screen (Configure)

5. Use one of the available programs for serial communication, such as Tera Term.
6. If you choose to use Tera Term:
 - Select serial option in the Tera term window and select the port as COM1 and press OK. (Refer to Figure 4-5)

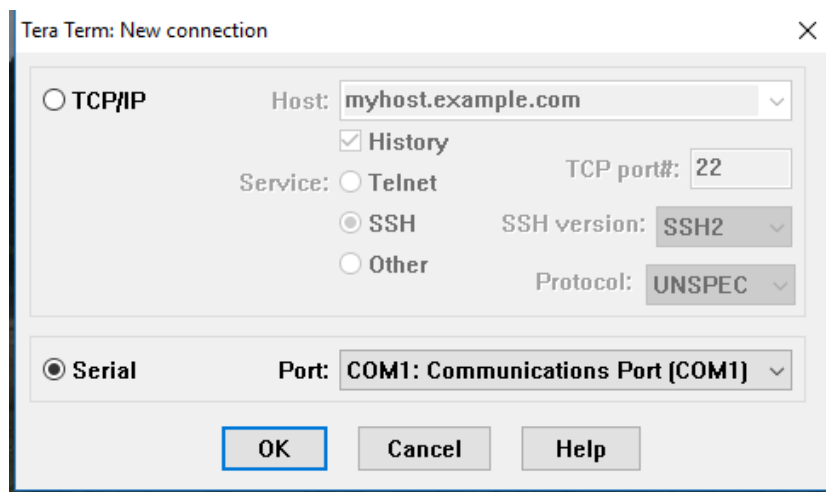


Figure 4-5: Port Setting in Tera Term

- Select setup and select serial port in the setup window.
- Set the baud rate to 9600, 8 data bits, no parity, 1 stop bit, and no flow control. (Refer to Figure 4-6)

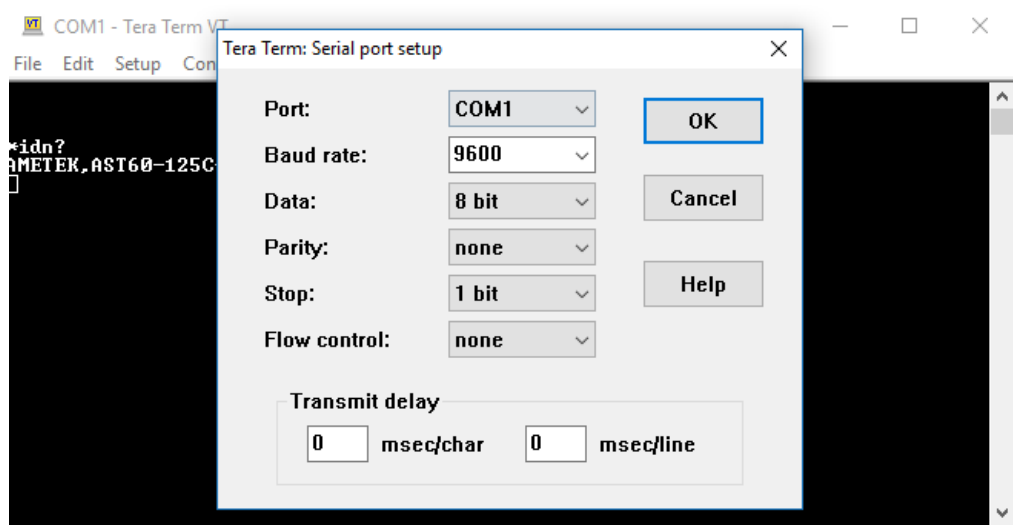


Figure 4-6: Baud rate Setting in Tera Term

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

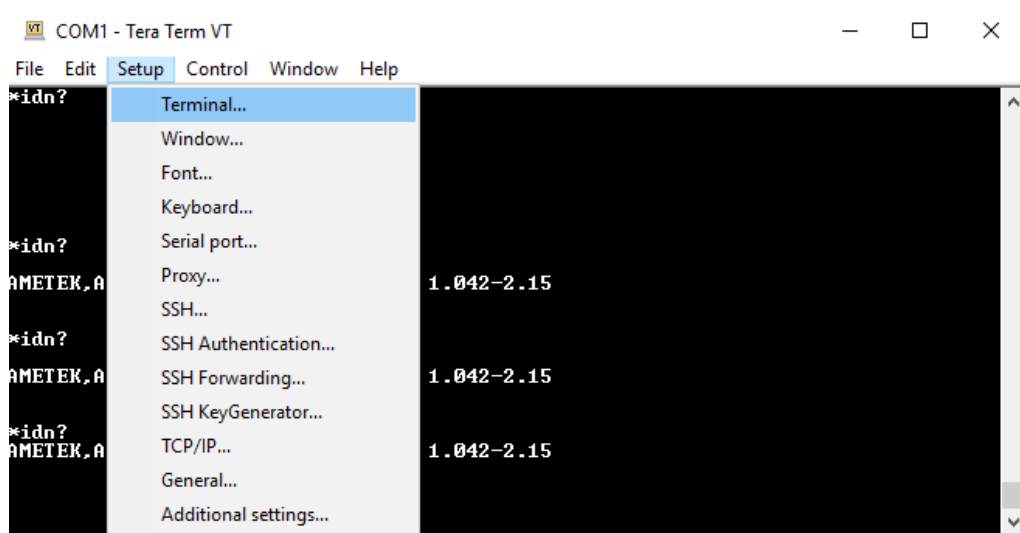


Figure 4-7: Terminal option in Tera Term

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

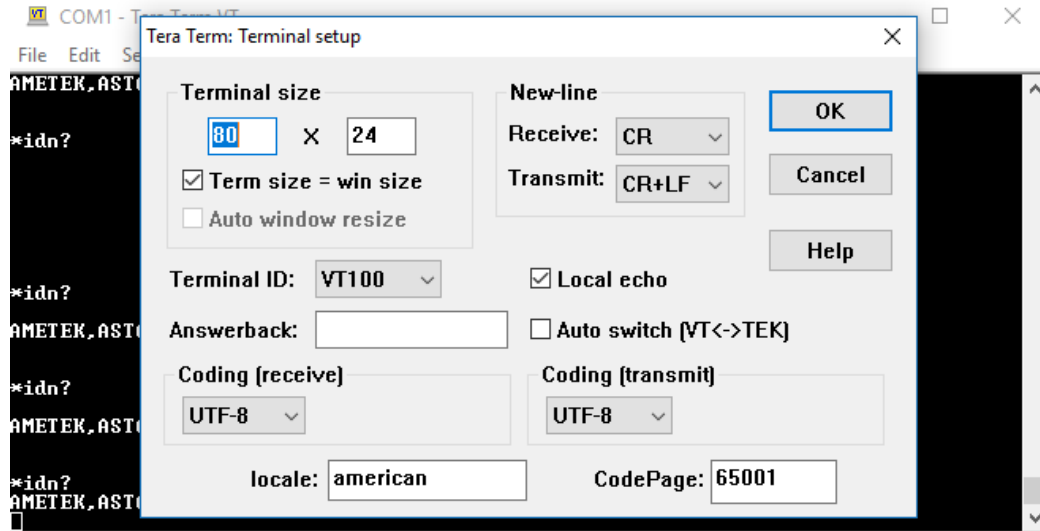


Figure 4-8: Terminal Setup in Tera Term

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

This page intentionally left blank.

USB CONFIGURATIONS AND REMOTE PROGRAMMING

5.1 Rear Panel

This section provides illustration of the Asterion DC Multioutput Series power supply's rear panel layout, 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.

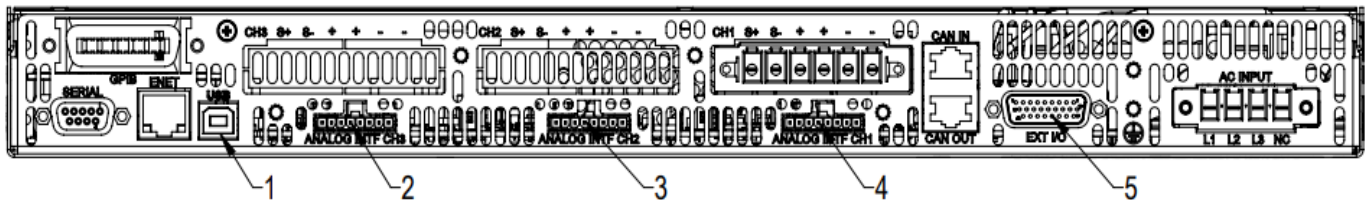


Figure 5-1: Rear Panel – Type-B USB Interface

- 1- Type-B USB Connector
- 2- Remote Analog Programming Connector Channel 3
- 3- Remote Analog Programming Connector Channel 2
- 4- Remote Analog Programming Connector Channel 1
- 5- External User Control Interface connector

5.2 USB Setup Procedure

5.2.1 Port Number Identification

Port Number of the Asterion DC Multioutput Series 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)

Note: If Multiple COM ports are shown, disconnect all the COM ports and connect only the Asterion DC Multioutput power supply and check for the new COM port.

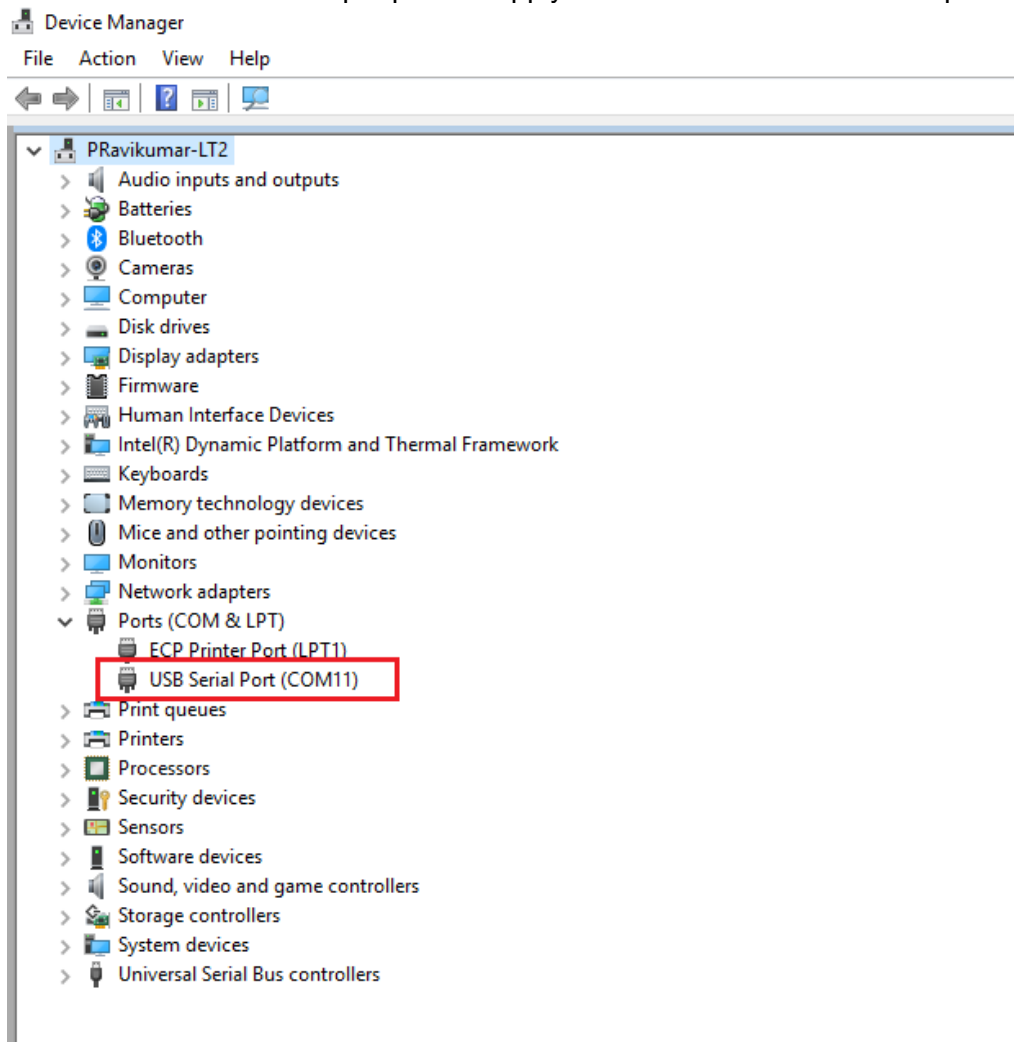


Figure 5-2: Port identification

5.2.2 Communication test using serial interface program

1. Use one of the available programs for serial communication, such as Tera Term.
2. If you choose to use Tera Term:
 - 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)

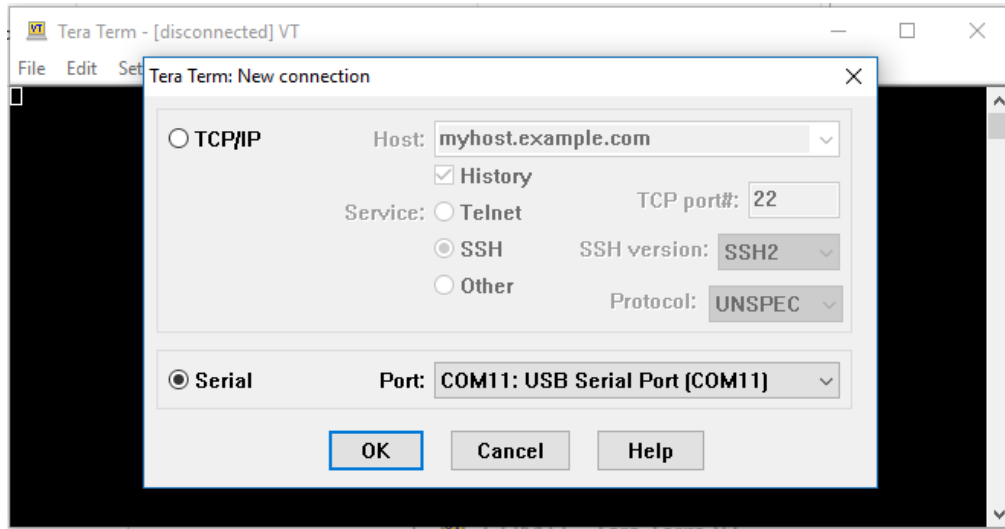


Figure 5-3: Port identification in Tera Term

- Select setup and select serial port in the setup window.
- Set the baud rate to 115200, 8 data bits, no parity, 1 stop bit, and none flow control. (Refer to Figure 5-4)

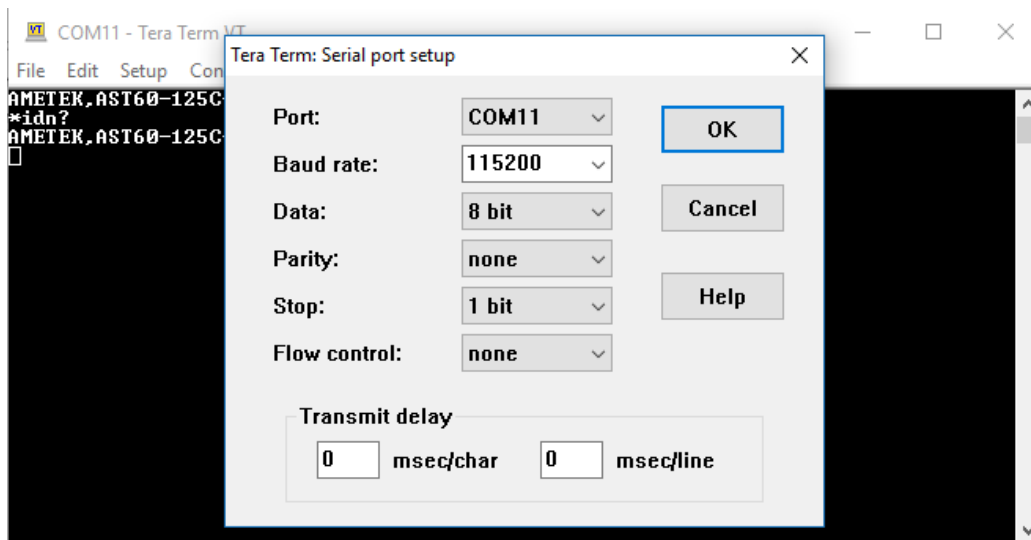


Figure 5-4: Baud rate selection in Tera Term

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

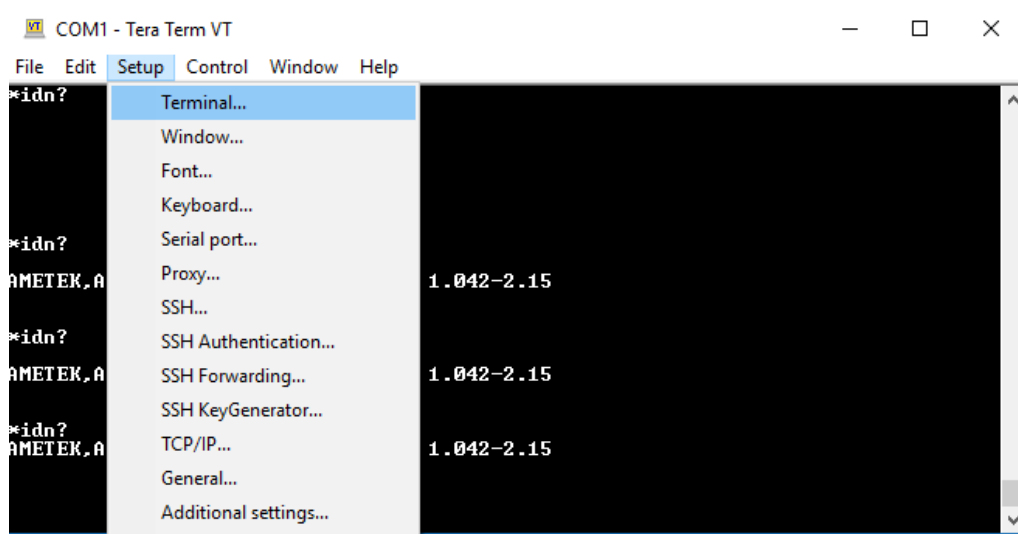


Figure 5-5: Terminal selection in Tera Term

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

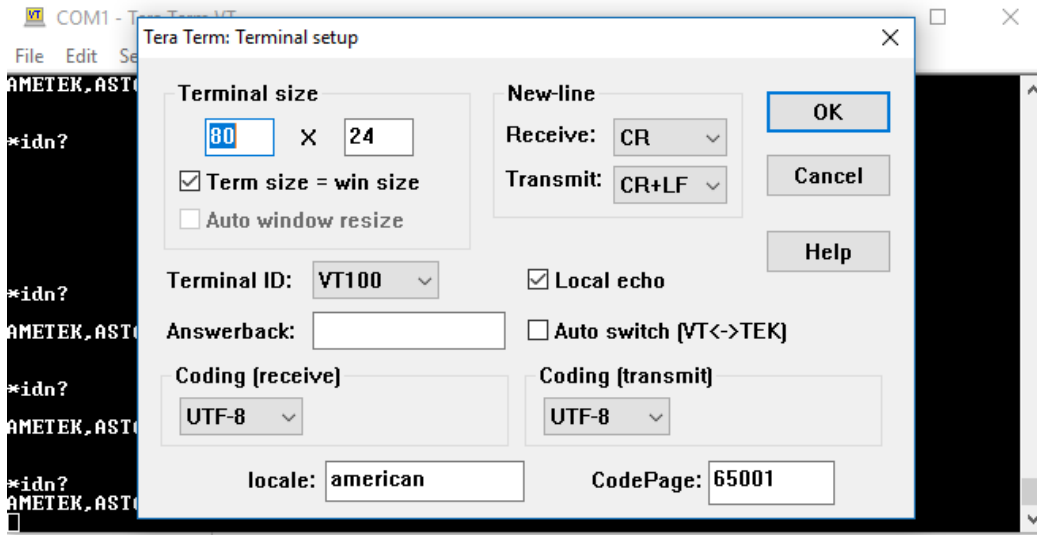


Figure 5-6: Terminal setup in Tera Term

- 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.

ETHERNET CONFIGURATION AND REMOTE PROGRAMMING

6.1 Introduction

This section covers the Remote Programming Ethernet Interface for the Asterion DC Multioutput Series power supply. This configuration enables you to operate Asterion DC Multioutput Series 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.

6.2 Rear Panel

This section provides illustrations of the Asterion DC Multioutput Series power supply's rear panel layout. 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.

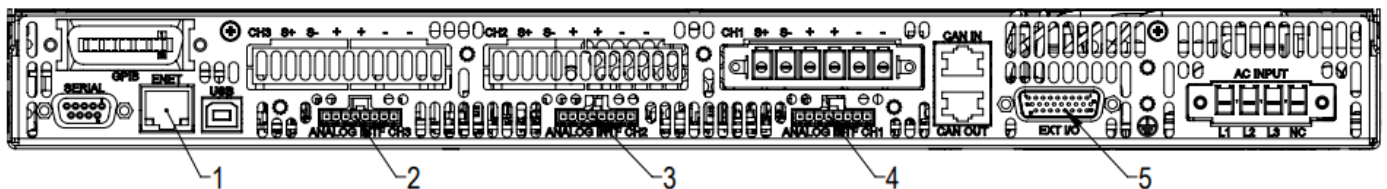


Figure 6-1: Rear Panel - Ethernet Interface

- 1- RJ45-Ethernet connector
- 2- Remote Analog Programming Connector Channel 3
- 3- Remote Analog Programming Connector Channel 2
- 4- Remote Analog Programming Connector Channel 1
- 5- External User Control Interface connector

6.3 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

6.3.1 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.

