OPERATOR'S MANUAL

BOP 2X-DG (C) (L) 200W, 400W
DUAL CHANNEL
BIPOLAR OPERATIONAL POWER SUPPLY

200W: BOP 2X5-20DG, BOP 2X20-5DG, BOP 2X50-2DG, BOP 2X100-1DG
400W: BOP 2X5-30DG, BOP 2X20-10DG, BOP 2X36-6DG, BOP 2X50-4DG, BOP 2X72-3DG, BOP 2X100-2DG

KEPCO INC.

MODEL
BOP 2X-DG 200W, 400W
POWER SUPPLY

ORDER NO.	REV. NO

IMPORTANT NOTES:

1) This manual is valid for the following Model and associated serial numbers:

MODEL	SERIAL NO.	REV. NO.

2) A Change Page may be included at the end of the manual. All applicable changes and revision number changes are documented with reference to the equipment serial numbers. Before using this Instruction Manual, check your equipment serial number to identify your model. If in doubt, contact your nearest Kepco Representative, or the Kepco Documentation Office in New York, (718) 461-7000, requesting the correct revision for your particular model and serial number.

3) The contents of this manual are protected by copyright. Reproduction of any part can be made only with the specific written permission of Kepco, Inc.

Data subject to change without notice.

©2023, KEPCO, INC
P/N 243-1421-r1

KEPCO®
THE POWER SUPPLIER™

KEPCO, INC. ● 131-38 SANFORD AVENUE ● FLUSHING, NY. 11355 U.S.A. ● TEL (718) 461-7000 ● FAX (718) 767-1102
email: hq@kepcopower.com ● World Wide Web: www.kepcopower.com

TABLE OF CONTENTS

SECTION PAGE

SECTION 1 - INTRODUCTION

1.1 Scope of Manual. 1-1
1.2 Model Suffix Explanations 1-1
1.2.1 Meters (D suffix): 1-1
1.2.2 Capacitive Load Optimization (C Suffix): 1-1
1.2.3 Inductive Load Optimization (L Suffix): 1-1
1.2.4 GPIB (IEEE 488) Interface (G Suffix): 1-1
1.2.5 LAN Interface (E Suffix): 1-1
1.3 General Description. 1-1
1.4 Electrical Specifications, General 1-2
1.5 Miscellaneous Features 1-8
1.5.1 Output Range 1-8
1.5.2 Rear Programming Features 1-9
1.5.3 Remote ON-OFF Feature 1-12
1.5.4 Load Reactance 1-12
1.5.4.1 Driving an Inductive Load in Current Mode 1-12
1.5.4.2 Driving a Capacitive Load in Voltage Mode 1-13
1.5.5 Remote Turn-off 1-13
1.5.6 Circuit Breaker Shutdown Prevention (at Power Loss) Circuit 1-13
1.5.7 References 1-13
1.5.8 Series Or Parallel Operation 1-13
1.5.9 Mode Setting 1-13
1.5.10 Voltage Control Loop 1-13
1.5.11 Current Control Loop 1-13
1.5.12 Mode Lights 1-14
1.5.13 Standards 1-14
1.6 Mechanical Specifications 1-14
1.7 Equipment Supplied 1-14
1.8 Accessories 1-14
1.9 Safety 1-15

SECTION 2 - INSTALLATION

2.1 Unpacking and Inspection. 2-1
2.2 Terminations, Controls and Indicators 2-1
2.3 A-C Input Requirements 2-3
2.4 Cooling 2-4
2.5 Set Start-up Default 2-4
2.5.1 Set GPIB (Device) Address 2-4
2.5.2 Set Baud Rate for RS 232 Interface 2-4
2.5.3 Power Supply Identification 2-6
2.6 Preliminary Checkout 2-6
2.6.1 Preliminary Checkout using GPIB or RS 232 Interface 2-7
2.7 Installation of BOP 2X-DG Series Models 2-7
2.7.1 Slide Installation. 2-8
2.8 Safety Precautions 2-9
2.9 Load Connection 2-10
2.9.1 Load Connection, Local Sensing 2-10
2.9.2 Load Connection, Remote Sensing 2-10
2.9.3 A-c Ground 2-10
2.9.4 D-c Ground 2-11
2.10 GPIB Connections 2-11
2.11 RS 232 Connections 2-11