6.4 Ethernet Setup Procedure

There are three ways to setup the Ethernet network in the Asterion DC Multioutput 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 Multioutput Series 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

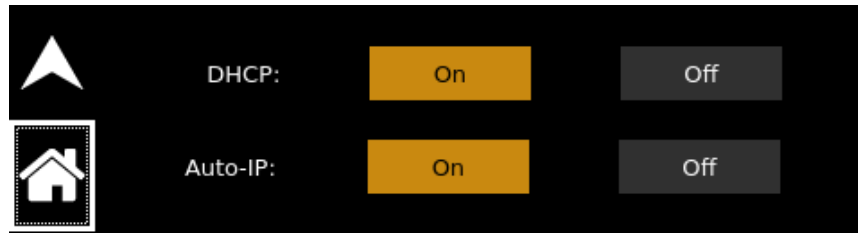


Figure 6-2: LAN Screen (Configure)

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 192.168.0.200. See Figure 6-3. This configuration is not usable for network connection.

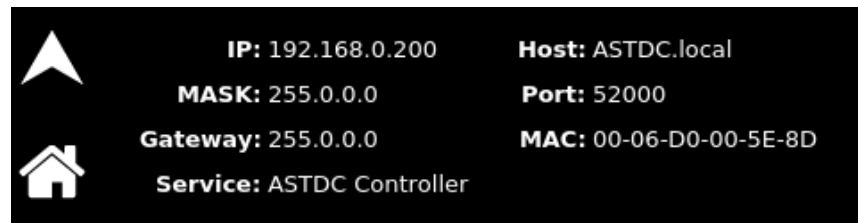


Figure 6-3: LAN Screen

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, Press 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.

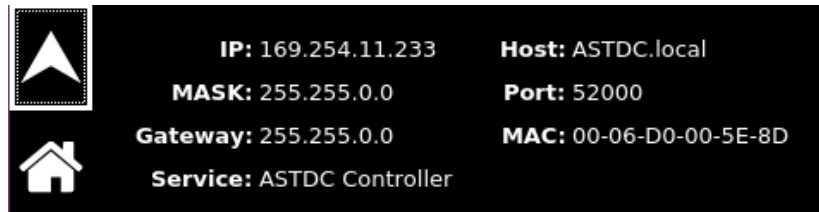


Figure 6-4: LAN Screen (Settings)

- d) Using a computer serial communications program such as Tera Term, set for the same baud rate as Asterion DC Multioutput Series power supply (Refer to Section 4.2.2) no parity, 8 data bits, 1 stop bit. Use SCPI command **SYST:NET:IP?** to get the IP address.

To find RS232 Settings of the power supply, navigate to Home → Control Interface → RS232 → RS232 Settings, Refer to Figure 6-5.



Figure 6-5: RS232 Screen (Settings)

5. The 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 6-12.

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 Multioutput Series 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 Multioutput Series 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, refer Section 6.4.1.3.
10. The Asterion DC Multioutput Series power supply Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page. Refer to Figure 6-12.

6.4.3 Network Setup using Static IP

This setup requires that DHCP is OFF. DHCP mode can be made OFF using the front

panel or using the serial interface.

6.4.3.1 STATIC IP SETUP USING FRONT PANEL

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

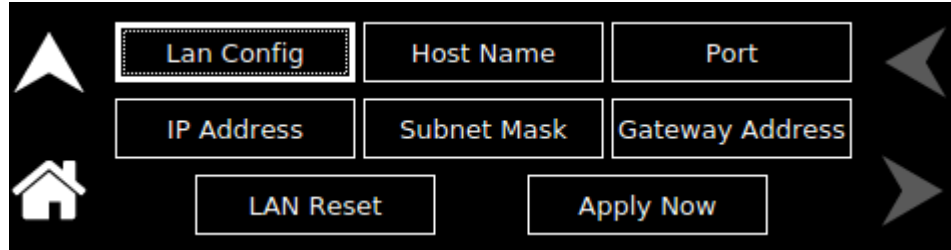


Figure 6-6: LAN configure (Settings)

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

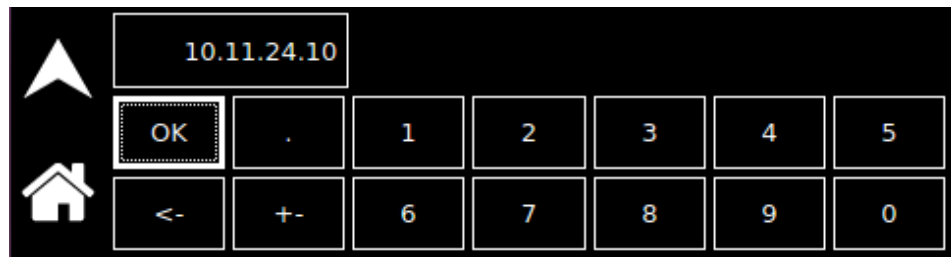


Figure 6-7: LAN Screen (Configure Static IP)

3. Enter the required IP address (Refer to Figure 6-6).

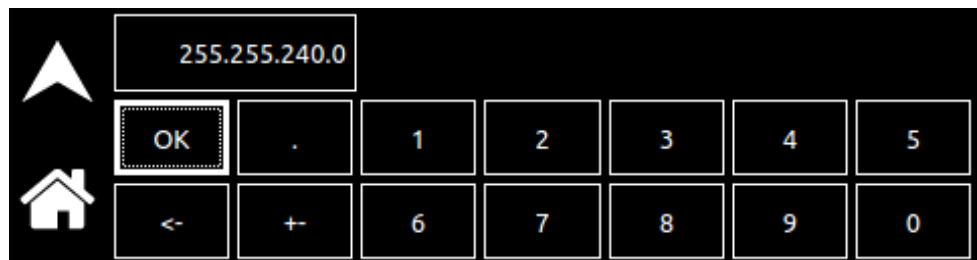


Figure 6-8: LAN Screen (Subnet Mask)

4. Enter the Subnet Mask (Refer to Figure 6-8).

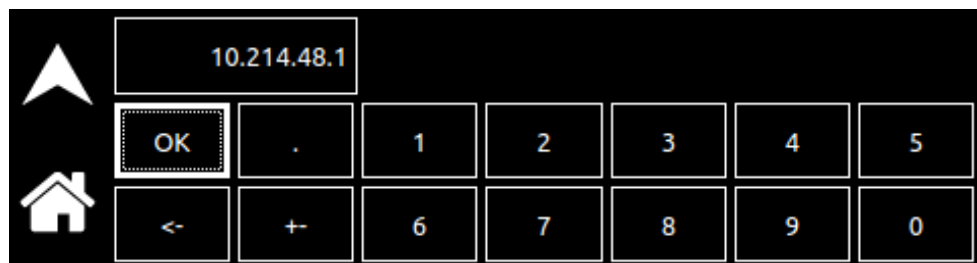


Figure 6-9: LAN Screen (Gateway Address)

5. Enter the Gateway Address (Refer to Figure 6-9).
6. Press “Apply Now” button for the settings to take effect.
7. The Asterion DC Multioutput Series power supply Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page. Refer to Figure 6-12.

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 Multioutput Series 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. Refer to section 10.8 for more details on SCPI commands.
4. Set the IP address by issuing SYST:NET:IP “xxx.xxx.xxx.xxx” (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”.

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

5. Set the subnet mask with SYST:NET:MASK xxx.xxx.xxx.xxx.
6. Set the gateway with SYST:NET:GATE xxx.xxx.xxx.xxx.
7. Type SYST:NET:APPLY, press enter to apply the Static IP configuration.
8. After configuring all settings, verify with the queries, SYST:NET:IP?, SYST:NET:GATE? and SYST:NET:MASK?.
9. The Asterion DC Multioutput Series power supply Ethernet hardware is now configured. Open Web browser and enter the IP address of the power supply to view the Home page. Refer to Figure 6-12.
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 0) or the RS232C interface.

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. ASA060200400C-E110), Manufacturer (AMETEK Programmable Power), AMETEK Logo and the Device name (e.g. LXI-ASA060200400C-E010-SER123). Below the Asterion DC Multioutput Series power supply banner are four tabs, each linked to its corresponding page. Refer to Figure 6-10.



Figure 6-10: Asterion DC Multioutput Series 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 6.3.1) for accessing the IP Configuration Tab. Refer to Figure 6-11.

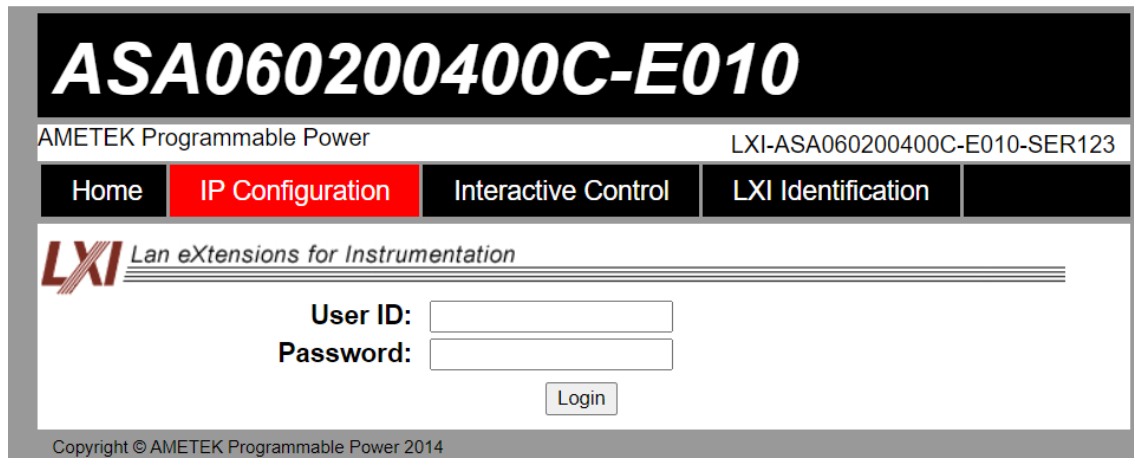


Figure 6-11: Login Window

6.5.1 Home

This is the default, information-only page (see Figure 6-12). It displays all the current information about the power supply that you are connected to:

ASA060200400C-E010

AMETEK Programmable Power LXI-ASA060200400C-E010-SER123

[Home](#)
[IP Configuration](#)
[Interactive Control](#)
[LXI Identification](#)

LXI Lan eXtensions for Instrumentation

Model: ASA060200400C-E010
Manufacturer: AMETEK Programmable Power
Serial Number: SER123

VISA Resource:

- TCPIP0::10.214.54.89::inst0::INSTR
- TCPIP0::10.214.54.89::52000::SOCKET

LXI Version: 1.5 LXI Device Specification 2016
Host Name: 10.11.24.10.local
Description: Viper Controller (2)
MAC Address: 00:06:D0:00:5E:84
TCP/IP Address: 10.214.54.89
Subnet Mask: 255.255.240.0
Gateway: 10.214.48.1
Firmware Version: 3.131
Device Identify:

Copyright © AMETEK Programmable Power 2014

Figure 6-12: Asterion DC Multioutput Series power supply Home Page

- The **Model** number, the **Manufacturer**, and the **Serial Number** of your Asterion DC Multioutput Series 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 on the front panel Screen of the power supply. Refer to Figure 6-13.

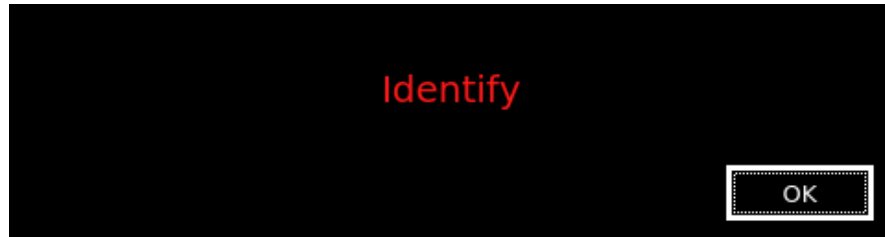


Figure 6-13: Asterion DC Multioutput Series power supply Device identify screen

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 6.3.1). 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 6-14).

ASA060200400C-E010

AMETEK Programmable Power LXI-ASA060200400C-E010-SER123

Home **IP Configuration** Interactive Control LXI Identification

LXI Lan eXtensions for Instrumentation

Host Name:

Description:

TCP/IP Configuration: DHCP Static IP

Auto IP:

IP Address:

Subnet Mask:

Gateway:

DNS Server:

Copyright © AMETEK Programmable Power 2014

Figure 6-14: IP Configuration Page

Host Name: the default host name is ASTDC.

You may change this name if 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 the following 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 6-12. 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) and click again to uncheck (disable Auto IP) (Refer to Figure 6-14)

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 power supply. The commands and response to the query command can be seen on the web page (Refer to Figure 6-15).

The screenshot shows a web interface for an AMETEK power supply. At the top, the model number **ASA060200400C-E010** is displayed in large white text on a black background. Below this, the text "AMETEK Programmable Power" is on the left and "LXI-ASA060200400C-E010-SER123" is on the right. A navigation bar contains four buttons: "Home", "IP Configuration", "Interactive Control" (which is highlighted in red), and "LXI Identification". The main content area is a scrollable text box showing the following SCPI queries and their responses:

```
sour1:volt?  
0.000  
sour2:volt?  
0.000  
sour3:volt?  
0.000  
sour1:curr?  
0.000  
sour2:curr?  
0.000  
sour3:curr?  
0.000
```

At the bottom of the page, there is a "Command:" label followed by an empty text input field and a "Send Command" button. The footer of the page reads "Copyright © AMETEK Programmable Power 2014".

Figure 6-15: Interactive Control Page

6.5.4 LXI Identification

The LXI Identification web page displays the LXI parameters of the Asterion DC Multioutput Series power supply (Refer to Figure 6-16).

ASA060200400C-E010

AMETEK Programmable Power LXI-ASA060200400C-E010-SER123

Home | IP Configuration | Interactive Control | **LXI Identification**

LXI Lan eXtensions for Instrumentation

Model: ASA060200400C-E010
Manufacturer: AMETEK Programmable Power
Serial Number: SER123

VISA Resource:

- TCPIP0::10.214.54.89::inst0::INSTR
- TCPIP0::10.214.54.89::52000::SOCKET

LXI Version: 1.5 LXI Device Specification 2016
Host Name: 10.11.24.10.local
Description: Viper Controller (2)
MAC Address: 00:06:D0:00:5E:84
TCP/IP Address: 10.214.54.89
Subnet Mask: 255.255.240.0
Gateway: 10.214.48.1
DHCP Enabled: true
AutoIP Enabled: true
Identification URL: <http://10.214.54.89/lxi/identification>
Firmware Revision: 3.131

Copyright © AMETEK Programmable Power 2014

Figure 6-16: LXI Identification Page

This page intentionally left blank.

7

IEEE 488.2 GPIB CONFIGURATIONS AND REMOTE PROGRAMMING

7.1 Rear Panel

This section provides illustrations of the Asterion DC Multioutput Series power supply's rear panel layout. Refer to Figure 7-1, regardless of the layout, the component functions are common across all models, and those that are pertinent to the IEEE 488.2 GPIB(Optional) options are described here.

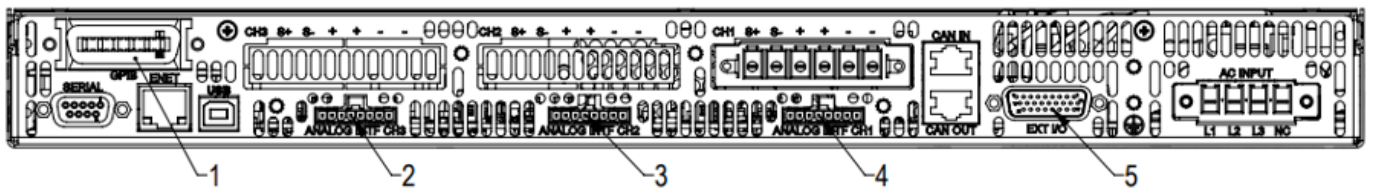


Figure 7-1: Rear Panel – IEEE 488.2 GPIB Interface

- 1- IEEE 488.2 GPIB Connector
- 2- Remote Analog Programming Connector Channel 3
- 3- Remote Analog Programming Connector Channel 2
- 4- Remote Analog Programming Connector Channel 1
- 5- External User Control Interface connector

7.2 Remote Programming Via IEEE 488.2 GPIB connector

7.2.1 Address Selection

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

Asterion DC Multioutput power supply's GPIB address can be configured by using the front panel menu, Refer to Figure 7-2. To navigate to GPIB screen, go to Home → Control Interface → GPIB.

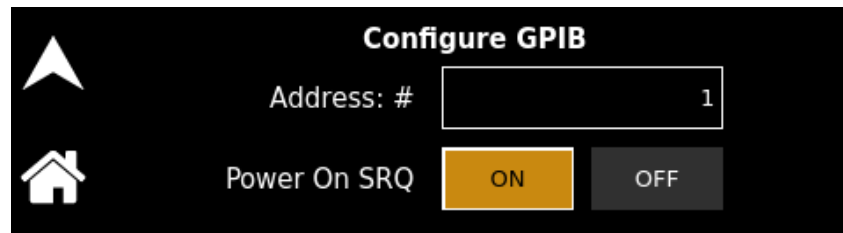


Figure 7-2: GPIB Screen

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

PON SRQ can be enable using the front panel menu, see Figure 7-2. When PON SRQ is enabled, a GPIB service request will be sent by the power supply to the computer controller on Power ON.

7.2.3 Shield Ground

Connects GPIB cable shield to chassis ground.

7.2.4 IEEE 488.2 GPIB Setup Procedure

1. Set the GPIB address via the front panel menu, see Figure 7-2.
2. Connect GPIB cable from the controlling computer to the Asterion DC Multioutput Series power supply.
NOTE: If operating in an inherently noisy environment, e.g., high RF or other radiated emissions, a double-shielded GPIB cable is recommended.
3. Connect power to the unit and turn on the unit.
4. Using a GPIB communication software, test the communication interface by issuing the *IDN? Command. This returns the supply's model, serial numbers, and software version(s). This command does not affect the output of the supply.

This page intentionally left blank.

8

REMOTE ANALOG PROGRAMMING INTERFACES

8.1 Introduction

This section provides illustrations of the Asterion DC Multioutput Series power supply's rear panel layout. Refer to Figure 8-1, regardless of the layout, the component functions are common across all models and those that are pertinent to the Remote Analog Programming Interface Connector is described here. The analog programming functionality can be exercised from either of the communication interfaces (RS232, USB, Ethernet and IEEE 488.2 GPIB).

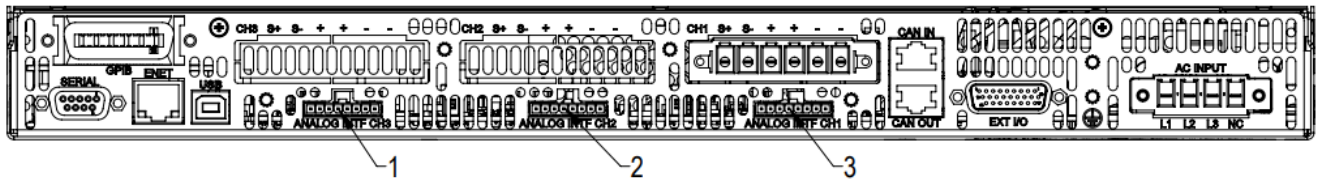


Figure 8-1: Rear Panel - Analog Programming Connector

- 1- Remote Analog Programming Connector 1 of Channel 3
- 2- Remote Analog Programming Connector 2 of Channel 2
- 3- Remote Analog Programming Connector 3 of Channel 1

8.2 Remote Analog Programming Connector details

There are three 8-pin rectangular connectors (Refer to Figure 8-2) located at the rear panel for each channel and provides remote Analog Programming control signals for individual channels to increase the user's operating control of the power supply. The mating receptacle is TE Connectivity P/N 2-1445055-8 with 8-Pin male terminals. The connector terminals accommodate wire sizes from #24 - #30.