TABLE OF CONTENTS

SECTION PAGE

SECTION 3 - OPERATION

3.1 Introduction 3-1
3.2 Power Supply Basics 3-1
3.2.0.1 Factory Defaults 3-1
3.2.1 Analog Control 3-1
3.2.2 PC 15 Rear Programming Connector 3-3
3.2.3 External Programming Resistors. 3-3
3.2.4 Active Programming Sources 3-4
3.2.5 External Leads 3-4
3.2.6 External Voltage Monitor 3-4
3.2.7 External Current Monitor 3-4
3.3 Remote Analog Control of Voltage Loop 3-6
3.3.1 Remote Voltage Control Using Resistance (Potentiometer) 3-7
3.3.2 Remote Voltage Control Using Resistance (Decade Box) 3-7
3.3.3 Remote Voltage Control Using D-C Signal Voltage 3-8
3.3.4 Using the BOP as a Voltage Amplifier 3-11
3.3.5 Remote Analog Control of Current Limiting 3-13
3.4 Remote Control of Current Control Channel 3-14
3.4.1 Remote Current Control Using Resistance (Potentiometer) 3-15
3.4.2 Remote Current Control Using Resistance (Decade Box) 3-15
3.4.3 Remote Current Control Using D-C Signal Voltage 3-16
3.4.4 Using the BOP as a Current Amplifier 3-17
3.4.5 Remote Control of Voltage Limiting 3-19
3.5 Operating a Channel Using Remote Digital Interface 3-20
3.5.1 Programming Techniques to Optimize performance 3-21
3.5.1.1 Setting BOP Voltage and Current Limits 3-21
3.5.1.2 Automatic Range Operation. 3-21
3.5.1.3 Using the GPIB or RS 232 interface to Produce a Software-timed Ramp at the Output 3-22
3.5.2 Making Sure the Previous Command is Complete 3-22
3.5.3 Password Setup 3-23
3.6 Built in Test 3-23
3.6.1 Power-up Test 3-23
3.6.2 Confidence Test 3-27
3.6.3 Channel test 3-27
3.7 RS 232 Serial INterface 3-27
3.7.1 RS 232 Implementation 3-28
3.7.1.1 Echo Mode 3-29
3.7.1.2 XON XOFF Method 3-30
3.7.1.3 Isolating RS 232 Communication Problems 3-30
3.7.2 Using SCPI commands for RS 232 Communication. 3-30
3.8 SCPI Programming 3-30
3.8.1 SCPI Messages 3-31
3.8.2 Common Commands/Queries 3-31
3.8.3 SCPI Subsystem Command/Query Structure 3-31
3.8.3.1 Initiate Subsystem 3-31
3.8.3.2 Measure Subsystem 3-31
3.8.3.3 [Source:]voltage And [Source:]current Subsystems 3-31
3.8.3.4 Output Subsystem 3-32
3.8.3.5 List Subsystem 3-32
3.8.3.5.1 Required LIST Commands 3-33
3.8.3.5.2 Other Required Commands 3-33
3.8.3.5.3 Other Useful Commands 3-33
3.8.3.5.4 Optional Commands 3-34
3.8.3.6 Status Subsystem 3-34
3.8.3.7 System Subsystem 3-34
3.8.4 Program Message Structure 3-34