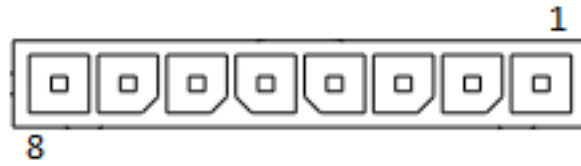




Figure 8-2: Remote Analog Programming Interface Connector Pin-Out

Pin	Signals	Type	Description
1	VPRG-VSOUR	ANALOG INPUT	<p>Remote control input for voltage programming using a voltage source connected between this pin 1 and 3 signal return.</p> <p>Signal return: Pin 3</p> <p>Range: Full scale Voltage could be set by the user from 5V to 10V.</p> <p>Input impedance: 20 kΩ, typical</p>
2	VPRG-ISOUR	ANALOG INPUT	<p>Remote control input for voltage programming using a resistance connected between this pin and signal return. Current Source of 1 mA is internally connected to this pin to enable resistance programming.</p> <p>Signal return: Pin 3</p> <p>Range: Full scale Voltage could be set by the user from 5kΩ to 10kΩ.</p> <p>Note: Do not exceed resistance of maximum 10 kΩ</p>
3	PRG-RTN	PROGRAMMING RETURN	<p>Return for Pin 1 and 2.</p> <p>Pin 3 – PRG-RTN is shorted with Pin 6 – MON-RTN internal to the power supply.</p>
4	IPRG-VSOUR	ANALOG INPUT	<p>Remote control input for current programming using a voltage source connected between this pin and signal return.</p> <p>Signal return: Pin 3</p> <p>Range: Full scale Current could be set by the user from 5V to 10V.</p> <p>Input impedance: 20 kΩ, typical</p>
5	IPRG-ISOUR	ANALOG INPUT	<p>Remote control input for current programming using a resistance connected between this pin and signal return. Current Source of 1 mA is connected to this pin from the power supply to enable resistance programming.</p> <p>Signal return: Pin 3</p> <p>Range: Full scale Current could be set by the user from 5kΩ to 10kΩ.</p> <p>Note: Do not exceed resistance of maximum 10 kΩ</p>

6	MON-RTN	MONITOR RETURN	Return for Pin 7 and 8. Pin 3 – PRG-RTN is shorted with Pin 6 – MON-RTN internal to the power supply.
7	VMON	ANALOG OUTPUT	Monitor signal for output voltage Signal return: Pin 6 Range: 0 V to 10 V corresponds to 0-100% full-scale output. Output impedance: 100 Ω, typical Minimum recommended Load: 100 kΩ, typical Maximum Load: 20 kΩ
8	IMON	ANALOG OUTPUT	Monitor signal for output current Signal return: Pin 6 Range: 0 V to 10 V corresponds to 0-100% full-scale output. Output impedance: 100 Ω, typical Minimum recommended Load: 100 kΩ, typical Maximum Load: 20 kΩ

Table 8-1: Remote Analog Programming Connector pin out details

The pinout function mentioned in Table 8-1 and Figure 9-2 is identical for each channel connectors. To program desired channel corresponding analog channel connector to be utilized.

	CAUTION! Return signal of all the three channels are shorted internal to the power supply. Applying a voltage potential between the channel return signal of the individual channels would damage the power supply.
	CAUTION! Analog programming interface signals are isolated from negative output terminal; Isolation voltage is rated for ±600 VRMS maximum; operation of Isolated Analog Interface signals should be at SELV safety voltage conditions to chassis ground.

8.3 Remote ANALOG Programming

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

8.3.1 REMOTE PROGRAMMING BY RESISTANCE

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

```
SOUR<n>:ANALOG:PROG:SOUR CURR
```

SOUR<n>:CURR:PROG:SOUR CURR

Note: The Data format “<n>” represents the Channel Number (1, 2 and 3).

8.3.1.1 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<n>:ANALOG:PROG:SOUR CURR

SOUR<n>:CURR:PROG EXT

The resistance-programming default coefficient for output current is Channel current rating (100% rated output current) / 5kΩ, with input at Pin 5 (IPRG_ISOUR) and return to Pin 3 (RTN_PRG). Refer to Figure 8-3. An internal current source, factory-set at 1 mA, from Pin 5 is utilized to drive the resistance. This produces a transfer function for output current, as follows:

$I_{out} = R * ((100\% \text{ rated output current}) / 5k\Omega)$, with R in kohms.

Full Scale current programming resistance can be modified from default 5kOhms to any other value (from 5kΩ to 10kΩ).

For example, To set 10kΩ as full scale for Current programming send below command.

SOUR<n>:CURR:PROG:FSCR 10

Then the transfer function for output current, as follows:

$I_{out} = R * ((100\% \text{ rated output current}) / \text{FSC } k\Omega)$, 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.

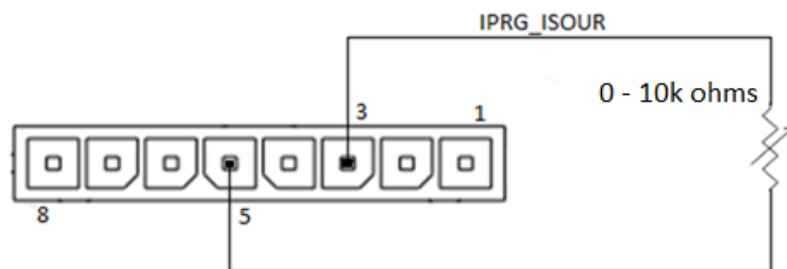


Figure 8-3: Remote Current Programming Using Resistance

8.3.1.2 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<n>:ANALOG:PROG:SOUR CURR

SOUR<n>:VOLT:PROG EXT

The resistance-programming default coefficient for output voltage is Channel Voltage rating (100% rated output voltage) / 5 kΩ, with input at Pin 2 (VPRG_ISOURL) and return to Pin 3 (RTN_PRG). Refer to Figure 8-4. An internal current source, factory-set at 1 mA, from Pin 2 is utilized to drive the resistance. This produces a transfer function for output voltage, as follows:

$$V_{out} = R * ((100\% \text{ rated output voltage}) / 5k\Omega), \text{ with } R \text{ in kohms.}$$

Full Scale voltage programming resistance can be modified from default 5kΩ to any other value, from 5kΩ to 10kΩ. For example, to set 10kΩ as full scale for Voltage programming send below command:

SOUR<n>:VOLT:PROG:FSCR 10

Then the transfer function for output voltage, as follows:

$$V_{out} = R * ((100\% \text{ rated output voltage}) / \text{FSC } k\Omega), \text{ with } R \text{ in kohms.}$$

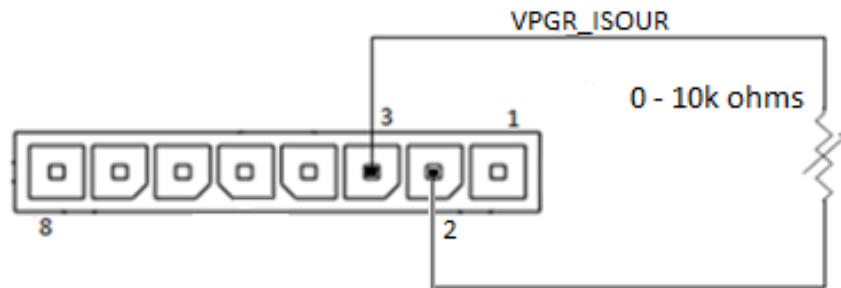


Figure 8-4: Remote Power Programming Using Resistance

8.3.2 Remote Programming BY VOLTAGE

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

SOUR<n>:ANALOG:PROG:SOUR VOLT

SOUR<n>:VOLT:PROG:SOUR VOLT

Note: The Data format “<n>” represents the Channel Number (1, 2 and 3).

8.3.2.1 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<n>:ANALOG:PROG:SOUR VOLT

SOUR<n>:CURR:PROG EXT

The DC voltage source is connected between Pin 4 (IPRG<n>_VSOUR) and the return Pin 3 (RTN_PRG), refer Figure 8-5. 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:

$I_{out} = V_{dc} * (100\% \text{ rated output current}) / 10 \text{ VDC}$, with Vdc in volts, or

$I_{out} = V_{dc} * (100\% \text{ rated output current}) / \text{FSC VDC}$, with Vdc in volts.

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

SOUR<n>:CURR:PROG:FSC 5

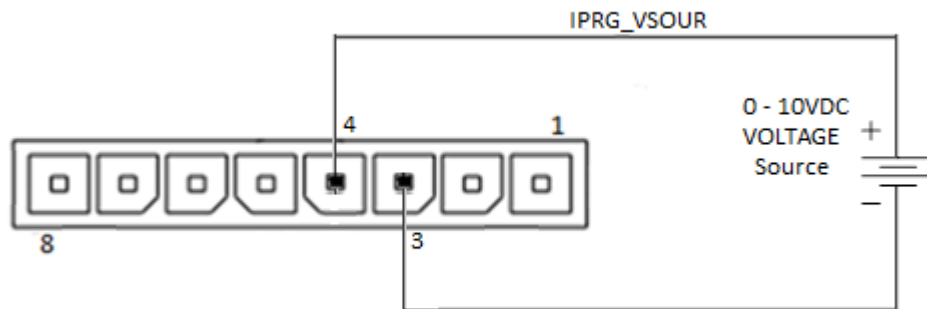


Figure 8-5: Remote Current Programming Using 0-5 VDC or 0-10 VDC Source

8.3.2.2 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<n>:ANALOG:PROG:SOUR VOLT

SOUR<n>:VOLT:PROG EXT

The DC voltage source is connected between Pin 1 (VPRG<n>_VSOUR) and the return Pin 3 (RTN_PRG). Refer to Figure 8-6. 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:

$V_{out} = V_{dc} * (100\% \text{ rated output voltage}) / 10 \text{ VDC}$, with Vdc in volts, or

$V_{out} = V_{dc} * (100\% \text{ rated output voltage}) / \text{FSC VDC}$, with Vdc in volts.

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

SOUR<n>:VOLT:PROG:FSCR 5

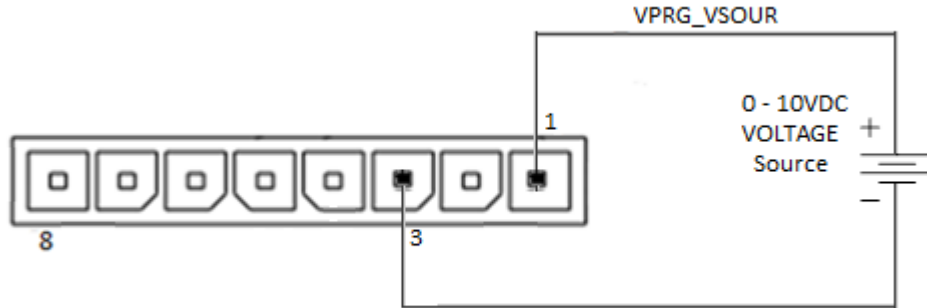


Figure 8-6: Remote Voltage Programming Using 0-10 VDC Source

8.4 Remote MONITOR OUTPUTS

Asterion DC Multioutput Series power supply provides independent analog voltage monitor signals for the Output Voltage and Current.

1. Voltage Monitor Output is in Pin 7, Return Pin 6.
2. Current Monitor Output is in Pin 8, Return Pin 6.
3. Full Scale Monitor voltage would correspond to rated output of the power supply. Default Full Scale Monitor Output is 10V. Same can be changed from 5V to 10 V by issuing following SCPI command. For example, to set to 5V as full scale for voltage monitoring send below command:
SOUR<n>:VOLT:MON:FSC 5
4. For the Current monitor to change the full-scale value to 5V, following SCPI command to be used.
SOUR<n>:CURR:MON:FSC 5

This page intentionally left blank.

REMOTE EXTERNAL USER CONTROL INTERFACE

9.1 Introduction

This section provides illustrations of the Asterion DC Multioutput Series power supply's rear panel layout. 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 External user control Interface Connector is described here. The external user Control interface control functionality can be exercised from either of the communication interfaces (RS232, USB, Ethernet and IEEE 488.2 GPIB).

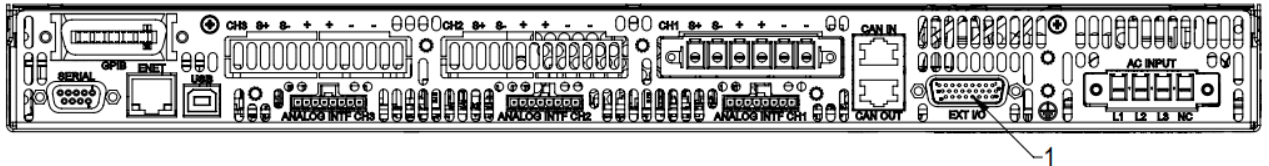


Figure 9-1: Rear Panel – External User Control Interface

1-Remote External User Control Interface Connector

9.2 Remote External User Control Interface Connector details

A 26-pin Subminiature-D connector (Refer to Figure 9-2) located at the rear panel provides remote external user interface control signals for all three channels individually to increase the user's operating control of the power supply. The mating receptacle is Norcomp P/N 180-026-103L001 with 26-Pin male terminals. The connector terminals accommodate wire sizes from 24AWG – 30AWG.

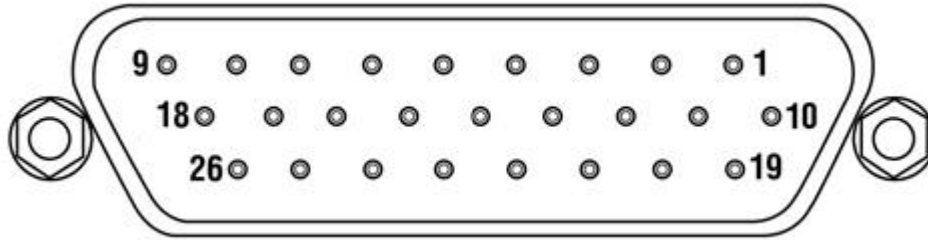


Figure 9-2: External User Control Interface Connector Pin-Out


Pin	Signal	Type	Description
Channel-1 Signals (Pin number 1 to 9)			
1	TRIG1-IN	DIGITAL INPUT	<p>Input signal, TTL active-high; provides external hardware triggering of voltage and current ramp functions.</p> <p>Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply.</p> <p>Signal return: Pin 8</p> <p>Voltage Rating: Maximum 24 V, Minimum -5V</p> <p>Low state 0.3 V max, High State 2.7 V min.</p>
2	TRIG1-OUT	DIGITAL OUTPUT	<p>Output signal, active-high; synchronization pulse of 10 ms when a change in the output occurs.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 8</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
3	CH1-FAULT-STATUS	DIGITAL OUTPUT	<p>Output Signal, High state indicates fault state of the power supply.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 8</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
4	CH1-OUTPUT-ON-OFF	DIGITAL INPUT	<p>Remote-control input for output on/off with a logic signal: a logic-high, will enable (turn-on) the output of the supply, and a logic-low signal disables (turns off) the output.</p> <p>Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply</p> <p>Signal return: Pin 8</p> <p>Voltage Rating: Maximum 24 V, Minimum -5V</p> <p>Low state 0.3 V max, High State 2.7 V min.</p>
5	CH1-CV/CC-STATUS	DIGITAL OUTPUT	<p>User digital output. Output low for Constant Voltage (CV) mode and high for Constant Current (CC) mode.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p>

			Signal return: Pin 8 Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA
6	CH1-OUTPUT-ON/OFF-STATUS	DIGITAL OUTPUT	Output Signal, High state indicates output of the power supply is enabled. Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector. Signal return: Pin 8 Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA.
7	CH1-Remote Inhibit	Input Switch/Relay Contact	Switch/Relay contact closure or direct short from this terminal to signal return is required to enable/disable the output of the power supply. Opening the contact would disable the output. Upon contact closure, if remote inhibit is selected as live mode, the fault would be cleared, and output could be enabled from the front panel or by issuing the SCPI command. If remote inhibit is selected as Latch mode, the output will be disabled, Output could not be enabled upon the contact closure. Fault should be cleared by issuing the SCPI command OUTP<n>:PROT:CLE or by clearing the fault from the front panel screen. Remote circuit must sink up to 10 mA from 5 VDC to enable. Signal return: Pin 8
8	RTN	RETURN	Return/GND. Pins 8,17 and 26 are shorted internal to the power supply.
9	NC	N/A	No Connection
Channel-2 Signals (Pin number 10 to 18)			
10	TRIG2-IN	DIGITAL INPUT	Input signal, TTL active-high; provides external hardware triggering of voltage and current ramp functions. Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply. Signal return: Pin 17 Voltage Rating: Maximum 24 V, Minimum -5V Low state 0.3 V max, High State 2.7 V min.
11	TRIG2-OUT	DIGITAL OUTPUT	Output signal, active-high; synchronization pulse of 10 ms when a change in the output occurs. Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector. Signal return: Pin 17 Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA
12	CH2-FAULT-STATUS	DIGITAL OUTPUT	Output Signal, High state indicates fault state of the power supply. Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector. Signal return: Pin 17 Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA
13	CH2-OUTPUT-ON-OFF	DIGITAL INPUT	Remote-control input for output on/off with a logic signal: a logic-high, will enable (turn-on) the output of the supply, and a logic-low signal disables (turns off) the output.

			<p>Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply</p> <p>Signal return: Pin 17</p> <p>Voltage Rating: Maximum 24 V, Minimum -5V</p> <p>Low state 0.3 V max, High State 2.7 V min.</p>
14	CH2-CV/CC-STATUS	DIGITAL OUTPUT	<p>User digital output. Output low for Constant Voltage (CV) mode and high for Constant Current (CC) mode.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 17</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
15	CH2-OUTPUT-ON/OFF-STATUS	DIGITAL OUTPUT	<p>Output Signal, High state indicates output of the power supply is enabled.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 17</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA.</p>
16	CH2-Remote Inhibit	Input Switch/ Relay Contact	<p>Switch/Relay contact closure or direct short from this terminal to signal return is required to enable/disable the output of the power supply. Opening the contact would disable the output. Upon contact closure, if remote inhibit is selected as live mode, the fault would be cleared, and output could be enabled from the front panel or by issuing the SCPI command.</p> <p>If remote inhibit is selected as Latch mode, the output will be disabled, Output could not be enabled upon the contact closure. Fault should be cleared by issuing the SCPI command <code>OUTP<n>:PROT:CLE</code> or by clearing the fault from the front panel screen.</p> <p>Remote circuit must sink up to 10 mA from 5 VDC to enable.</p> <p>Signal return: Pin 17</p>
17	RTN	RETURN	Return/GND. Pins 8,17 and 26 are shorted internal to the power supply.
18	NC	N/A	N/A
Channel-3 Signals (Pin number 19 to 26)			
19	TRIG3-IN	DIGITAL INPUT	<p>Input signal, TTL active-high; provides external hardware triggering of voltage and current ramp functions.</p> <p>Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply.</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 24 V, Minimum -5V</p> <p>Low state 0.3 V max, High State 2.7 V min.</p>
20	TRIG3-OUT	DIGITAL OUTPUT	<p>Output signal, active-high; synchronization pulse of 10 ms when a change in the output occurs.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
21	CH3-FAULT-STATUS	DIGITAL OUTPUT	Output Signal, High state indicates fault state of the power supply.

			<p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
22	CH3-OUTPUT-ON-OFF	DIGITAL INPUT	<p>Remote-control input for output on/off with a logic signal: a logic-high, will enable (turn-on) the output of the supply, and a logic-low signal disables (turns off) the output.</p> <p>Signal connects to Open-anode of opto-isolator diode with 1kΩ series resistor internal to power supply</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 24 V, Minimum -5V</p> <p>Low state 0.3 V max, High State 2.7 V min.</p>
23	CH3-CV/CC-STATUS	DIGITAL OUTPUT	<p>User digital output. Output low for Constant Voltage (CV) mode and high for Constant Current (CC) mode.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA</p>
24	CH3-OUTPUT-ON/OFF-STATUS	DIGITAL OUTPUT	<p>Output Signal, High state indicates output of the power supply is enabled.</p> <p>Open collector transistor output, Collector is connected the 26-pin connector. Emitter point of transistor is connected to common return pin of the interface connector.</p> <p>Signal return: Pin 26</p> <p>Voltage Rating: Maximum 30 V, Minimum 3V for Active High, Sink Current: 50mA.</p>
25	CH3-Remote Inhibit	Input Switch/Relay Contact	<p>Switch/Relay contact closure or direct short from this terminal to signal return is required to enable/disable the output of the power supply. Opening the contact would disable the output. Upon contact closure, if remote inhibit is selected as live mode, the fault would be cleared, and output could be enabled from the front panel or by issuing the SCPI command.</p> <p>If remote inhibit is selected as Latch mode, the output will be disabled, Output could not be enabled upon the contact closure. Fault should be cleared by issuing the SCPI command <code>OUTP<n>:PROT:CLE</code> or by clearing the fault from the front panel screen.</p> <p>Remote circuit must sink up to 10 mA from 5 VDC to enable.</p> <p>Signal return: Pin 26</p>
26	RTN	RETURN	Return/GND. Pins 8,17 and 26 are shorted internal to the power supply.

Table 9-1: Remote External User Interface connector pin out details



CAUTION! All the three channels signal returns (Pin 8, 17 and 26) are shorted internally to the power supply. Applying a voltage potential between them would damage the power supply.

**CAUTION!**

External User Control interface signals are isolated from negative output terminal; Isolation voltage is rated for ± 600 VRMS, maximum; operation of Isolated Analog Interface signals should be at SELV safety voltage conditions to chassis ground.