TABLE OF CONTENTS

SECTION PAGE

3.8.4.1 Keyword ... 3-35
3.8.4.2 Keyword Separator ... 3-35
3.8.4.3 Query Indicator ... 3-35
3.8.4.4 Data ... 3-35
3.8.4.5 Data Separator ... 3-35
3.8.4.6 Message Unit Separator ... 3-35
3.8.4.7 Root Specifier ... 3-36
3.8.4.8 Message Terminator ... 3-36
3.8.5 Understanding The Command Structure ... 3-36
3.8.6 Program Message Syntax Summary ... 3-37
3.8.7 SCPI Program Example ... 3-38
3.9 Enhanced Operation ... 3-42
3.9.1 Error Detection ... 3-42
3.9.2 Limit Channel Control ... 3-43
3.9.3 Using and Saving System Variables ... 3-43
3.9.4 Flash Memory EEPROM Operation ... 3-43
3.9.4.1 Calibration Storage ... 3-47
3.10 Series and Parallel Connection of BOP Channels ... 3-48
3.10.1 Series Connection, Master/slave Operation ... 3-49
3.10.2 Circuit Breaker Control Circuit ... 3-51
3.10.3 Parallel Connection, Master/Slave Operation ... 3-53

SECTION 4 - CALIBRATION

4.1 Introduction ... 4-1
4.2 Test Equipment Requirements ... 4-2
4.3 Internal Adjustments and Calibrations ... 4-4
4.3.1 Adjustment of ±10 Volt Calibration Controls (R31, R32) ... 4-4
4.3.2 Preamplifier “A” Zero (R41) ... 4-4
4.3.3 Preamplifier “B” Zero (R42) ... 4-5
4.3.4 Voltage Channel Zero Adjustment (R81) ... 4-6
4.3.5 Ammeter Zero Control Adjustment (R50) ... 4-6
4.3.6 Current Channel Zero Adjustment (R83) ... 4-7
4.3.7 Full Scale Output Current Calibration (R316) ... 4-7
4.3.8 Adjustment of I_O-OFF Zero (A1R133) and E_O-OFF Zero (A1R136) ... 4-8
4.4 Calibration of BOP 2X-DG Channel Using GPIB Interface ... 4-9
4.4.1 Manual calibration ... 4-12
4.4.2 Calibration Using IVI Driver ... 4-16
4.4.2.1 Setup ... 4-16
4.4.2.2 Main Control Panel ... 4-17
4.4.2.3 Calibration Controls ... 4-18
4.4.2.4 Calibration Procedure ... 4-18

APPENDIX A - SCPI COMMON COMMAND/QUERY DEFINITIONS

A.2 *CLS — Clear Status Command ... A-1
A.3 *ESE — Standard Event Status Enable Command ... A-1
A.4 *ESE? — Standard Event Status Enable Query ... A-2
A.5 *ESR? — Event Status Register Query ... A-2
A.6 *IDN? — Identification Query ... A-2
A.7 *OPC — Operation Complete Command ... A-3
A.8 *OPT? — Options Query ... A-4
A.9 *RCL — Recall Command ... A-4
A.10 *RST — Reset Command ... A-4
A.11 *SAV — Save Command ... A-4
A.12 *SRE — Service Request Enable Command ... A-4
A.13 *SRE? — Service Request Enable Query ... A-5

TABLE OF CONTENTS

SECTION PAGE

A.14 *STB? — Status Byte Register Query A-5
A.15 *TRG — Trigger Command A-5
A.16 *TST? — Self Test Query A-5
A.17 *WAI — Wait-To-Continue Command A-5