9.2.1 Fault common to all channels

High state in the following pin indicates the Fault in the power supply. Fault Signal Pin is 3 and return is 8 for Channel 1, pin 12 and return is 17 for Channel 2 and pin 21 and return is 26 for Channel 3. Fault Signal Pin become 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 and Maximum Current is 0.5 A

Following is the list of faults that can be generated:

S.No	Description	Explanation	Actions
1	Overvoltage Protection Fault	Output voltage exceeded Over Voltage trip limit.	<ul style="list-style-type: none"> Check sufficient margin between set voltage and OVP limit
2	DC Module OTP Fault	Power supply internal hardware temperature exceeded limit	<ul style="list-style-type: none"> Check and make sure there are 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
3	External Shutdown	External Shutdown activated from External User Interface Control Connector	<ul style="list-style-type: none"> Make sure Pin 7 and Pin 8 are connected for disabling External Shutdown of channel 1, Pin 16 and Pin 17 are connected for disabling External Shutdown of channel 2, Pin 25 and Pin 26 are connected for disabling External Shutdown of channel 3.
4	Foldback Mode Operation	Trip activated due to foldback operation setting	<ul style="list-style-type: none"> Check the foldback operation setting from <code>OUTP<n>:PROT:FOLD?</code> Query and modify as required
5	Calibration Fault	Calibration command is incorrect or entered parameter with command is incorrect.	<ul style="list-style-type: none"> Follow procedure to calibrate given in the section 13. 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
6	Fan Fault	Fault from cooling Fan	<ul style="list-style-type: none"> 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.
7	Line Drop Fault	Input voltage to the power supply is not in operating range	<ul style="list-style-type: none"> Check if input voltage to the supply is within specified range in datasheet
8	DC Module Fault	Internal DC module hardware fault	<ul style="list-style-type: none"> Try restarting the power supply if fault persists Contact factory

9	PFC Fault	Internal PFC module hardware fault	<ul style="list-style-type: none"> Try restarting the power supply if fault persists Contact factory
10	OCP Fault	Output current exceeded Over Current trip limit.	<ul style="list-style-type: none"> Check sufficient margin between set current and Over Current trip limit.
11	AUX Supply Fault	Auxiliary Supply to internal hardware fault	<ul style="list-style-type: none"> Try restarting the power supply if fault persists Contact factory
12	Line Status Changed	Input voltage changed from Low range to High range or vice versa	<ul style="list-style-type: none"> 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
13	Remote Sense Fault	Remote voltage sensing is out of range from power supply capacity or cable connected to Output and Remote Sense connector fault	<ul style="list-style-type: none"> Check if the Remote Sense cable connected to DC Output and Remote Sense Connector at rear side power supply, Cables are 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
14	PFC Module OTP Fault	Power supply internal hardware temperature exceeded limit	<ul style="list-style-type: none"> Check and make sure there are 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

9.2.2 Table 9-2: List of Faults in Asterion DC Multioutput Series Power Supply Trigger-In & Trigger Out

Asterion DC Multioutput Series power supply has Trigger-In and Trigger-Out functionalities.

Trigger-In:

Channel 1:

Applying high voltage signal on pin 1 and return pin 8 will trigger the ramp for channel 1.

Channel 2:

Applying high voltage signal on pin 10 and return pin 17 will trigger the ramp for channel 2.

Channel 3:

Applying high voltage signal on pin 19 and return pin 26 will trigger the ramp for channel 3.

***NOTE: Data format <n> represents channel number (1, 2 & 3)**

- *CLS // clear the unit to its power-on default settings.
- *RST // reset the unit.
- SOUR<n>:VOLT 20 // Set the voltage to 20V in respective channel
- SOUR<n>:CURR 20 // Set the current to 20A in respective channel
- OUTP<n>:STAT 1 // Turn on the output in respective channel
- SOUR<n>:VOLT:RAMP:HTRIG 25 5 //Give an external trigger i.e., an active high pulse of 10ms on the pin-1 (TRIGGER_IN) and pin-8 (RTN) for channel 1, pin-

10 (TRIGGER_IN) and pin-17 (RTN) for channel 2 and pin-19 (TRIGGER_IN) and pin-26 (RTN) for channel 3 to generate the Voltage Ramp on individual channels. 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 pin 2 and return pin 8 for channel 1, pin 11 and return pin 17 for channel 2 and pin 20 and return pin 26 for channel 3 when a change in the output occurs.

9.2.3 Remote Output On/Off Control

Remote output on/off control of each channel may be accomplished by contact closure of individual channels, and through an opto-isolated interface with external DC voltage source of 2.7V - 24V.

9.2.3.1 REMOTE OUTPUT ON/OFF BY CONTACT CLOSURE

Application of a contact closure

1. A short between Pin 7 and Pin 8 will enable the output of channel 1. Output can be enabled through front panel or SCPI command. See Figure 9-3 for connection requirements.
2. A short between Pin 16 and Pin 17 will enable the output of channel 2. Output can be enabled through front panel or SCPI command. See Figure 9-4 for connection requirements.
3. A short between Pin 25 and Pin 26 will enable the output of channel 3. Output can be enabled through front panel or SCPI command. See Figure 9-5 for connection requirements.

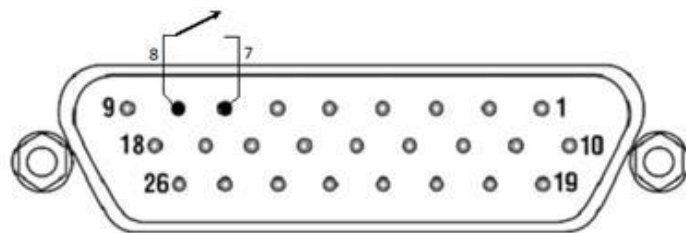


Figure 9-3: Remote Output ON/OFF by contact closure channel 1

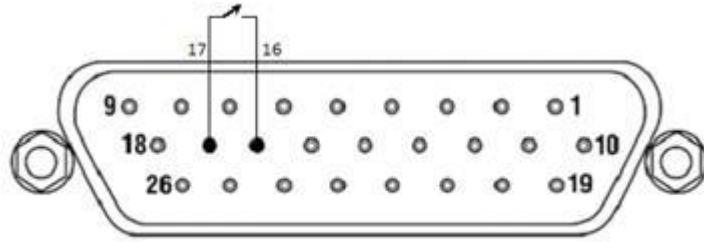


Figure 9-4: Remote Output ON/OFF by contact closure channel 2

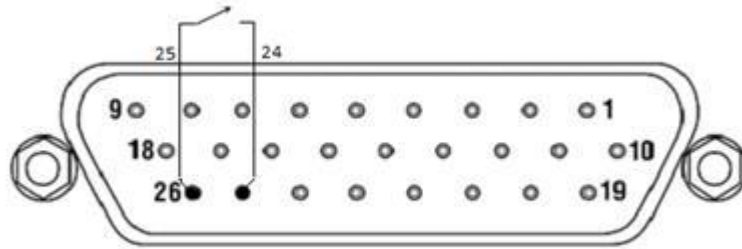


Figure 9-5: Remote Output ON/OFF by contact closure channel 3

This page intentionally left blank.

RS232/ETHERNET/ IEEE 488.2 GPIB REGISTER COMMAND OPERATION

10.1 Introduction

This section describes the operation of the Digital Interfaces RS232, USB, Ethernet and IEEE 488.2 GPIB, by using SCPI Command sets. The command set comprises of programming, query and status commands that facilitate remote control of the power supply individual Channels.

10.2 Register Definitions

The Asterion DC Multioutput Series power supply supports the RS232, Ethernet IEEE 488.2 GPIB, and SCPI status reporting data structures. These structures are comprised of status registers and status registers 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 of each channel by issuing the following command.

STATus<n>:PROTection:CONDtion?

The above command gives the present status condition of the individual channel power hardware, so the data is not latched. It is meant to be used as a polling register. The query response is hexadecimal number of 32 bits, the definition corresponding to each bit is listed in Table 10-1. Bit value "1" indicates fault and "0" indicates normal condition.

NOTE: the data format <n> represents individual channels (1, 2 & 3)

For Example:

STAT1:PROT:COND?

#H0000042

// Response for the above command

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

- Status of the Individual channels can be known by appending the respective channel number.
- By knowing the IP address of the Individual unit, condition of the unit can also be known. Channel enumerated number should still be appended while querying the individual channel in Asterion DC Multioutput Series power supply.

10.2.1.2 STATUS PROTECTION EVENT REGISTER OR FAULT REGISTER

Read the Protection Event Status Register by issuing the following command for respected channel.

STATus<n>:PROTection:EVENT?

Refer 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 of each channel with the following command.

STATus1:PROTection:ENABLE <mask> // To set Status Protection Register of channel 1

STATus2:PROTection:ENABLE <mask> // To set Status Protection Register of channel 2

STATus3:PROTection:ENABLE <mask> // To set Status Protection Register of channel 3

Read the Status Protection Enable Register of each channel with the following query command.

STATus<n>: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 respected channels output, 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 14 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	Not Used
16	0x10000	Remote Sense Fault
17-31	Not Used (Always returns zero for these bits)	

Table 10-1: Protection Condition and Event Status Registers

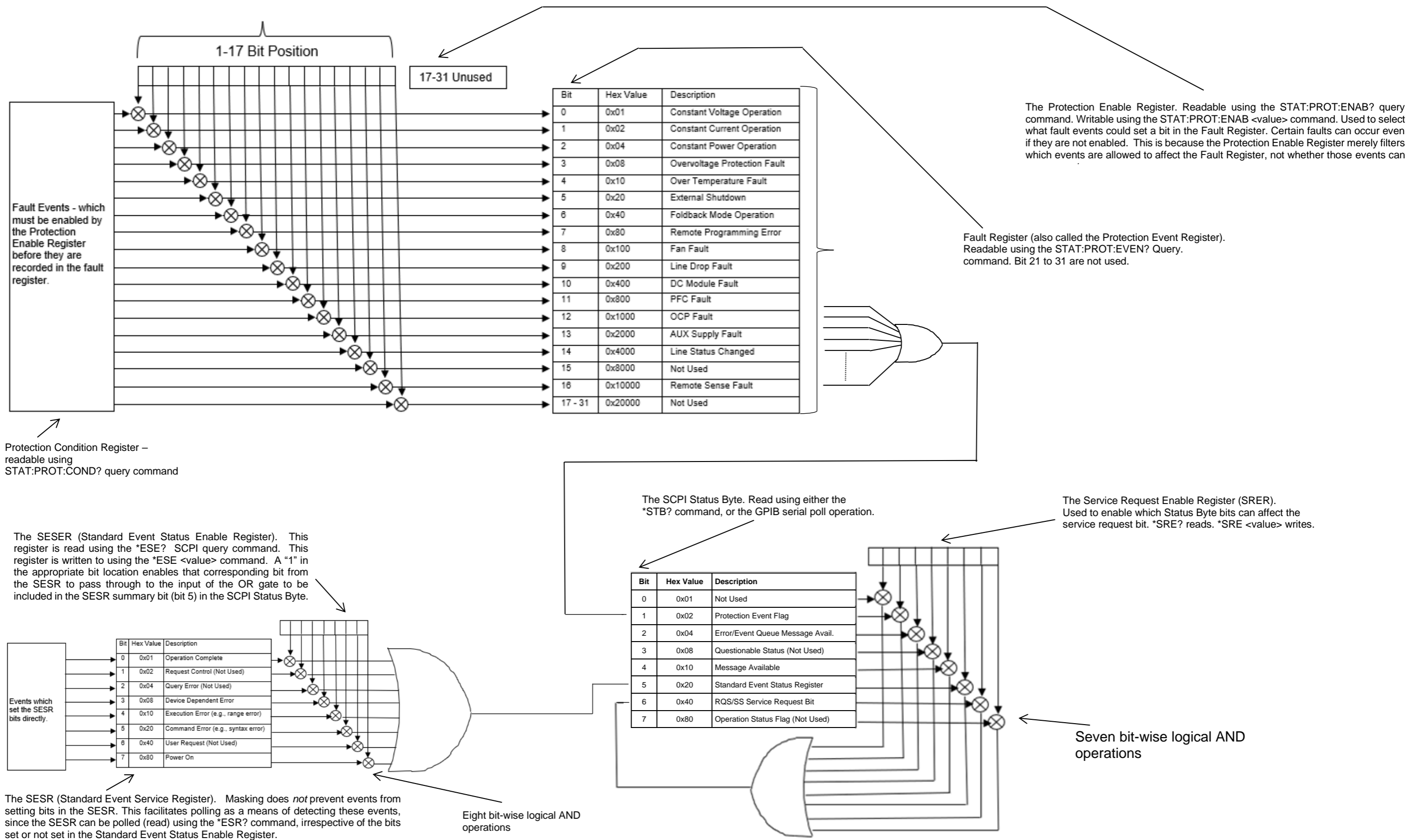


Figure 10-1: Protection Condition and Protection Event Register

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.

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

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

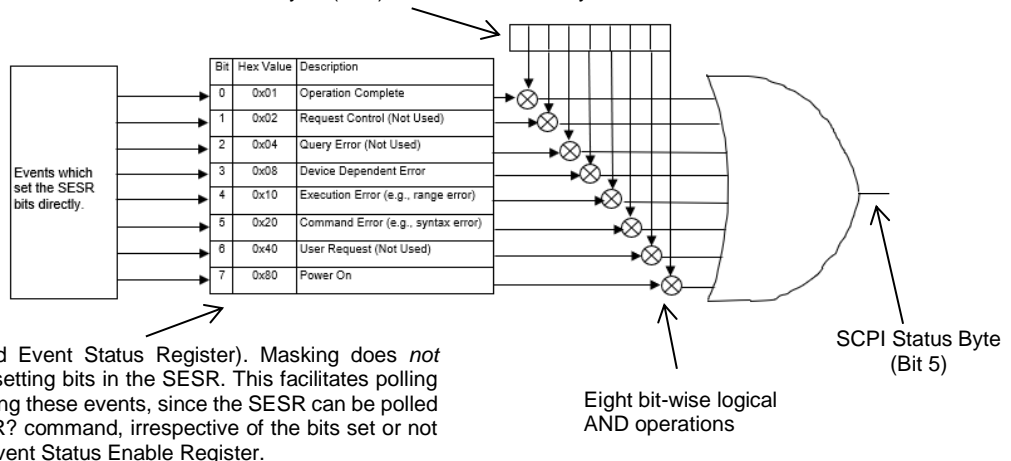


Figure 10-2: Standard Event Status and Standard Event Status Enable Register

10.2.3 SCPI Status Byte

The SCPI Status Byte registers the status of the instrument, in one of seven bits described in Table 4-1. 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 14 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

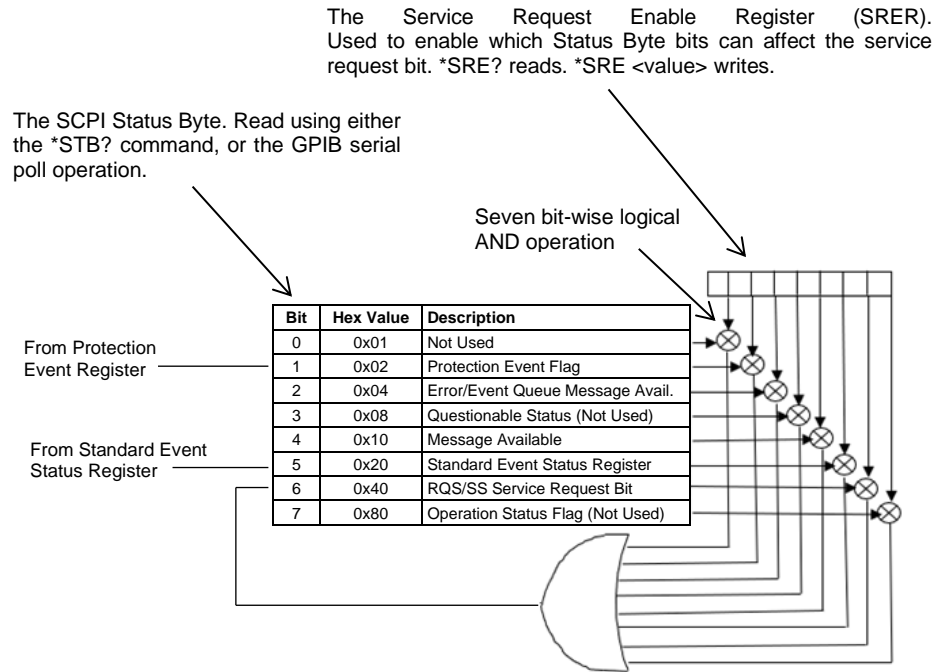


Figure 10-3: SCPI Status Byte and Service Request Enable Register

10.2.4 Error/Event Queue

The Asterion DC Multioutput Series 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 Multioutput Series 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<n>: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.

Error Code	Description
-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: <ul style="list-style-type: none"> • A <program data> element out of range, such as programming 35 volts in a 33-volt device. • A command could not be executed due to the current condition of the device.
-203	Command protected Attempted to store calibration values to EEPROM without unlocking.
-221	Settings conflict Attempted to set output greater than soft limits or to set soft limits less than output.
-222	Data out of range Parameter exceeded range of valid values.
-241	Hardware missing A legal command or query could not be executed because of a hardware fault.
-340	Calibration failed Error during calculation of calibration values occurred.
-350	Queue overflow The error queue can contain up to 10 entries. If more than 10 error/event conditions are logged before the SYSTem:ERRor? query, an overflow will occur; the last queue entry will be overwritten with error -350. When the queue overflows, the least recent error/events remain in the queue and the most recent error/events are discarded.
-360	Communication error Communications to a channel was disrupted.

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 SRQ (bit 6) for a Asterion DC Multioutput 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 Multioutput 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 [1- 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 “0” represents that the respected fault state is high and Binary “1” 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.

Asterion DC Multioutput Series power supply has self-test functionality for individual Output channels, this could be performed by sending the following SCPI command *TST<n>. The Fault state of respected output channel could be identified by issuing the following command *TST<n>? , this will return integer value of the fault state that occurs during self-test; refer Table 10-5.

The following fault state are supported by the Asterion DC Multioutput 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

10.2.1 Fault specific to channels

Power Supply faults status can be identified by sending the following command `STAT:FAULT?`. This would return Hex value of the fault state value. If the return value is 0 then there is no fault with the supply. The following fault states are supported by the Asterion DC Multioutput Series power supply, Refer Table 10-6.

Bit	Hex Value	Description
0	0x01	PFC Fault
1	0x02	Line Status Changed
2	0x04	Fan Fault
3	0x08	Line Drop Fault
4	0x10	AUX Supply Fault
5	0x20	Channel 1 Calibration Fault
6	0x40	Channel 1 Remote Sense Fault
7	0x80	OCP Fault
8	0x100	Channel 1 Overvoltage Protection Fault
9	0x200	Channel 1 DC Module OTP Fault
10	0x400	Channel 1 Foldback Mode Operation
11	0x800	Channel 1 External Shutdown
12	0x1000	Channel 1 DC Module Fault
13	0x2000	PFC Module OTP Fault
14 to 19	0x4000	Not Used (Always returns zero for these bits)
20	0x100000	AUX Supply Fault
21	0x200000	Channel 2 Calibration Fault
22	0x400000	Channel 2 Remote Sense Fault
23	0x800000	OCP Fault
24	0x1000000	Channel 2 Overvoltage Protection Fault
25	0x2000000	Channel 2 DC Module OTP Fault
26	0x4000000	Channel 2 Foldback Mode Operation
27	0x8000000	Channel 2 External Shutdown
28	0x10000000	Channel 2 DC Module Fault
29 to 35	0x20000000	Not Used (Always returns zero for these bits)
36	0x1000000000	AUX Supply Fault
37	0x2000000000	Channel 3 Calibration Fault
38	0x4000000000	Channel 3 Remote Sense Fault
39	0x8000000000	OCP Fault
40	0x10000000000	Channel 3 Overvoltage Protection Fault
41	0x20000000000	Channel 3 DC Module OTP Fault
42	0x40000000000	Channel 3 Foldback Mode Operation
43	0x80000000000	Channel 3 External Shutdown
44	0x100000000000	Channel 3 DC Module Fault
45 to 47	0x200000000000	Not Used (Always returns zero for these bits)

Table 10-6: Faults

Example: if `STAT:FAULT?` Returns “ 80008000800 ”

Binary equivalent of 0x80008000800 is “1000 0000 0000 0000 1000 0000 0000 0000 1000 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	10	11	12	13	14	15
Faults	PFC Fault	Line State Change	Fan Fault	Line Drop Fault	AUX Supply Fault	Calibration Fault	Remote sense Fault	OCP Fault	OVP Fault	DC Module OTP Fault	Foldback Fault	External Shutdown Fault	DC Fault	PFC Module OTP Fault	NA	NA
Binary Equivalent	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PFC Fault	Line State Change	Fan Fault	Line Drop Fault	AUX Supply Fault	Calibration Fault	Remote sense Fault	OCP Fault	OVP Fault	DC Module OTP Fault	Foldback Fault	External Shutdown Fault	DC Fault	PFC Module OTP Fault	NA	NA
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	46 ^{MSB}
PFC Fault	Line State Change	Fan Fault	Line Drop Fault	AUX Supply Fault	Calibration Fault	Remote sense Fault	OCP Fault	OVP Fault	DC Module OTP Fault	Foldback Fault	External Shutdown Fault	DC Fault	PFC Module OTP Fault	NA	NA
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Faults that are specific to individual output channel could be identified by sending the following SCPI command `STAT<n>:FAULT?`. This would return Hex value of the fault state in the range of [0 – FFFF] for respected channel; refer Table 10-7. If the return value is 0 then there is no fault with the output channel.