APPENDIX B - SCPI COMMAND/QUERY DEFINITIONS

B.1 Introduction B-1
B.2 Numerical Values B-2
B.3 CALibrate:CPRotect Command B-2
B.4 CALibrate:STATus Command B-2
B.5 CALibrate[:STATus]? Query B-2
B.6 CALibrate:CURRent Command B-3
B.7 CALibrate:DATA Command B-3
B.8 CALibrate:DPOT Command B-3
B.9 CALibrate:LCURR Command B-4
B.10 CALibrate:LVOLT Command B-4
B.11 CALibrate:SAVE Command B-4
B.12 CALibrate:VOLTage Command B-4
B.13 CALibrate:VPRotect Command B-4
B.14 CALibrate:ZERO Command B-4
B.15 INITiate[:IMMediate] Command B-5
B.16 INITiate:CONTinuous Command B-5
B.17 INITiate:CONTinuous Query B-5
B.18 MEASure[:SCALar]:CURRent[:DC]? Query B-5
B.19 MEASure[:VOLTage][:SCALar][:DC]? Query B-5
B.20 OUTPut[:STATe] Command B-5
B.21 OUTPut[:STATe] Query B-5
B.22 [SOURce:]FUNCtion:MODE Command B-6
B.23 [SOURce:]FUNCtion:MODE? Query B-6
B.24 [SOURce:]FUNCtion:MODE:TRIGger B-6
B.25 [SOURce:]FUNCtion:MODE:TRIGger? Query B-6
B.26 [SOURce:]LIST:CLEar Command B-6
B.27 [SOURce:]LIST:COUNt Command B-6
B.28 [SOURce:]LIST:COUNt? Query B-6
B.29 [SOURce:]LIST:COUNt:SKIP Command B-7
B.30 [SOURce:]LIST:COUNt:SKIP? Query B-7
B.31 [SOURce:]LIST:CURRent Command B-7
B.32 [SOURce:]LIST:CURRent? Query B-7
B.33 [SOURce:]LIST:CURRent:POINts? Query B-8
B.34 [SOURce:]LIST:DIRection Command B-8
B.35 [SOURce:]LIST:DIRection? Query B-8
B.36 [SOURce:]LIST:DWELl Command B-8
B.37 [SOURce:]LIST:DWELl? Query B-8
B.38 [SOURce:]LIST:DWELl:POINts? Query B-8
B.39 [SOURce:]LIST:GENeration Command B-12
B.40 [SOURce:]LIST:GENeration? Query B-12
B.41 [SOURce:]LIST:QUERy Command B-12
B.42 [SOURce:]LIST:QUERy? Query B-12
B.43 [SOURce:]LIST:SEQuence Command B-12
B.44 [SOURce:]LIST:SEQuence? Query B-13
B.45 [SOURce:]LIST:VOLTage Command B-13
B.46 [SOURce:]LIST:VOLTage? Query B-13
B.47 [SOURce:]LIST:VOLTage:POINts? Query B-13
B.48 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude] Command B-13
B.49 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude] Query B-13

TABLE OF CONTENTS

SECTION		PAGE
B.50	[SOURce:]CURRent:MODe Command	B-14
B.51	[SOURce:]CURRent:MODE? Query	B-15
B.52	[SOURce:]CURRent[:LEVel]RANGe Command	B-15
B.53	[SOURce:]CURRent[:LEVel]RANGe? Query	B-15
B.54	[SOURce:]CURRent:[:LEVel]RANGe:AUTO Command	B-15
B.55	[SOURce:]CURRent[:LEVel]TRIGgered[:AMPlitude] Command	B-15
B.56	[SOURce:]CURRent[:LEVel]TRIGgered[:AMPlitude]? Query	B-15
B.57	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude] Command	B-16
B.58	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude]? Query	B-16
B.59	[SOURce:]VOLTage:MODE Command	B-16
B.60	[SOURce:]VOLTage:MODE? Query	B-17
B.61	[SOURce:]VOLTage[:LEVel]:RANGe Command	B-17
B.62	[SOURce:]VOLTage[:LEVel]:RANGe? Query	B-17
B.63	[SOURce:]VOLTage[:LEVel]:RANGe:AUTO Command	B-17
B.64	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude] Command	B-17
B.65	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude]? Query	B-18
B.66	STATus:OPERation:CONDition Query	B-18
B.67	STATus:OPERation:ENABle Command	B-18
B.68	STATus:OPERation:ENABle? Query	B-18
B.69	STATus:OPERation[:EVENt]? Query	B-18
B.70	STATus:PRESet Command	B-18
B.71	STATus:QUEStionable[:EVENt]? Query	B-19
B.72	STATus:QUEStionable:CONDition? Query	B-20
B.73	STATus:QUEStionable:ENABle Command	B-20
B.74	STATus:QUEStionable:ENABle? Query	B-20
B.75	SYSTem:BEEP Command	B-20
B.76	SYSTem:COMMunication:SERial:ECHO Command	B-20
B.77	SYSTem:COMMunication:SERial:ECHO? Query	B-20
B.78	SYSTem:COMMunication:SERial:PACE Command	B-20
B.79	SYSTem:COMMunication:SERial:PACE? Query	B-20
B.80	SYSTem:ERRor? Query	B-21
B.81	SYSTem:ERRor:CODE? Query	B-22
B.82	SYSTem:ERRor:CODE:ALL? Query	B-22
B.83	SYSTem:PASSword:CENable Command	B-22
B.84	SYSTem:PASSword:CDISable Command	B-22
B.85	SYSTem:PASSword:NEW Command	B-22
B.86	SYSTem:PASSword:STATe? Query	B-22
B.87	SYSTem:REMote Command	B-22
B.88	SYSTem:REMote? Query	B-22
B.89	SYSTem:SECurity:IMMediate Command	B-22
B.90	SYSTem:SET Command	B-23
B.91	SYSTem:SET? Query	B-23
B.92	SYSTem:VERSion? Query	B-23