Bit	Hex Value	Description	Explanation	Actions
0	0x01	PFC Fault	Internal PFC module hardware fault	Try restarting the power supply if fault persists Contact factory
1	0x02	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
2	0x04	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.
3	0x08	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
4	0x10	AUX Supply Fault	Auxiliary Supply to internal hardware fault	Try restarting the power supply if fault persists Contact factory

5	0x20	Calibration Fault	Calibration command is incorrect or entered parameter with command is incorrect.	Follow procedure to calibrate given in the section 13. 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
6	0x40	Remote Sense Fault	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
7	0x80	OCP Fault	Output current exceeded Over Current trip limit.	Check sufficient margin between set current and Over Current trip limit.
8	0x100	Overvoltage Protection Fault	Output voltage exceeded Over Voltage trip limit.	Check sufficient margin between set voltage and OVP limit
9	0x200	DC Module OTP Fault	Power supply internal hardware temperature exceeded limit	Check and make sure there are 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
10	0x400	Foldback Mode Operation	Trip activated due to foldback operation setting	Check the foldback operation setting from OUTP<n>:PROT:FOLD? Query and modify as required
11	0x800	External Shutdown	External Shutdown activated from External User Interface Control Connector	Make sure Pin 7 and Pin 8 are connected for disabling External Shutdown of channel 1, Pin 16 and Pin 17 are connected for disabling External Shutdown of channel 2, Pin 25 and Pin 26 are connected for disabling External Shutdown of channel 3.
12	0x1000	DC Module Fault	Internal DC module hardware fault	Try restarting the power supply if fault persists Contact factory
13	0x2000	PFC Module OTP Fault	Power supply internal hardware temperature exceeded limit	Check and make sure there are 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
14	0x4000	Not Used (Always returns zero for these bits)	NA	NA
15	0x8000	Not Used (Always returns zero for these bits)	NA	NA

Table 10-7: Faults Specific to Channels

Example: if STAT<n>:FAULT? returns “C00”

Binary Equivalent of 0xC00 is 1100 0000 0000

Bit	0 ^{LSB}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 ^{MSB}
Faults	PFC Fault	Line State Change	Fan Fault	Line Drop Fault	AUX Supply Fault	Calibration Fault	Remote sense Fault	OCF Fault	OVP Fault	DC Module OTP Fault	Foldback Fault	External Shutdown Fault	DC Fault	PFC Module OTP Fault	NA	NA
Binary Equivalent	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

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

10.3 SCPI Conformance Information

The syntax of all SCPI commands implemented by the Asterion DC Multioutput Series 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 Multioutput Series 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.

<n>	The data format <n> represents channel number. 1, 2 and 3 are the valid data. "1" represents channel 1, "2" represents channel 2 and "3" represents channel 3.
<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,

```
SOURce2:VOLTage[:LEVel][:IMMediate][:AMPLitude] 120.0
```

can be written as

```
SOURce2:VOLTage 120.0
```

or

```
SOUR2: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<n>: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 <code>STAT<n>:PROT:ENAB</code> (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.
*ESE?	Returns the integer value of the Standard Event Status Enable Register. See section 10.2.2 for valid values. Response: <0+NR1>

Command	Description
*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>, <UUT model Configuration>, <serial number>,<SC firmware version>, AIB firmware version,<FPC firmware version> SC-System Controller AIB- Analog Interface Board FPC- Front Panel Controller Example: AMETEK programable power, ASA060200400C-E010,1234,1.000,1.00,1.00
*IDN<n>?	Returns the Channel identification as an ASCII string. Response: <Manufacturer>, <UUT Channel Configuration>, <serial number>,<SC firmware version>, AIB firmware version,<FPC firmware version> SC-System Controller AIB- Analog Interface Board FPC- Front Panel Controller Example: AMETEK programable power, ASA200-17ARC-E010,1234,1.000,1.00,1.00
*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<n>:PROT:ENAB (protection event enable register) is cleared by this command; other enable registers are not cleared by this command.
*RST<n>	Resets the respected channel 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<n>: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 on Status Byte for valid values.
*SRE?	Returns the integer value of the Service Request Enable Register. See section on Status Byte for valid values. Values range from 0-63 or 128-191. Response: <0+NR1>
*STB?	Returns the integer value of the Status Byte with bit 6 representing the 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 on Status Byte for valid values. Values range from 0-255. Response: <0+NR1>

Command	Description
*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>
*TST<n>	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<n>?	Returns the integer value of self-test status (*TST<n>). 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<n>
  :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?
  :PROGram
    :SOURce <string>
    :SOURce?

```

```
SOURce<n>
  :VOLTage
  :VOLTage?
    [:LEVel] <NRf>
    [:LEVel]?
      [:IMMediate] <NRf>
      [:IMMediate]?
        [:AMPLitude] <NRf>
        [:AMPLitude]?
  :LIMit <NRf>
  :LIMit?
    [:AMPLitude] <NRf>
    [:AMPLitude]?
  :PROTection <NRf>
  :PROTection?
    [:LEVel] <NRf>
    [:LEVel]?
  :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<n>
```

```

:POWer <NRf>
:POWer?
  [:LEVel] <NRf>
  [:LEVel?]
    [:IMMediate] <NRf>
    [:IMMediate?]
      [:AMPLitude] <NRf>
      [:AMPLitude?]
:MAX?
:LIMit <NRf>
:LIMit?
SOURCE<n>
  :ANALOG
  :PROGram
    :SOURce <string>
  :REMote
    :OUTPut <boolean>

```

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. The Data format “ <n> “ represents the Channel Number (1, 2 and 3).

Command	Description	SCPI
SOURCE<n>:CURRENT <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURCE<n>:CURRENT?	Returns the output current in amps or in milliamps	C
SOURCE<n>:CURRENT[:LEVel] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURCE<n>:CURRENT[:LEVel]?	Returns the output current in amps or in milliamps.	C
SOURCE<n>:CURRENT[:LEVel][:IMMediate] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURCE<n>:CURRENT[:LEVel][:IMMediate]?	Returns the output current in amps or in milliamps	C
SOURCE<n>:CURRENT[:LEVel][:IMMediate][:AMPLitude] <NRf>	Sets the output current in amps (default) or in milliamps.	C
SOURCE<n>:CURRENT[:LEVel][:IMMediate][:AMPLitude]?	Returns the output current in amps or in milliamps	C
SOURCE<n>:CURRENT:MODE?	Returns the mode of operation of the power supply. 0 - Constant voltage operation, 1 - Constant current operation and 2 - Constant power operation	N
SOURCE<n>: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<n>:CURRENT:LIMit?	Returns the upper soft limit on the programmed output current for the supply.	C

Command	Description	SCPI
SOURce<n>:CURRent:LIMit[:AMPLitude] <NRf>	Sets an upper soft limit on the programmed output current for the supply.	C
SOURce<n>:CURRent:LIMit[:AMPLitude]?	Returns the upper soft limit on the programmed output current for the supply.	C
SOURce<n>: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<n>:CURRent:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
SOURce<n>:CURRent:RAMP:ABORT	Aborts ramping and clears trigger mode.	N
SOURce<n>:CURRent:RAMP:ALL?	Returns the ramping status.	N
SOURce<n>: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<n>:CURRent:RAMP:HTRIGgered?	Returns the value of the output current to ramp to be implemented when the hardware trigger is received.	N
SOURce<n>: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. See Ramp description below.	C
SOURce<n>: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<n>:CURRent:TRIGgered <NRf>	Sets the output current to the values <NRf> when TRIGger<n>:TYPE 2 or 3 is sent	C
SOURce<n>:CURRent:TRIGgered?	Returns the current level that will be set upon receipt of the trigger.	C
SOURce<n>:CURRent:TRIGgered[:AMPLitude] <NRf>	Sets the output current to the values <NRf> when TRIGger<n>:TYPE 2 or 3 is sent	C
SOURce<n>:CURRent:TRIGgered[:AMPLitude]?	Returns the current level that will be set upon receipt of the trigger.	C
SOURce<n>:CURRent:TRIGgered:CLEar	Clears the value stored by the SOURce<n>:CURRent:TRIGger:AMPLitude command.	C
SOURce<n>: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
SOURce<n>:CURRent:PROGram?	Returns the setting of Current programming mode	N
SOURce<n>: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 5 to 10V.	N
SOURce<n>:CURRent:PROGram:FSC?	Returns the Full-scale Voltage, at which Full scale Current will be programmed.	N
SOURce<n>: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 5 to 10kOhm.	N

Command	Description	SCPI
SOURce<n>:CURRent:PROGram:FSCR?	Returns the Full-scale Resistance, at which Full scale Current will be programmed.	N
SOURce<n>:CURRent:MONitor:FSC <NRf>	Sets Full Scale voltage on Current monitor pin (IMON), when power supply is producing full scale output current.	N
SOURce<n>:CURRent:MONitor:FSC?	Returns the full scale voltage set for Current monitor pin (IMON)	N
SOURce<n>:CURRent:PROGram:SOURce <string>	Changes the source for EXTERNAL analog programming Mode. Valid arguments are VOLT and CURR. VOLT – Voltage source (0-10V range) CURR – Resistive load on constant 1mA source (0-10kOhm). 1mA current source is in built in power supply.	N
SOURce<n>:CURRent:PROGram:SOURce?	Returns the setting of External analog programming source.	N
SOURce<n>:VOLTagE <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce<n>:VOLTagE[:LEVel] <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE[:LEVel]?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce<n>:VOLTagE[:LEVel][:IMMediate] <NRf>	Sets the output voltage of the supply in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE[:LEVel][:IMMediate]?	Returns the output voltage of the supply in volts or in millivolts.	C
SOURce<n>:VOLTagE[:LEVel][:IMMediate][:AMPLitude] <NRf>	Sets the output voltage of the supply in amps (default) or in milliamps.	C
SOURce<n>:VOLTagE[:LEVel][:IMMediate][:AMPLitude]?	Returns the output voltage of the supply in amps or in milliamps.	C
SOURce<n>: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<n>:VOLTagE:LIMit?	Returns the upper soft limit set on the programmed output voltage.	C
SOURce<n>:VOLTagE:LIMit[:AMPLitude] <NRf>	Sets the upper soft limit on the programmed output voltage.	C
SOURce<n>:VOLTagE:LIMit[:AMPLitude]?	Returns the upper soft limit on the programmed output voltage.	C
SOURce<n>:VOLTagE:PROTection <NRf>	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE:PROTection?	Returns the set overvoltage protection trip point in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE:PROTection[:LEVel] <NRf>	Sets the overvoltage protection trip point in volts (default) or in millivolts.	C
SOURce<n>:VOLTagE:PROTection[:LEVel]?	Returns the set overvoltage protection trip point in volts or in millivolts.	C
SOURce<n>:VOLTagE:PROTection:TRIPped?	Returns 1 (TRIPPED) or 0 (UNTRIPPED) state of the overvoltage protection circuit.	C

Command	Description	SCPI
SOURce<n>:VOLTage:PROTection:STATe?	Returns the state 1 (ON) or 0 (OFF) if the overvoltage protection is enabled.	C
SOURce<n>:VOLTage:PROTection:CLEar	Clears the overvoltage protection circuit.	C
SOURce<n>: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<n>:VOLTage:RAMP?	Returns 1 if the ramp is in progress, and 0 if the ramp is completed.	N
SOURce<n>:VOLTage:RAMP:ABORT	Aborts ramping and clears trigger mode.	N
SOURce<n>:VOLTage:ALL?	Returns the ramping status.	N
SOURce<n>:VOLTage:RAMP:HTRIGgered <NRf><NRf>	Sets the value of the output voltage ramp to be implemented when the hardware trigger is received.	N
SOURce<n>:VOLTage:RAMP:HTRIGgered?	Returns the value of the output voltage ramp to be implemented when the hardware trigger is received.	N
SOURce<n>: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<n>:VOLTage:RAMP:TRIGgered?	Returns the output voltage to ramp	N
SOURce<n>:VOLTage:TRIGgered <NRf>	Sets the output voltage to the values stored by sending the command TRIGger<n>:TYPe 1 or 3.	C
SOURce<n>:VOLTage:TRIGgered?	Returns the voltage level that will be set upon receipt of the trigger.	C
SOURce<n>:VOLTage:TRIGgered:CLEar	Clears the value stored by the SOURce<n>:VOLTage:TRIGger:AMPLitude command.	C
SOURce<n>:VOLTage:TRIGgered:[AMPLitude] <NRf>	Sets the output voltage to the values stored by sending the command TRIGger:TYPe 1 or 3.	C
SOURce<n>:VOLTage:TRIGgered:[AMPLitude]?	Returns the stored value of the output current to be set when the SOURce<n>:VOLTage:TRIGgered command is sent.	C
SOURce<n>: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<n>:VOLTage:PROGram?	Returns the setting of Voltage programming mode	N
SOURce<n>: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 5 to 10V.	N
SOURce<n>:VOLTage:PROGram:FSC?	Returns the Full-scale Voltage, at which Full scale Voltage will be programmed.	N
SOURce<n>: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 5 to 10kOhm.	N

Command	Description	SCPI
SOURce<n>:VOLTage:PROGram:FSCR?	Returns the Full-scale Resistance, at which Full scale Voltage will be programmed.	N
SOURce<n>:VOLTage:MONitor:FSC <Nrf>	Sets Full Scale voltage on voltage monitor pin (VMON), when power supply is producing full scale output voltage.	N
SOURce<n>:VOLTage:MONitor:FSC?	Returns the full-scale voltage set for Voltage monitor pin (VMON)	N
SOURce<n>:POWer <Nrf>	Sets the maximum power limit	C
SOURce<n>:POWer?	Returns the set power.	C
SOURce<n>:POWer:MAXimum?	Returns the Maximum power that can be programmed with given hardware configuration and Input voltage conditions.	N
SOURce<n>: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 a method for protecting the load against damaging power.	N
SOURce<n>:POWer:LIMit?	Returns the upper soft limit on the programmed output power for the supply.	N
SOURce<n>:ANALOG:PROGram:SOURce <string>	Changes the source for EXTERNAL analog programming Mode. Valid arguments are VOLT and CURR. VOLT – Voltage source (0-10V range) CURR – Resistive load on constant 1mA source (0-10kOhm). 1mA current source is built in power supply.	N
SOURce<n>:ANALOG:PROGram:SOURce?	Returns the setting of External analog programming source.	N
SOURce<n>:ANALOG:REMOte:OUTPut <Nrf>	Enable/Disable the Remote output ON/OFF feature. 0 = Disables the Remote output ON/OFF feature. 1 = Enables the Remote output ON/OFF feature.	N
SOURce<n>:ANALOG:REMOte:OUTPut?	Returns the state of the Remote output ON/OFF feature.	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 ramping both at a given time.

For example, SOUR<n>:VOLT:RAMP:TRIG 1 1 followed by SOUR<n>: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.

10.6 MEASURE SCPI Command Subsystem

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

10.6.1 MEASURE SCPI Command Summary

```
MEASure<n>
  :CURRent
  :CURRent?
    :AVERage <NR1>
    :AVERage?
    :PROGram?
    :ANALOG?
  :POWER?
  :VOLTage?
  :VOLTage
    :AVERage <NR1>
    :AVERage?
    :PROGram?
```

10.6.2 MEASURE SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition and The Data format <n> represents channel number.

Command	Description	SCPI
MEASure<n>:CURRent?	Returns the floating point value of the DC output current in amps.	C
MEASure<n>:CURRent:AVERage <NR1>	Enter a value of 3 to 9 to set the number of readings to average together when returning the current value from the MEAS<n>:CURR? command to reduce noise in the readback readings. The value of 3 (factory default) provides the fastest response time in the readings, but less rejection of noise.	N
MEASure<n>:CURRent:AVERage ?	Returns the number 3 to 9 to indicate the number of readings to average together when taking a current reading.	N
MEASure<n>:CURRent:PROGram ?	Returns the programmed output current from external Analog current programming feature.	N

Command	Description	SCPI
MEASure<n>:CURRent:ANALOG?	Returns the Co-efficient of programmed output current from external Analog current programming feature.	N
MEASure<n>:POWer?	Returns the floating point value of the measured output power in watts.	C
MEASure<n>:VOLTage?	Returns the floating point value of the DC output voltage in volts.	C
MEASure<n>:VOLTage:AVERage<NR1>	Enter a value of 1 to 5 to set the number of readings to average together when returning the voltage value from the MEAS<n>:VOLT? command. This function reduces noise in the readback readings. The value of 1 (factory default) provides the fastest response time in the readings, but less rejection of noise.	N
MEASure<n>:VOLTage:AVERage?	Returns the number 1 to 5 to indicate the last set number of readings to average together when taking a voltage reading.	N
MEASure<n>:VOLTage:PROGram?	Returns the programmed output voltage from external Analog voltage programming feature.	N
MEASure<n>:VOLTage:ANALOG?	Returns the Co-efficient of programmed output voltage from external Analog voltage programming feature.	N

10.7 OUTPUT SCPI Command Subsystem

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

10.7.1 OUTPUT SCPI Command Summary

```

OUTPut<n>
  :PROTection
    :DELay <NRf>
    :DELay?
    :FOLD <0|1|2|3>
    :FOLD?
    :CLE
  :SENSe <boolean>
  :SENSe?
    :DEFault <Boolean>
    :DEFault?
  :STATe <boolean>
  :STATe?
  :TRIPped?
  :REMote
    :INHIBIT<0|1>
    :INHIBIT?

```

10.7.2 OUTPUT SCPI Command Reference

The letter “C” in the “SCPI” column means that the command syntax is SCPI compliant; an “N” in the “SCPI” column means that the command syntax is not part of the SCPI definition and The Data format <n> represents individual channel number.

Command	Description	SCPI
OUTPut<n>: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<n>:PROTection:DELaY?	Returns the time delay to be executed by the supply.	N
OUTPut<n>: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 constant-voltage mode or constant power mode), or 2 (program down to zero upon entering constant-current mode or constant power mode), or 3 (program down to zero upon entering constant - current mode or constant-voltage mode),	N
OUTPut<n>: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 = will program down to zero upon entering constant - current mode or constant-voltage mode.	N
OUTPut<n>:PROTection:CLE	Clears the LATCH condition, when Remote Inhibit is set to Latched condition.	N
OUTPut<n>: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<n>:SENSe?	Returns the setting of the output voltage sense signal.	N
OUTPut<n>: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<n>:SENSe:DEFAult?	Returns the Power ON setting of the output voltage sense signal.	N

Command	Description	SCPI
OUTPut<n>:STATe <boolean>	Sets the output to zero or the programmed value; enable or disable the output Valid arguments are 1/ON or 0/OFF. *RST state value is ON.	C
OUTPut<n>:STATe?	Returns the state of the output: 1 = ON 0 = OFF	C
OUTPut<n>:TRIPped?	Returns the integer value 1 (TRIPPED) or 0 (UNTRIPPED) state of the output.	N
OUTPut<n>:REMote:INHIBIT <0 1>	The Remote Inhibit will work in LATCH and LIVE mode. LATCH: An open contact between remote inhibit terminal and return latches the output in the protection shutdown state; this state could be cleared either through the front panel fault screen or by sending the SCPI command OUTP<n>:PROT:CLE. LIVE: The output state follows the state of the Remote Inhibit input. An open contact between the remote inhibit terminal and return will turns the output off; a contact closer between remote inhibit terminal and return turns the output on.	N
OUTPut<n>:REMote:INHIBIT?	Returns the state of the Inhibit: 1 = LATCH 0 = LIVE	N

10.8 SYSTEM SCPI Command Subsystem

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

10.8.1 SYSTEM SCPI Command Summary

```

SYSTem
  :ERRor?      :PROGDELAY
  :REFrence <1|2|3>
  :REFrence?
  :ENABLE <1> <2> <3>
  :ENABLE?
  :DISABLE
  :DELAY <NRf>,<NRf>,<NRf>
  :DELAY?
  :OUTP
  :STAT <boolean>
  :STAT?
: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?

```

10.8.2 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
SYSTem:PROGDELAY:REFrence <1 2 3>	Sets the Reference channel of output ON or OFF delay. Valid arguments are : 1 (sets channel 1 as reference channel for output ON or OFF delay), or 2 (sets channel 2 as reference channel for output ON or OFF delay), or 3 (sets channel 3 as reference channel for output ON or OFF delay).	N
SYSTem:PROGDELAY:REFerence?	Returns the programed reference channel for output ON or OFF delay.	N
SYSTem:PROGDELAY:ENABLE <1> <2> <3>	Configures the channels to turn ON or OFF the output, on Output ON or OFF Delay Feature. Including Refence channel, Minimum 2 channels need to be Enabled.	N
SYSTem:PROGDELAY:ENABLE?	Returns the configured channels that are Enabled on Output ON or OFF Delay feature.	N

Command	Description	SCPI
SYSTem:PROGDELAY:DISABLE	Disables the configurations to turn ON or OFF the output on Output ON or OFF Delay feature.	N
SYSTem:PROGDELAY:DELAY <Nrf>, <Nrf>, <Nrf>	Sets the channels delay, Argument 1 represents channel 1 delay, argument 2 represents channel 2 delay and argument 3 represents channel 3 delay. Delay should be entered for all 3 channels, For example : Channel 1,2 are enabled, Delay should be 0,<delay for 2 channel>,0 SCPI: SYST:PROGDELAY:DELAY 0,30,0 Reference Channel and non-selected channel should have 0 delay.	N
SYSTem:PROGDELAY:DELAY?	Returns the programed delay for the channel that were enabled.	N
SYSTem:PROGDELAY:OUTPut:STATE <boolean>	Sets the Output ON or OFF Delay feature to turn ON or OFF with programed delay time. 0 = sets the Output ON or OFF Delay feature to turn OFF the output of power supply for the Enabled channels with programmed delay time. 1 = sets the Output ON or OFF Delay feature to turn ON the output of power supply for the Enabled channels with programmed delay time.	N
SYSTem:PROGDELAY:OUTPut:STATE?	Returns the Output ON or Off delay feature's state.	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

Command	Description	SCPI
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
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. 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

Command	Description	SCPI
SYST:NET:TERM?	Returns the string terminators to be used by the device.	N

10.9 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.

10.9.1 HTRIGGER SCPI Command Summary

```
HTRIGger<n>
:ABORt
```

10.9.2 HTRIGGER SCPI Command Reference

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

10.10 TRIGGER SCPI Command Subsystem

This section describes the programming soft trigger function.

10.10.1 TRIGGER SCPI Command Summary

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

10.10.2 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<n>:ABORt	Stops the execution of a currently running trigger function and clears all settings of voltage and current.	N
TRIGger<n>:RAMP	Executes voltage or current ramping function previously programmed by the SOURce command, i.e., SOURce<n>:VOLTage:RAMP:TRIG SOURce<n>:CURRent:RAMP:TRIG	N

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

10.11 CALIBRATION SCPI Command Subsystem

Note: See section Calibration for calibration procedures.



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.

10.11.1 CALIBRATION SCPI Command Summary

```

CALibrate<n>
  :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?
  :REMote
    :INHIBIT<0|1>
    :INHIBIT?
  :ANALOG
    :REMote
      :OUTPut <boolean>

```

```
CALibrate<n>
  :INITial
    :POWer <NRf>
    :POWer?
      :LIMit <NRf>
      :LIMit?
      :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<n>
  :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<n>
  :OUTPut
    :CURRent
      :CALCulate
      :DAC <NR1>
      :FIVEPOINT <1|2|3|4|5> <NRf>
      :FIVEPOINT?
      :GAIN <NRf>
      :GAIN?
      :OFFSet <NRf>
      :OFFSet?
      :EXTV
        :OFFSet <NRf>
        :OFFSet?
        :FSC <NRf>
        :FSC?
      :EXTI
        :OFFSet <NRf>
        :OFFSet?
        :FSC <NRf>
        :FSC?
      :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>
        :OFFSet?
        :FSC <NRf>
        :FSC?
      :EXTI
        :OFFSet <NRf>
        :OFFSet?
        :FSC <NRf>
        :FSC?
      :PROTection
        :DAC <NR1>
      :MON
        :OFSET <NRf>
        :FSC <NRf>
```

```

CALibrate<n>
  :UNLock <string>
  :STORe
  :LOCK

```

10.11.2 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<n>:INITial:CURRent <NRf>	Sets the power-on default value of current.	N
CALibrate<n>:INITial:CURRent?	Returns the default value of power-on current.	N
CALibrate<n>:INITial:CURRent:LIMit <NRf>	Sets the power-on default value of current limit.	N
CALibrate<n>:INITial:CURRent:LIMit?	Returns the default value of power-on current limit.	N
CALibrate<n>:INITial:CURRent:PROGram <string>	Sets the power-on default setting for output current programming. Valid arguments - INT, EXT or ADD	N
CALibrate<n>:INITial:CURRent:PROGram:FSC <NRf>	Sets the power-on default voltage value for full scale current from external analog programming.	N
CALibrate<n>:INITial:CURRent:PROGram:FSC?	Returns the power-on default full scale voltage value.	N
CALibrate<n>:INITial:CURRent:PROGram:FSCR <NRf>	Sets the power-on default resistance value for full scale current from external analog programming.	N
CALibrate<n>:INITial:CURRent:PROGram:FSCR?	Returns the power-on default full scale resistance value.	N
CALibrate<n>:INITial:CURRent:PROGram?	Returns the default setting of power-on output current programming.	N
CALibrate<n>:INITial:CURRent:MONitor	Initializes the current monitor (IMON) signal calibration	N
CALibrate<n>:INITial:CURRent:MONitor:FSC <NRf>	Sets the power-on default voltage on IMON signal for full scale output current.	N
CALibrate<n>:INITial:CURRent:MONitor:FSC?	Returns the power-on default full scale voltage value on IMON signal.	N
CALibrate<n>:INITial:VOLTagE <NRf>	Sets the power-on default voltage.	N
CALibrate<n>:INITial:VOLTagE?	Returns the default value of power-on voltage	N
CALibrate<n>:INITial:VOLTagE[:AMPLitude] <NRf>	Sets the power-on default voltage.	N
CALibrate<n>:INITial:VOLTagE[:AMPLitude]?	Returns the power-on default voltage.	N

Command	Description	SCPI
CALibrate<n>:INITial:VOLTage:PROTect ion <NRf>	Sets the power-on default value of the overvoltage protection.	N
CALibrate<n>:INITial:VOLTage:PROTect ion?	Returns the default value of the power-on overvoltage protection.	N
CALibrate<n>:INITial:VOLTage:LIMit <NRf>	Sets the power-on default value of voltage limit.	N
CALibrate<n>:INITial:VOLTage:LIMit?	Returns the default value of power-on current limit.	N
CALibrate<n>:INITial:VOLTage:PROGram <string>	Sets the power-on default setting for output voltage programming. Valid arguments - INT, EXT or ADD	N
CALibrate<n>:INITial:VOLTage:PROGram ?	Returns the default setting of power-on output voltage programming.	N
CALibrate<n>:INITial:VOLTage:PROGram :SOURce <string>	Sets the power-on default source for external analog programming for Voltage and Current. Valid arguments- VOLT and CURR	N
CALibrate<n>:INITial:VOLTage:PROGram :SOURce?	Returns the default power-on source for external analog programming.	N
CALibrate<n>:INITial:VOLTage:PROGram :FSC <NRf>	Sets the power-on default voltage value for full scale voltage from external analog programming.	N
CALibrate<n>:INITial:VOLTage:PROGram :FSC?	Returns the power-on default full scale voltage value.	N
CALibrate<n>:INITial:VOLTage:PROGram :FSCR <NRf>	Sets the power-on default resistance value for full scale voltage from external analog programming.	N
CALibrate<n>:INITial:VOLTage:PROGram :FSCR?	Returns the power-on default full scale resistance value.	N
CALibrate<n>:INITial:VOLTage:MONitor	Initializes the voltage monitor (VMON) signal calibration	N
CALibrate<n>:INITial:VOLTage:MONitor :FSC <NRf>	Sets the power-on default voltage on VMON signal for full scale output voltage.	N
CALibrate<n>:INITial:VOLTage:MONitor :FSC?	Returns the power-on default full scale voltage value on VMON signal.	N
CALibrate<n>:INITial:POWer <NRf>	Sets the power-on default value of power.	N
CALibrate<n>:INITial:POWer?	Returns the default value of power-on power.	N
CALibrate<n>:INITial:POWer:LIMit <NRf>	Sets the power-on default value of power limit.	N
CALibrate<n>:INITial:POWer:LIMit?	Returns the default value of power-on power limit.	N

Command	Description	SCPI
CALibrate<n>:INITial:MEASure:CURRent:AVErage <NR1>	Sets the number of readings to average together when returning the current value with the MEAS<n>: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<n>: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<n>:INITial:MEASure:VOLTage:AVErage <NR1>	Sets the number of readings to average together when returning the voltage value with the MEAS<n>: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
CALibrate<n>: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<n>:INITial:OUTPut:PROTecti on:FOLD <NR1>	Sets the power-on default foldback protection setting. Valid arguments are same as for OUTP<n>:PROT:FOLD	N
CALibrate<n>:INITial:OUTPut:PROTecti on:FOLD?	Returns the power-on default setting of foldback protection	N
CALibrate<n>:INITial:REMote:INHIBIT <0 1>	Sets the power-on default Remote Inhibit Mode.	N
CALibrate<n>:INITial:REMote:INHIBIT?	Returns the power-on default Remote Inhibit Mode settings.	N
CALibration<n>:INITial:ANALOG:REMote:OUTPut <boolean>	Sets the power-on state of the Remote output ON/OFF feature, either enables or disables the feature. 0 = Disables the Remote output ON/OFF feature. 1 = Enables the Remote output ON/OFF feature	N
CALibration<n>:INITial:ANALOG:REMote:OUTPut?	Returns the default state of power-on Remote ON/OFF feature	N
CALibrate<n>:MEASure:CURRent:CALCula te	Calculates the value of the gain and offset for current measurements.	N
CALibrate<n>:MEASure:CURRent:GAIN <NRf>	Sets the value of the gain for current measurements.	N
CALibrate<n>:MEASure:CURRent:GAIN?	Returns the value of the gain for current measurements.	N

Command	Description	SCPI
CALibrate<n>:MEASure:CURRent:OFFSet<NRf>	Sets the value of the offset for current measurements.	N
CALibrate<n>:MEASure:CURRent:OFFSet?	Returns the value of the offset for current measurements.	N
CALibrate<n>:MEASure:VOLTage:CALCulate	Calculates the value of the gain and offset for voltage measurements.	N
CALibrate<n>:MEASure:VOLTage:GAIN<NRf>	Sets the value of the gain for voltage measurements.	N
CALibrate<n>:MEASure:VOLTage:GAIN?	Returns the value of the gain for voltage measurements.	N
CALibrate<n>:MEASure:VOLTage:OFFSet<NRf>	Sets the value of the offset for voltage measurements.	N
CALibrate<n>:MEASure:VOLTage:OFFSet?	Returns the value of the offset for the voltage measurements.	N
CALibrate<n>:MODEl:LASTCALDATE<string>	Sets the date last calibrated; format: MM DD YYYY (space after MM and DD required)	N
CALibrate<n>:MODEl:LASTCALDATE?	Returns the date last calibrated.	N
CALibrate<n>:MODEl:NEXTCALDATE<string>	Sets the date next calibration is required; format: MM DD YYYY (space after MM and DD required)	N
CALibrate<n>:MODEl:NEXTCALDATE?	Returns the date next calibration is required.	N
CALibrate<n>:MODEl:POWERON <string>	Sets the default output enable condition at power on. Input format: ON = enables the output at next power on. OFF = disables the output at next power on .	N
CALibrate<n>:MODEl:POWERON?	Returns the status of the output enable condition at power on	N
CALibrate<n>: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<n>:MODEl:RESET?	Returns the status of the default output enable condition when the *RST command is issued	N
CALibrate<n>:OUTPut:CURRent:CALCulate	Calculates the value of the gain and offset for output current.	N
CALibrate<n>:OUTPut:CURRent:DAC<NR1>	Sets the output of the output current D/A converter.	N
CALibrate<n>:OUTPut:CURRent:FIVEPOINT <1 2 3 4 5> <NRf>	Sets output current value for each calibration point (1-5)	N
CALibrate<n>:OUTPut:CURRent:FIVEPOINT?	Returns the entered values for 5-point calibration.	N

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

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

10.12.1 STATUS SCPI Command Summary

```

STATus<n>
  :PROTection
    :CONDition?
    :ENABle <NR1>
    :ENABle?
    :EVENT?
  :FAULT?
  :FAULT<n>?

```

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<n>:PROTEction	Status Protection subsystem.	C
STATus<n>: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<n>: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<n>:PROTEction:ENABle?	Returns the value of the current mask of the Protection Event Register.	C
STATus<n>:PROTEction:EVENT?	Returns the integer value of the Protection Event Register.	C
System:FAULT?	Returns the Hexa Decimal value of fault state that occurs in the power supply. See section 10.2.1 for a detailed table of fault status.	N
System<n>:FAULT?	Returns the Hexa Decimal value of fault state that occurs in individual output channels. See section to 10.2.1 for a detailed table of the fault status.	N

10.13 Examples of Using the SCPI Commands

The following examples demonstrate programming a power supply to control all three channels and to readback the output using the SCPI commands. The maximum voltage and current output are dependent on the particular model channel rating. 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 in channel 1, 10 VDC @ 5A In channel 2 and 15 VDC @ 7A and verify the output of all three channels.

```
// 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.
SOUR1:CURREN 1.0 // program output current of channel 1 to 1.0 A.
SOUR1:CURREN? // confirm the output current setting of channel1 (response: 1.0).
SOUR1:VOLT 5.0 // program output voltage of channel 1 to 5.0 VDC.
SOUR1:VOLT? // confirm the output voltage setting of channel 1 (response: 5.0).
OUTP1:STAT 1 // program the output state of channel 1 enabled (1).
OUTP1:STAT? // confirm the output state of channel 1 (response: 1).
MEAS1:CURREN? // measure the actual output current of channel 1 (response: ~ 0.0 with no
load on output).
MEAS1:VOLT? // measure the actual output voltage of channel 1 (response: ~ 5.0).

SOUR2:CURREN 5.0 // program output current of channel 2 to 5.0 A.
SOUR2:CURREN? // confirm the output current setting of channel 2 (response: 5.0).
```

```

SOUR2:VOLT 10.0 // program output voltage of channel 2 to 10.0 VDC.
SOUR2:VOLT? // confirm the output voltage setting of channel 2 (response: 10.0).
OUTP2:STAT 1 // program the output state of channel 2 enabled (1).
OUTP2:STAT? // confirm the output state of channel 2 (response: 1).
MEAS2:CURR? // measure the actual output current of channel 2 (response: ~ 0.0 with no
load on output).
MEAS2:VOLT? // measure the actual output voltage of channel 2 (response: ~ 10.0).

SOUR3:CURR 7.0 // program output current of channel 3 to 7.0 A.
SOUR3:CURR? // confirm the output current setting of channel 3 (response: 7.0).
SOUR3:VOLT 15.0 // program output voltage of channel 3 to 15.0 VDC.
SOUR3:VOLT? // confirm the output voltage setting of channel 3 (response: 15.0).
OUTP3:STAT 1 // program the output state of channel 3 enabled (1).
OUTP3:STAT? // confirm the output state of channel 3 (response: 1).
MEAS3:CURR? // measure the actual output current of channel 3 (response: ~ 0.0 with no
load on output).
MEAS3:VOLT? // measure the actual output voltage of channel 3 (response: ~ 15.0).

```

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<n>: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.
SOUR1:VOLT:PROT 4.0 // program the channel 1 OVP trip point to 4.0 VDC.
SOUR1:VOLT:PROT? // confirm the OVP trip point setting (response: 4.0).
SOUR1:CURR 1.0 // program channel 1 output current to 1.0 A.
SOUR1:VOLT 3.0 // program channel 1 output voltage to 3.0 VDC.
CAL1:UNLOCK "6867" // Unlock overvoltage simulation of channel 1.
*SRE 2 // enable the GPIB service request upon a fault.
*SRE? // confirm the GPIB service request enabled (response 2).
STAT1:PROT:EVENT? // confirm no faults occurred for channel 1 (response: 0).
// confirm that the fault led is not active.
SOUR1:VOLT 4.0 // program channel 1 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 of channel 2 to change its output voltage and current to 5 VDC @ 1A at the same time.

```
// Use SYST<n>: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.
SOUR2:CURR:TRIG 1.0 // program channel 2 output current to 1.0 A upon trigger.
SOUR2:CURR:TRIG? // confirm output current set to 1.0 A upon trigger.
SOUR2:VOLT:TRIG 5.0 // program channel 2 output voltage to 5.0 VDC upon trigger
SOUR2:VOLT:TRIG? // confirm output current set to 5.0 VDC upon trigger.
MEAS2:CURR? // measure the actual output current of channel 2 (response: 0.0).
MEAS2:VOLT? // measure the actual output voltage of channel 2 (response: 0.0).
TRIG2:TYPE 3 // trigger the channel 2 to implement curr and volt programming.
MEAS2:CURR? // measure the actual output current of channel 2 (response: ~ 0. 0
with no load on output).
MEAS2:VOLT? // measure the actual output voltage channel 2 (response: ~ 5.0).
TRIG2:ABORT // turn off channel 2 trigger mode.
```

10.13.4 Ramp I Example

Program a unit with the output shorted of channel 3 to ramp its output current from 5A to 25A in 30 seconds.

```
// Use SYST<n>: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 of channel 3.
SOUR3:VOLT 33.0 // program channel 3 output voltage to 33.0 VDC.
SOUR3:CURR 5.0 // program channel 3 output current to 5.0 A.
SOUR3: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 the trigger command.
TRIG3:RAMP // start ramp execution of channel 3.
TRIG3:ABORT // turn off channel 3 trigger mode .
```

10.13.5 Ramp V Example 2

Program a unit with no load at the output of channel 3, 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.
SOUR3:CURREN 33.0 // program channel 3 output current to 33.0 A.
SOUR3:VOLT 5.0 // program channel 3 output voltage to 5.0 VDC.
SOUR3:VOLT:RAMP:TRIG 25.0 30.0 // program channel 3 voltage to ramp from the present
// value (5.0 VDC) to 25.0 VDC in 30 secs.
// upon the trigger command.
TRIG3:RAMP // start ramp execution of channel 3.
TRIG3:ABORT // turn off channel 3 trigger mode.
```

10.13.6 Power On INIT Example

Program a unit to power-on with initialize 2 VDC @ 1A with an overvoltage protection level of 3 VDC for channel1, initialize 5 VDC @ 2A with an overvoltage protection level of 6 VDC for channel 2 and initialize 10 VDC @ 5A with an Overvoltage protection level of 11 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.
CAL1:INIT:CURREN 1.0 // set power-on initial current of channel 1 to 1.0 A.
CAL1:INIT:CURREN? // confirm power-on initial current setting.
CAL1:INIT:VOLT 2.0 // set power-on initial voltage of channel 1 to 2.0 VDC.
CAL1:INIT:VOLT? // confirm power-on initial voltage setting.
CAL1:INIT:VOLT:PROT 3.0 // set power-on initial overvoltage protection of channel 1 to 3.0
//VDC.
CAL1:INIT:VOLT:PROT? // confirm power-on initial overvoltage protection setting.

CAL2:INIT:CURREN 2.0 // set power-on initial current of channel 2 to 2.0 A.
CAL2:INIT:CURREN? // confirm power-on initial current setting.
CAL2:INIT:VOLT 5.0 // set power-on initial voltage of channel 2 to 5.0 VDC.
CAL2:INIT:VOLT? // confirm power-on initial voltage setting.
CAL2:INIT:VOLT:PROT 6.0 // set power-on initial overvoltage protection of channel 2 to 6.0
//VDC.
CAL2:INIT:VOLT:PROT? // confirm power-on initial overvoltage protection setting.
```

```
CAL3:INIT:CURR 5.0 // set power-on initial current of channel 3 to 5.0 A.
CAL3:INIT:CURR? // confirm power-on initial current setting.
CAL3:INIT:VOLT 10.0 // set power-on initial voltage of channel 3 to 10.0 VDC.
CAL3:INIT:VOLT? // confirm power-on initial voltage setting.
CAL3:INIT:VOLT:PROT 11.0 // set power-on initial overvoltage protection of channel 3 to 11.0
//VDC.
CAL3: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.
SOUR1:CURR? // confirm power-on initial current setting of channel 1.
SOUR1:VOLT? // confirm power-on initial voltage setting of channel 1.
SOUR1:VOLT:PROT? // confirm power-on initial overvoltage protection setting of
//channel 1.
SOUR2:CURR? // confirm power-on initial current setting of channel 2.
SOUR2:VOLT? // confirm power-on initial voltage setting of channel 2.
SOUR2:VOLT:PROT? // confirm power-on initial overvoltage protection setting of
//channel 2.
SOUR3:CURR? // confirm power-on initial current setting of channel 3.
SOUR3:VOLT? // confirm power-on initial voltage setting of channel 3.
SOUR3:VOLT:PROT? // confirm power-on initial overvoltage protection setting of
//channel 3.
```

This page intentionally left blank.