LIST OF FIGURES

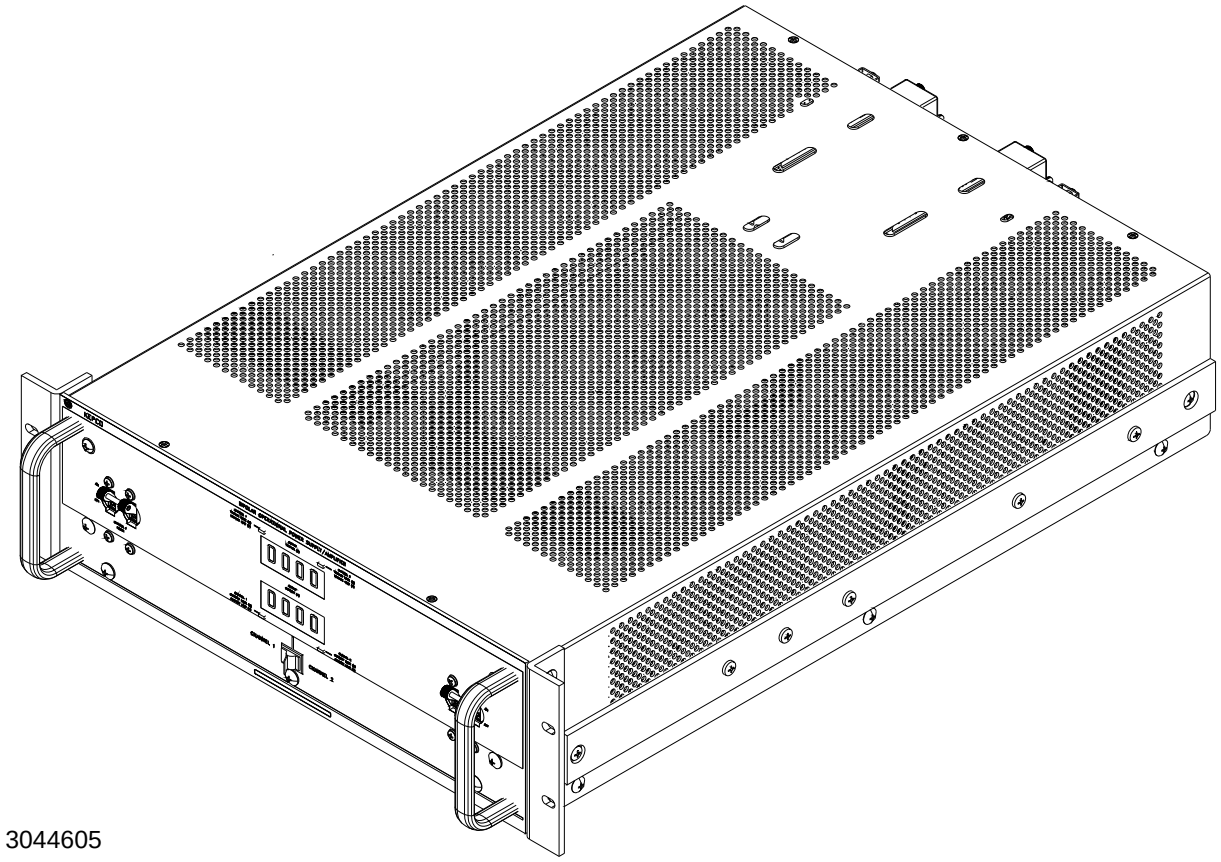
FIGURE	TITLE	PAGE
1-1	BOP 2X-DG Series 200W, 400W Dual Channel Power Supply	x
1-2	BOP Channel Output Characteristic	1-9
1-3	BOP Output WaveForms with Phase Shift	1-9
1-4	Mechanical Outline Drawing, (BOP 2X-DG Series Models)	2-17
2-1	BOP 2X-DG Controls and Indicators, Front Panel	2-2
2-2	BOP 2X-DG Terminations and Controls, Rear Panel	2-3
2-3	A-C Source Voltage Selection, Each Channel	2-3
2-4	Setting the GPIB Address	2-5
2-5	Slide Installation on BOP 2X-DG Units	2-9
2-6	Load Connection (Channel 1, Typ.) Using Local Sensing	2-12
2-7	Load Connection Using Remote Sensing	2-12
3-1	BOP Basic Voltage Control Loop (Local Sensing)	3-5
3-2	BOP Basic (±) Current Limiting Circuit (Local Sensing)	3-5
3-3	BOP BASIC Current Control (Local Sensing)	3-6
3-4	BOP BASIC (±) Voltage Limiting Circuit (Local Sensing)	3-6
3-5	Remote Potentiometer Control of D-C Output Voltage	3-7
3-6	Remote D-C Output Voltage Control Using Two-Terminal Resistance (Decade)	3-8
3-7	External Digital Control of the D-C Output Voltage	3-9
3-8	D-C Output Voltage Control with Non-inverting Configuration and High Impedance ±1V Signal Source	3-10
3-9	Graphs Of Possible Input/output Waveshapes	3-11
3-10	Basic Programming Circuit for Use of BOP as Bipolar Amplifier (Voltage Mode)	3-12
3-11	Programming Circuit for Driving Output Voltage with ±1V Input Programming Voltage	3-13
3-12	Symmetrical Remote Control of the Current Limits	3-14
3-13	Independent Remote Control of + I_O and – I_O Limits	3-14
3-14	Remote Potentiometer Control of the BOP Output Current	3-15
3-15	Remote D-C Output Current Control Using Two-Terminal Resistance (Decade)	3-16
3-16	Digital Control of Output Current	3-17
3-17	BOP Output Current Control with Non-inverting Configuration and High Impedance (±)1 Volt Signal Source	3-17
3-18	Basic Programming Circuit for BOP as Bipolar Current-stabilized Amplifier	3-18
3-19	Programming Circuit for Driving Output Current with ±1V Input Programming Voltage	3-18
3-20	Programming Circuit for Driving Output Current with a High Impedance Source	3-19
3-21	Symmetrical Remote Control of Voltage Limit	3-20
3-22	Independent Remote Control of $+E_O$ and $–E_O$ Voltage Limit	3-20
3-23	Programming Example to Verify Previous Command has Completed	3-24
3-24	RS 232 Implementation	3-28
3-25	Tree Diagram of SCPI Commands Used with BIT 4886 Interface Card	3-32
3-26	Message Structure	3-34
3-27	Typical Example of Interface Card Program Using SCPI Commands	3-39
3-28	Series Connection, Voltage Mode (shown), Master/Slave with Digital Control of Master, Remote Error Sensing, Slave BIT CARD Disabled	3-50
3-29	Circuit Breaker Control Circuit, Simplified Diagram.	3-52
3-30	Parallel Connection, Voltage Mode (shown) Master/Slave with Digital Control of Master, Remote Error Sensing, Slave BIT CARD Disabled	3-54
3-31	Simplified Schematic Diagram, Channel 1 or 2, BOP 2X SERIES	3-55
3-32	PC 15/PC 12 Rear Programming Connector Interconnections	3-57
4-1	Locations of Adjustment and Calibration Controls	4-2
4-2	Current Shunt Connections	4-3
4-3	BOP 2X-DG Power Supply, Top View, Cover Removed, R316 Location	4-8
4-4	Manual Calibration Controls (Top View)	4-13
4-5	Input VISA Resource Descriptor	4-17
4-6	Main Control Panel, Typical for BOP 2X20-10DG, Channel 2	4-17
4-7	Calibration Controls	4-18
4-8	Password Entry	4-19
4-9	Voltage Calibration	4-19
A-1	IEEE 488 Commands	A-3
B-1	Using Calibration Commands and Queries	B-3