OUTPUT ON/OFF DELAY

11.1 Introduction

This section describes the operation of the OUTPUT ON/OFF DELAY feature. This allows user to configure the output to turn ON or to turn OFF the output in a sequence with respect to one of the reference channels with a programmable delay time.

11.2 Applications under OUTPUT ON/OFF DELAY

Reference Channel:

Reference Channel allows the user to configure the supply to turn ON or to turn OFF the output of programmed channel as Output ON/OFF delay starting sequence.

Reference Channel of the Output ON/OFF Delay can be set by sending the following SCPI command using any of the interface available on the unit (RS232,USB,LAN,GPIB):

```
SYSTEM:PROGDELAY:REF <1|2|3>
```

Valid arguments are: 1 (sets channel 1 as reference channel for output on/off delay)

2 (sets channel 2 as reference channel for output on/off delay)

3 (sets channel 3 as reference channel for output on/off delay)

```
SYSTEM:PROGDELAY:REF? // Returns the programed reference channel for output ON or OFF delay
```

Enable Channel:

Enable channel allows the user to select the required channels that are to be turned on or off in the output ON/OFF delay sequence.

To enable the channels of the Output ON/OFF Delay can be set by sending the following SCPI command using any of the interface available on the unit (RS232,USB,LAN,GPIB):

```
SYSTEM:PROGDELAY:ENABLE <1> <2> <3>
```

A space should be passed between each argument.

Minimum of 2 channels must be enabled, inclusive of the reference channel to configure the channels to work on Output ON/OFF Delay feature.

For example: If the user wants to enable channel 1 and channel 3 in this Output ON/OFF feature user must send the SCPI command
(SYSTEM:PROGDELAY:ENABLE 1 3).

If the user wants to enable channel 1, channel 2 and channel 3 in this Output ON/OFF feature user must send the SCPI command
(SYSTEM:PROGDELAY:ENABLE 1 2 3).

SYSTEM:PROGDELAY:ENABLE? // Returns the configured channels that are Enabled on Output ON or OFF Delay feature.

Disable Channel:

Disable channel allows the user to disable the configurations to turn ON or OFF the output on Output ON or OFF Delay feature. i.e., it Disable all the channels which are enabled.

To disable the channels of the Output ON/OFF Delay can be set by sending the following SCPI command using any of the interface available on the unit (RS232,USB,LAN,GPIB):

SYSTEM:PROGDELAY:DISABLE

Channel Delay:

The channel delay allows the user to set the channel's delay to turn on or off the output of the enabled channels with a specified delay time. Reference channel's delay and the non-selected channel should be programmed to 0.

To program the delay for the enabled channels, it can be set by sending the following SCPI command using any of the interface available on the unit (RS232,USB,LAN,GPIB):

SYSTEM:PROGDELAY:DELAY <NRf>,<NRf>,<NRf>

For example: Reference channel is 1 and channels 1 and 2 are enabled,

SCPI: **SYSTEM:PROGDELAY:DELAY 0,30,0**

Arguments should be (0 (channel 1) , <delay for 2 channel> , 0 (channel 3))

“Comma (,)” should be passed after each argument.

SYSTEM:PROGDELAY:DELAY? // Returns the programmed delay for the channel that were enabled.

Output ON/OFF State:

Output ON/OFF State allows the user to either turn ON or OFF the outputs of the enabled channels with the delays specified. State 0 (OFF) allows the user to turn OFF the channels output with programmed delay time. State 1 (ON) allows the user to turn ON the channels output with programmed delay time.

To program the output state for the enabled channels, it can be set by sending the following SCPI command using any of the interface available on the unit (RS232, USB, LAN and GPIB):

SYSTEM:PROGDELAY:OUTPut:STATe <boolean>

SYSTem:PROGDELAY:OUTPut:STATe? // Returns the Output ON or Off delay feature's state.

Example 1: To turn ON the output of channel 1,2 and 3 in a sequence with configuration as follows:

Reference channel = 1, Enabled channel = 1,2 and 3, Delay of channel 2 = 300 ms, Delay of channel 3 = 1000 ms and output ON/OFF state = 1 (ON).

```
SYST:PROGDELAY:REF 1 //sets the reference channel as channel 1.
SYST:PROGDELAY:ENABLE 1 2 3 //enables all three channels.
SYST:PROGDELAY:OUTP:STAT 1 // Allows the selected channels to turn ON the
output in a sequence with programmable delay.

SYST:PROGDELAY:DELAY 0,300,1000 // Since channel 1 is the reference channel.
Reference channel must be programmed to 0 ms,
sets channel 2 delay to 300 ms and sets channel 3
delay to 1000 ms.

OUTP1:STAT 1 //Since channel 1 is set as reference channel, it
should be turned ON to initiate the sequence of
Output ON/OFF Delay.
```

Example 2: To turn OFF the output of channel 1,2 and 3 in a sequence with configuration as follows:

Reference channel = 3, Enabled channel = 2 and 3, Delay of channel 2 = 500 ms and output ON/OFF state = 0 (OFF).

```
SYST:PROGDELAY:REF 3 //sets the reference channel as channel 3
SYST:PROGDELAY:ENABLE 2 3 //enables only channels 2 and 3.
SYST:PROGDELAY:OUTP:STAT 0 // Allows the selected channels to turn OFF the
output in a sequence with programable delay.

SYST:PROGDELAY:DELAY 0,500,0 // Since channel 3 is the reference channel.
Reference channel must be programmed to 0 ms,
sets channel 2 delay to 500 ms and the unselected
channel i.e., channel 1 must also be programmed to
0 ms.

OUTP1:STAT 0 //Since channel 3 is set as reference channel, it
should be turned OFF to initiate the sequence of
Output ON/OFF Delay.
```

Example 3: To turn ON the output of channel 1 and 2 in a sequence with configuration as follows:

Reference channel = 1, Enabled channel = 1 and 3, Delay of channel 2 = 300 ms, and output ON/OFF state = 1 (ON).

```
SYST:PROGDELAY:REF 1 //sets the reference channel as channel 1.
SYST:PROGDELAY:ENABLE 1 2 //enables channel 2.
SYST:PROGDELAY:OUTP:STAT 1 // Allows the selected channels to turn ON the
                             output in a sequence with programable delay.

SYST:PROGDELAY:DELAY 0,300 // Since channel 1 is the reference channel.
                             Reference channel must be programmed to 0 ms,
                             sets channel 2 delay to 300 ms.

OUTP1:STAT 1 //Since channel 1 is set as reference channel, it
              should be turned ON to initiate the sequence of
              Output ON/OFF Delay.
```

This page intentionally left blank.

SEQUENCING FUNCTION

12.1 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 or Ethernet, 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.

12.2 Program SCPI Command Subsystem

This section describes the operation of the Digital Interfaces GPIB, USB, RS232 and Ethernet 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 individual Channels.

12.2.1 Program SCPI Command Summary

```

PROGram<n>
  :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?

```

12.2.2 Program SCPI Command Reference

The letter "C" in the "SCPI" column means that the command syntax is SCPI compliant; an "N" in the "SCPI" column means that the command syntax is not part of the SCPI definition and The Data format <n> represents individual channel number.

Command	Description	SCP I
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

<p>PROGram:DEFine <step#>, REPEAT</p>	<p>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).</p>	<p>C</p>
<p>PROGram:DEFine <step#>, SUBCALL, "name"</p>	<p>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.</p>	<p>C</p>
<p>PROGram:DEFine <step#>, RETURN</p>	<p>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.</p>	<p>C</p>
<p>PROGram:DEFine <step#>, LOOP, <count></p>	<p>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.</p>	<p>C</p>
<p>PROGram:DEFine <step#>, NEXT</p>	<p>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.</p>	<p>C</p>
<p>PROGram:DEFine <step#>, STOP</p>	<p>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.</p>	<p>C</p>

PROG: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
PROG: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
PROG: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
PROG:DELete	Program Delete subsystem	C
PROG: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
PROG:DELete:ALL	This command causes all defined sequences to be deleted from ram and non-volatile memory.	C
PROG:SAVe	The sub tree for the SAVe commands.	N
PROG: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
PROG:SAVe:ALL	Saves all defines sequences to non-volatile memory.	N
PROG: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
PROG:EXECute VIMODE, <volts>, <amps>, <ovp>	Allows setting of active voltage, current, and ovp.	C
PROG: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
PROG: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
PROG: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

<p>PROG<code>ram</code>:MALLocate DEFAULT</p>	<p>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.</p>	<p>C</p>																																																
<p>PROG<code>ram</code>:NAME "sequence name"</p>	<p>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 it 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.</p>	<p>C</p>																																																
<p>PROG<code>ram</code>:NAME?</p>	<p>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.</p>	<p>C</p>																																																
<p>PROG<code>ram</code>:STATe <state name></p>	<p>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:</p> <table border="1" data-bbox="638 1058 1349 1308"> <thead> <tr> <th>requested \ present</th> <th>RUN</th> <th>RESUME</th> <th>PAUSE</th> <th>STOP</th> <th>COMPL</th> </tr> </thead> <tbody> <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> </tbody> </table>	requested \ present	RUN	RESUME	PAUSE	STOP	COMPL	RUNNING	*error*	*error*	PAUSED	STOPPED	*error*	PAUSED	*error*	RUNNING	PAUSED	STOPPED	*error*	STOPPED	RUNNING	*error*	STOPPED	STOPPED	*error*	EDIT	*error*	*error*	*error*	*error*	STOPPED	EMPTY	*error*	*error*	*error*	*error*	*error*	ERROR	*error*	*error*	*error*	*error*	*error*	INITIALIZE	*error*	*error*	*error*	*error*	*error*	<p>N</p>
requested \ present	RUN	RESUME	PAUSE	STOP	COMPL																																													
RUNNING	*error*	*error*	PAUSED	STOPPED	*error*																																													
PAUSED	*error*	RUNNING	PAUSED	STOPPED	*error*																																													
STOPPED	RUNNING	*error*	STOPPED	STOPPED	*error*																																													
EDIT	*error*	*error*	*error*	*error*	STOPPED																																													
EMPTY	*error*	*error*	*error*	*error*	*error*																																													
ERROR	*error*	*error*	*error*	*error*	*error*																																													
INITIALIZE	*error*	*error*	*error*	*error*	*error*																																													
<p>PROG<code>ram</code>:STATe?</p>	<p>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"</p>	<p>N</p>																																																

12.2.3 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<n>:DEL:ALL`.

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

```
PROG<n>: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<n>:NAME "SEQ1"
```

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

```
PROG<n>:STAT?
```

Response: "EMPTY"

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

```
PROG<n>:MALL DEFAULT
```

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

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

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

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

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

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

PROG<n>: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<n>:STAT COMPLETE
```

```
PROG<n>:SAVE:SEL
```

```
PROG<n>: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:

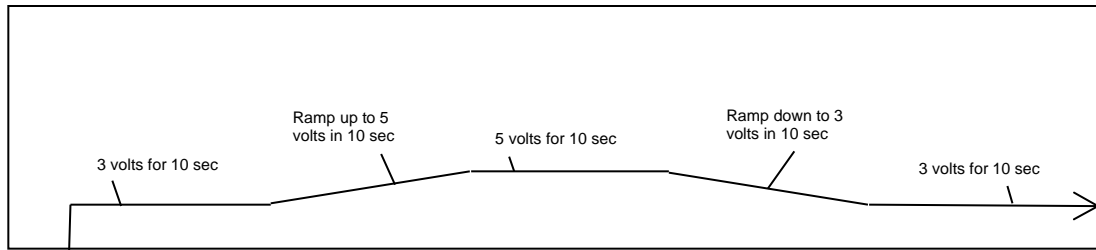


Figure 12-1: Power Supply Output for Example 1

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<n>:NAME "SEQ2"
```

```
PROG<n>:MALL DEFAULT
```

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

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

```
PROG<n>:DEF 3, RETURN
```

```
PROG<n>:STAT COMPLETE
```

```
PROG<n>:SAVE:SEL
```

```
PROG<n>: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<n>:STAT ON
```

```
PROG<n>:STATE  RUN
```

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

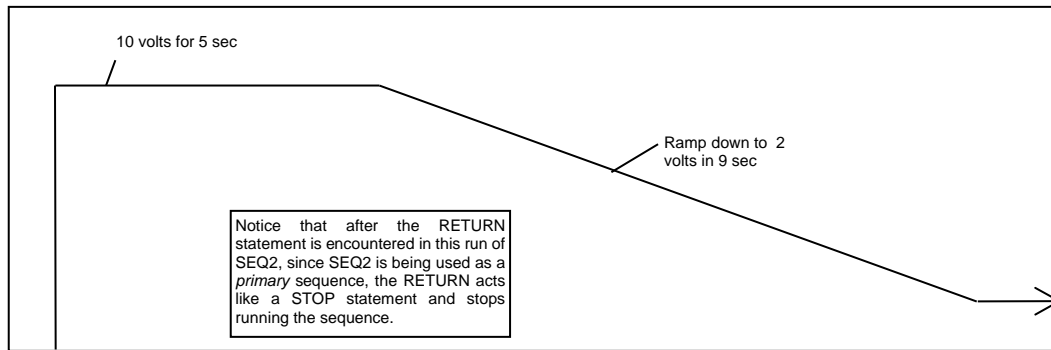


Figure 12-2: Power Supply Output for Example 2

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<n>:NAME "SEQ1"
PROG<n>:DEL:SEL
```

That deletes SEQ1.

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

```
PROG<n>:NAME "SEQ1"
PROG<n>:MALL DEFAULT
PROG<n>:DEF 1, VIMODE, 3, 4, 11, 10 //go to 3 volts, 4 amps, 11 volts ovp,
                                     for 10 sec
PROG<n>:DEF 2, RAMPTOV, 3, 5, 4, 11, 10 //ramp from 3 volts to 5 volts in
                                     10 sec
PROG<n>:DEF 3, VIMODE, 5, 4, 11, 10 //hold 5 volts for 10 sec
PROG<n>:DEF 4, RAMPTOV, 5, 3, 4, 11, 10 //ramp from 5 volts to 3 volts in
                                     10 sec
PROG<n>:DEF 5, VIMODE, 3, 4, 11, 10 //hold 3 volts for 10 sec
PROG<n>:DEF 6, SUBCALL, "SEQ2" //call SEQ2 as a subsequence
```

```

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

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

PROG<n>:STAT COMPLETE

PROG<n>:SAVE:SEL

PROG<n>:STAT?

```

Response: State: STOPPED

```

OUTP<n>:STAT ON

PROG<n>:STAT RUN

```

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

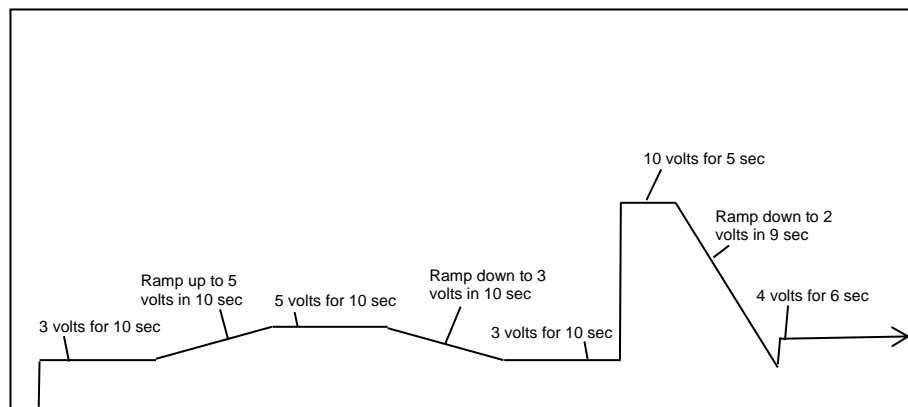


Figure 12-3: Power Supply Output for Example 3

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<n>:SEL:NAME "SEQ3"

PROG<n>:MALL DEFAULT

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

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

```

```

PROG<n>:DEF 3, VIMODE, 5, 4, 11, 10 //hold 5 volts for 10 sec
PROG<n>:DEF 4, RAMPTOV, 5, 3, 4, 11, 10 //ramp from 5 volts to 3 volts in
10 sec
PROG<n>:DEF 5, VIMODE, 3, 4, 11, 10 //hold 3 volts for 10 sec
PROG<n>:DEF 6, SUBCALL, "SEQ2" //call SEQ2 as a subsequence
PROG<n>:DEF 7, VIMODE, 4, 5, 11, 6 //go to 4 volts, 5 amps, 11 volts
ovp, for 6 sec

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

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

PROG<n>:STAT COMPLETE
PROG<n>:SAVE:SEL
PROG<n>:STAT?
    
```

Response: State: STOPPED

```

OUTP<n>:STAT ON
PROG<n>:STAT RUN
    
```

Following is a diagram for Example 4.

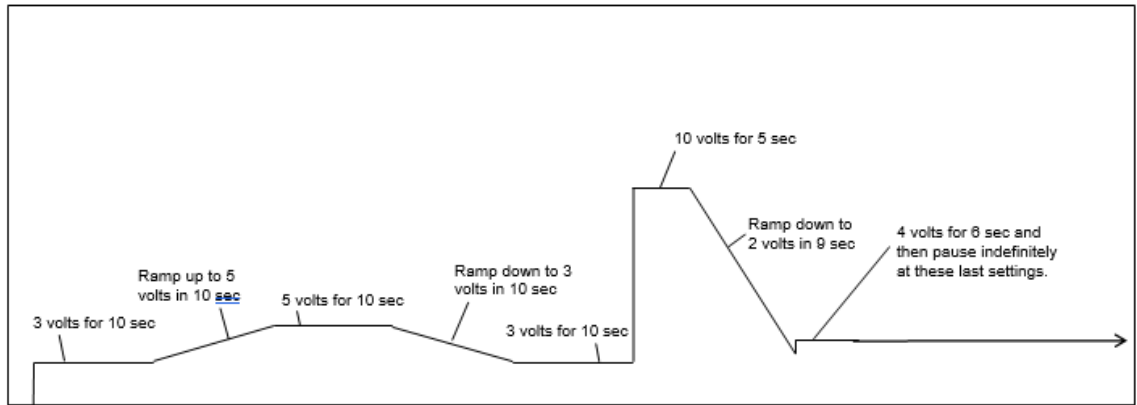


Figure 12-4: End-of-Sequence Pause for Example 4

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<n>:NAME "Charge" //This sequence will simulate an RC=1
                        charge curve

PROG<n>:MALL DEFAULT

PROG<n>:DEF 1, VIMODE, 0, 5, 20, 1 //go to 0 volts for 1 second
PROG<n>:DEF 2, RAMPTOV, 0, 0.95, 5, 20, 0.1 //ramp from 0 to 0.95 volts in 0.1 sec
PROG<n>:DEF 3, RAMPTOV, 0.95, 1.81, 5, 20, 0.1 //ramp from 0.95 volts to 1.81 volts in
                                                0.1sec
PROG<n>:DEF 4, RAMPTOV, 1.81, 2.59, 5, 20, 0.1 //ramp from 1.81 volts to 2.59 volts in
                                                0.1sec
PROG<n>:DEF 5, RAMPTOV, 2.59, 3.30, 5, 20, 0.1 //ramp from 2.59 volts to 3.30 volts in
PROG<n>:DEF 6, RAMPTOV, 3.30, 3.93, 5, 20, 0.1 //ramp from 3.30 volts to 3.93 volts in
                                                0.1sec
PROG<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>:DEF 21, GOTO, "Discharge" (step 21 may only be a STOP,
RETURN, or GOTO)

PROG<n>:DEF 22, STOP
PROG<n>:STAT COMPLETE
PROG<n>:SAVE:SEL
PROG<n>: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<n>:NAME "Discharge"
PROG<n>:MALL DEFAULT
PROG<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>: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<n>:DEF 21, RETURN //step 21 may only be a STOP,
                                         RETURN, or GOTO

```

```
PROG<n>:STAT COMPLETE
```

```
PROG<n>:SAVE:SEL
```

```
PROG<n>:STAT?
```

Response: State: STOPPED

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

```
PROG<n>:NAME "Charge"
```

```
OUTP<n>:STAT ON
```

```
PROG<n>:STAT  RUN
```

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

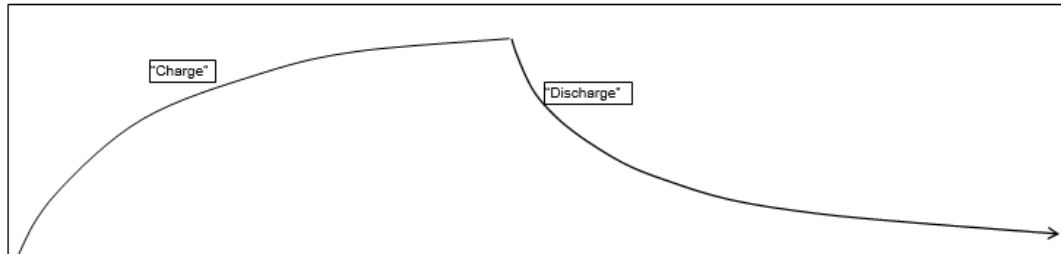


Figure 12-5: Power Supply Output Example 5

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:SEL
PROG<n>:STAT?
```

Response: State: STOPPED

```
OUTP<n>:STAT  ON
PROG<n>:STAT  RUN
```

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

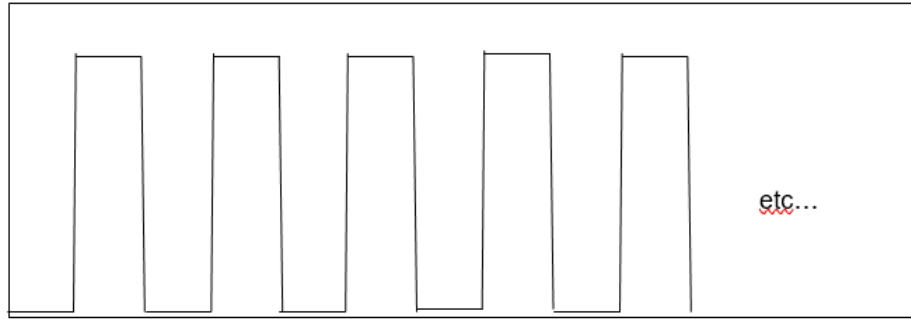


Figure 12-6: Power Supply Output Example 6

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:

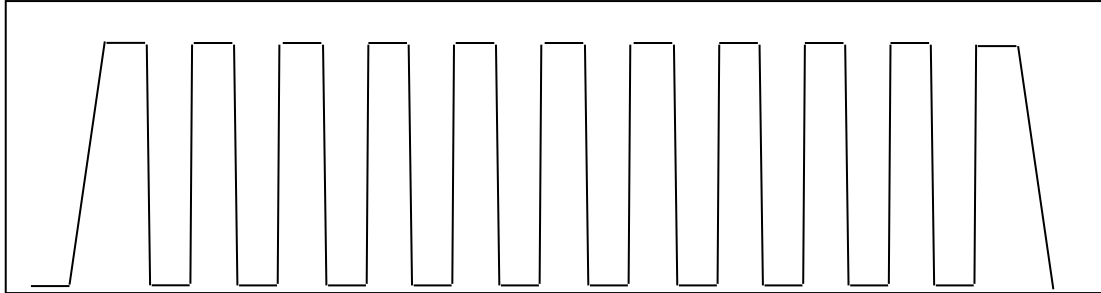


Figure 12-7: Power Supply Output for Example 7

This page intentionally left blank.

13 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.

13.1 Introduction

The Asterion DC Multioutput Series power supply is calibrated to adjust internal signal levels to correspond to the expected supply output signal levels. You must perform the calibration procedures if the power supply's programming or readback performance falls out of specification due to component aging drifts. Refer to your power supply operation manual to find the required calibration interval. The Asterion DC Multioutput Series 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 $25^{\circ}\text{C} \pm 5^{\circ}\text{C}$, 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 CAL1:MOD:LASTCALDATE and CAL1:MOD:NEXTCALDATE commands (Section 10.11) should be issued once calibration procedures have been performed.

13.2 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 5-3).

For **Ethernet:**

Refer to Section 6.4 for Ethernet communication setup. Connect the RJ-45 Ethernet cable to the rear panel of the UUT.

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.

*****Note:** The Data format <n> each channel number (1, 2 and 3) and each channel should be calibrated individually.

13.3 Voltage Programming Calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. Program the supply to unlock the non-volatile memory for calibration value storage:

```
CAL<n>:UNLOCK "6867"
```
4. Program the overvoltage protection to maximum to prevent nuisance trips:

```
CAL<n>:OUTP:VOLT:PROT:DAC 65535
```
5. Program the output current to full scale to prevent Constant-Current operation:

```
CAL<n>:OUTP:CURR:DAC 65535
```
6. Set the output voltage sensing to "Local"

```
OUTP<n>:SENS LOCAL
```
7. Set the output ON

```
OUTP<n>:STAT ON
```
8. Program the output of the first calibration point by sending the following command string from the computer:

```
CAL<n>:OUTP:VOLT:DAC 3275
```
9. Let the output settle and measure the voltage with a high precision voltmeter; this is value 1.
10. Enter the actual output voltage value of the first calibration point:

```
CAL<n>:OUTP:VOLT:FIVEPOINT 1 <value 1>
```
11. Program the output of the second calibration point by sending the following command string from the computer:

```
CAL<n>:OUTP:VOLT:DAC 19000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 32000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 45000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 62250
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:FIVEPOINT?
```

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

```
CAL<n>:OUTP:VOLT:CALC
```

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

```
CAL<n>: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<n>:LOCK
```

13.4 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 Output connector on the rear side of the power supply. Connect positive terminal to positive of Output connector and negative terminal to negative of Output 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 supply to unlock the non-volatile memory for calibration value storage:

```
CAL<n>:UNLOCK "6867"
```

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

```
CAL<n>:OUTP:VOLT:PROT:DAC 65535
```

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

```
CAL<n>:OUTP:CURR:DAC 65535
```

6. Set the output voltage sensing to "Remote"

```
OUTP<n>:SENS REMOTE
```

7. Set the output ON

```
OUTP<n>:STAT ON
```

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

```
CAL<n>:OUTP:VOLT:DAC 3275
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 19000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 32000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 45000
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:DAC 62250
```

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

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

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

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

```
CAL<n>:OUTP:VOLT:FIVEPOINT?
```

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

```
CAL<n>:OUTP:VOLT:CALC
```

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

```
CAL<n>: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<n>:LOCK

13.5 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.

$$R = (\text{FSV} * 80\%) / \text{FSC} * (\text{Present DAC value} / \text{MAX DAC value})$$
 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 supply to unlock the non-volatile memory for calibration value storage:


```
CAL<n>:UNLOCK "6867"
```
4. Program the overvoltage protection to maximum to prevent nuisance trips:


```
CAL<n>:OUTP:VOLT:PROT:DAC 65535 (65535 is the max DAC value).
```
5. Program the output voltage to full scale to prevent Constant-Voltage operation:


```
CAL<n>:OUTP:VOLT:DAC 65535
```
6. Program the output of the first calibration point by sending the following command string from the computer:


```
CAL<n>:OUTP:CURR:DAC 3275 (3275 is the present DAC value for point1)
```
7. Let the output settle and measure the output current with the current shunt and the high precision voltmeter; this is value 1.
8. Enter the actual output current of the first calibration point:


```
CAL<n>:OUTP:CURR:FIVEPOINT 1 <value 1>
```
9. Program the output of the second calibration point by sending the following command string from the computer:

CAL<n>:OUTP:CURR:DAC 19000 (*19000 is the present DAC value for point2*)

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

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

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

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

CAL<n>:OUTP:CURR:DAC 32000 (*32000 is the present DAC value for point3*)

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

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

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

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

CAL<n>:OUTP:CURR:DAC 45000 (*45000 is the present DAC value for point4*)

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<n>: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<n>:OUTP:CURR:DAC 62250 (*62250 is the present DAC value for point5*)

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<n>:OUTP:CURR:FIVEPOINT 5 <value 5>

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


```
CAL<n>:OUTP:CURR:FIVEPOINT?
```

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

```
CAL<n>:OUTP:CURR:CALC
```

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

```
CAL<n>:STORE
```

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

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

```
CAL<n>:LOCK
```

13.6 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 7.2

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. 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.

13.6.1 Voltage programming – 10V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.

4. To calibrate channel 2, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Voltage source and 10V range. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR VOLT
SOUR<n>:VOLT:PROG INT
SOUR<n>:VOLT:PROG:FSC 10
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:VOLT:EXTV:OFFS 0
```

9. To calibrate channel 1, Set the DC voltage to 10V +/-0.003V at PIN1 (VPRG1_VSOUR) of channel 1 Analog Programming connector. Calibrate the full scale point.
10. To calibrate channel 2, Set the DC voltage to 10V +/-0.003V at PIN1 (VPRG2_VSOUR) of channel 2 Analog Programming connector. Calibrate the full scale point.
11. To calibrate channel 3, Set the DC voltage to 10V +/-0.003V at PIN1 (VPRG3_VSOUR) of channel 3 Analog Programming connector. Calibrate the full scale point.

```
CAL<n>:OUTP:VOLT:EXTV:FSC <100% channel rating voltage>
```

12. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

13.6.2 Voltage programming – 5V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.

3. To calibrate channel 1, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero volt (0 +/- 0.003V) across PIN 1 (VPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Voltage source and 5V range. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR VOLT
SOUR<n>:VOLT:PROG INT
SOUR<n>:VOLT:PROG:FSC 5
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:VOLT:EXTV:OFFS 0
```

9. To calibrate channel 1, Set the DC voltage to 5V +/-0.003V at PIN1 (VPRG1_VSOUR) of channel 1 Analog Programming connector. Calibrate the full scale point.
10. To calibrate channel 2, Set the DC voltage to 5V +/-0.003V at PIN1 (VPRG2_VSOUR) of channel 2 Analog Programming connector. Calibrate the full scale point.
11. To calibrate channel 3, Set the DC voltage to 5V +/-0.003V at PIN1 (VPRG3_VSOUR) of channel 3 Analog Programming connector. Calibrate the full scale point.
12. CAL<n>:OUTP:VOLT:EXTV:FSC <100% channel rating voltage>

13. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

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

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero ohm across PIN 2 (VPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero ohm across PIN 2 (VPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero ohm across PIN 2 (VPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Connect zero ohm across PIN 3 and PIN 18 of Analog Programming connector.
7. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR CURR
SOUR<n>:VOLT:PROG INT
SOUR<n>:VOLT:PROG:FSCR 10
```

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

```
CAL<n>:UNLOCK "6867"
```

9. Calibrate the Offset point.

```
CAL<n>:OUTP:VOLT:EXTI:OFFS 0
```

10. To calibrate channel 1, Connect 10k ohm across PIN 2 (VPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
11. To calibrate channel 2, Connect 10k ohm across PIN 2 (VPRG2_ISOUR) (VPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
12. To calibrate channel 3, Connect 10k ohm across PIN 2 (VPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

```
CAL<n>:OUTP:VOLT:EXTI:FSCR <100% channel rating voltage>
```

13. Store the Calibration.

```
CAL<n>:STORE
```

CAL<n>:LOCK

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

STEP	DESCRIPTION
------	-------------

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero ohm across PIN 2 (VPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero ohm across PIN 2 (VPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero ohm across PIN 2 (VPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Connect zero ohm across PIN 3 and PIN 18 of Analog Programming connector.
7. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR CURR
SOUR<n>:VOLT:PROG INT
SOUR<n>:VOLT:PROG:FSCR 10
```

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

```
CAL<n>:UNLOCK "6867"
```

9. Calibrate the Offset point.

```
CAL<n>:OUTP:VOLT:EXTI:OFFS 0
```

10. To calibrate channel 1, Connect 5k ohm across PIN 2 (VPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
11. To calibrate channel 2, Connect 5k ohm across PIN 2 (VPRG2_ISOUR) (VPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
12. To calibrate channel 3, Connect 5k ohm across PIN 2 (VPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

```
CAL<n>:OUTP:VOLT:EXTI:FSCR <100% channel rating voltage>
```

13. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

13.6.5 Current programming – 10V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Voltage source and 10V range. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR VOLT
SOUR<n>:CURR:PROG INT
SOUR<n>:CURR:PROG:FSC 10
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:CURR:EXTV:OFFS 0
```

9. To calibrate channel 1, Set the DC voltage to 10V +/-0.003V at PIN 4 (IPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
10. To calibrate channel 2, Set the DC voltage to 10V +/-0.003V at PIN 4 (IPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
11. To calibrate channel 3, Set the DC voltage to 10V +/-0.003V at PIN 4 (IPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

12. Set the DC voltage at PIN 19 to 10V +/-0.003V. Calibrate the full scale point.

```
CAL<n>:OUTP:CURR:EXTV:FSC <100% channel rating current>
```

13. Store the Calibration.

```
CAL<n>:STORE
```

```
CAL<n>:LOCK
```

13.6.6 Current programming – 5V range voltage source calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero volt (0 +/- 0.003V) across PIN 4 (IPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Voltage source and 5V range. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR VOLT
```

```
SOUR<n>:CURR:PROG INT
```

```
SOUR<n>:CURR:PROG:FSC 5
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:CURR:EXTV:OFFS 0
```

9. To calibrate channel 1, Set the DC voltage to 5V +/-0.003V at PIN 4 (IPRG1_VSOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.

10. To calibrate channel 2, Set the DC voltage to 5V +/-0.003V at PIN 4 (IPRG2_VSOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
11. To calibrate channel 3, Set the DC voltage to 5V +/-0.003V at PIN 4 (IPRG3_VSOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

```
CAL<n>:OUTP:CURR:EXTV:FSC <100% channel rating current>
```

12. Store the Calibration.

```
CAL<n>:STORE
```

```
CAL<n>:LOCK
```

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

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero Ohm across PIN 5 (IPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero Ohm across PIN 5 (IPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero Ohm across PIN 5 (IPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR CURR
```

```
SOUR<n>:CURR:PROG INT
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:CURR:EXTI:OFFS 0
```

9. To calibrate channel 1, Connect 10k Ohm across PIN 5 (IPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.

10. To calibrate channel 2, Connect 10k Ohm across PIN 5 (IPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.

11. To calibrate channel 3, Connect 10k Ohm across PIN 5 (IPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

```
CAL<n>:OUTP:CURR:EXTI:FSCR <100% channel rating current>
```

12. Store the Calibration.

```
CAL<n>:STORE
```

```
CAL<n>:LOCK
```

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

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. To calibrate channel 1, Connect zero Ohm across PIN 5 (IPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.
4. To calibrate channel 2, Connect zero Ohm across PIN 5 (IPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.
5. To calibrate channel 3, Connect zero Ohm across PIN 5 (IPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.
6. Select the Analog programming source as Current source. Also select programming mode as internal during calibration procedure.

```
SOUR<n>:ANALOG:PROG:SOUR CURR
```

```
SOUR<n>:CURR:PROG INT
```

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

```
CAL<n>:UNLOCK "6867"
```

8. Calibrate the Offset point.

```
CAL<n>:OUTP:CURR:EXTI:OFFS 0
```

9. To calibrate channel 1, Connect 5k Ohm across PIN 5 (IPRG1_ISOUR) and PIN 3 (PRG_RTN) of channel 1 Analog Programming connector.

10. To calibrate channel 2, Connect 5k Ohm across PIN 5 (IPRG2_ISOUR) and PIN 3 (PRG_RTN) of channel 2 Analog Programming connector.

11. To calibrate channel 3, Connect 5k Ohm across PIN 5 (IPRG3_ISOUR) and PIN 3 (PRG_RTN) of channel 3 Analog Programming connector.

```
CAL<n>:OUTP:CURR:EXTI:FSCR <100% channel rating current>
```

12. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

13.7 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 8.4

13.7.1 Voltage Monitor Output Calibration

STEP	DESCRIPTION
------	-------------

1. Issue a *CLS command.
2. Issue a *RST command.
3. Connect the DMM at PIN 7 (VMON1) and PIN 6 (MON_RTN) for channel 1, PIN 7 (VMON2) and PIN 6 (MON_RTN) for channel 2 and PIN 7 (VMON3) and PIN 6 (MON_RTN) for channel 3 to measure voltage monitor pin voltage and Set the load for approximately 25% of rated output current respected channel. Type the following commands.

```
SOUR<n>:VOLT 0
SOUR<n>:CURR <100% channel current rating>
OUTP<n>:STAT 1
CAL<n>:UNLOCK "6867"
CAL<n>:INIT:VOLT:MON
```

4. Calibrate the Offset point.

```
CAL<n>:OUTP:VOLT:MON:OFFS <voltage at PIN7 read from DMM>
```

5. Set the output voltage to 80% of channel Full-scale rated voltage.

```
SOUR<n>:VOLT <0.8 times channel Full-scale voltage>
CAL<n>:OUTP:VOLT:MON:FSC <voltage at PIN7 read from DMM>
```

6. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

13.7.2 Current Monitor Output Calibration

STEP DESCRIPTION

1. Issue a *CLS command.
2. Issue a *RST command.
3. Connect the DMM at PIN 8 (IMON1) and PIN 6 (MON_RTN) for channel 1, PIN 8 (IMON2) and PIN 6 (MON_RTN) for channel 2 and PIN 8 (IMON3) and PIN 6 (MON_RTN) for channel 3. Connect a short across the output of respected channel. Type the following commands.

```
SOUR<n>:VOLT <UUT's Full-scale voltage>
SOUR<n>:CURR 0
OUTP<n>:STAT 1
CAL<n>:UNLOCK "6867"
CAL<n>:INIT:CURR:MON
```

4. Calibrate the Offset point.

```
CAL<n>:OUTP:CURR:MON:OFFS <voltage at PIN8 read from DMM>
```

5. Set the output current to 80% of channel Full-scale rated current.

```
SOUR<n>:CURR <0.8 times channel Full-scale current>
CAL<n>:OUTP:CURR:MON:FSC <voltage at PIN8 read from DMM>
```

6. Store the Calibration.

```
CAL<n>:STORE
CAL<n>:LOCK
```

This page intentionally left blank.

SCPI STATUS IMPLEMENTATION

14.1 SCPI Status Byte Implementation

SCPI status byte implementation for the Asterion DC Multioutput Series power supply is described in Figure 14-1.

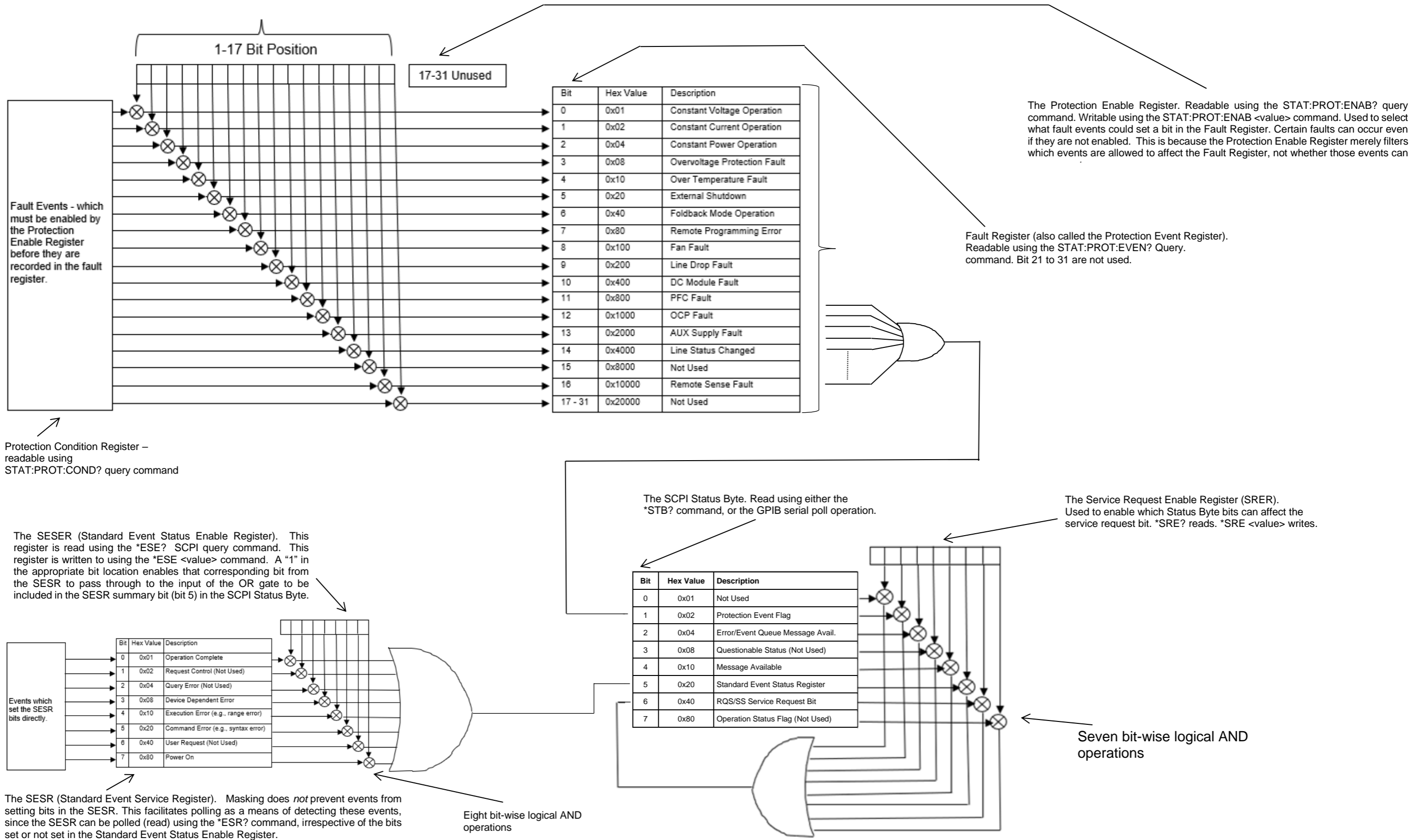


Figure 14-1. SCPI Status Implementation Diagram