LIST OF FIGURES

FIGURE	TITLE	PAGE
B-2	Using LIST Commands for Sawtooth and Triangle Waveforms	B-9
B-3	Using LIST Commands and Queries	B-10
B-4	Programming the Output	B-14
B-5	Using Recall and Trigger Functions	B-16
B-6	Using Status Commands and Queries	B-19
B-7	Setting the Unit to Remote Mode via RS 232 (Serial) Port	B-23
B-8	Using System Commands and Queries	B-23

LIST OF TABLES

TABLE	TITLE	PAGE
1-1	General Specifications	1-3
1-2	Input Specifications Per Channel	1-7
1-3	Output Ranges, Transfer Factor and Output Impedance per Channel	1-7
1-4	Output Dynamic Specifications PER Channel	1-8
1-5	Rear Programming Connector (1 per Channel) Terminal Functions	1-10
1-6	Equipment Supplied	1-14
1-7	Accessories - Not Supplied	1-14
1-8	Safety Symbols	1-15
2-1	BOP 2X-DG Front Panel Controls and Indicators	2-1
2-2	BOP 2X-DG Rear Panel Terminations and Controls	2-2
2-3	GPIB (Device) Address Selection	2-5
2-4	Chassis Slide Mounting Holes	2-8
3-1	Built-in test Error Codes	3-27
3-2	Enhanced Operation - Error Response	3-44
3-3	Calibration Storage	3-47
4-1	Internal Adjustments and Calibrations	4-1
4-2	Suggested Sense Resistors	4-3
4-1	Calibration Measurements and Tolerances - Voltage	4-10
4-2	Calibration Measurements and Tolerances - Current	4-10
4-3	Suggested Sense Resistors	4-12
4-4	VISA Resource String Corresponding to Interface	4-16
4-5	Calibration Panel Functions	4-18
A-1	SCPI Command/query Index	A-1
A-2	Standard Event Status Enable Register and Standard Event Status Register Bits	A-1
A-3	Service Request Enable and Status Byte Register Bits	A-5
B-1	SCPI Subsystem Command/query Index	B-1
B-2	List Data Table	B-7
B-3	List Sequence Table	B-12
B-4	Operation Condition Register, Operation Enable Register, and Operation Event Register Bits	B-18
B-5	Questionable Event Register, Questionable Condition Register and Questionable Condition Enable Register Bits	B-19
B-6	Error Messages	B-21



3044605

FIGURE 1-1. BOP 2X-DG SERIES 200W, 400W DUAL CHANNEL POWER SUPPLY

SECTION 1 - INTRODUCTION

1.1 SCOPE OF MANUAL.

This manual contains instructions for the installation and operation of the BOP 2X-DG Dual Channel Series of 200, and 400 Watt Bipolar Operational Power supplies manufactured by Kepco, Inc., Flushing, New York, U.S.A.

Throughout this manual E_O represents the rated output voltage, and I_O represents the rated output current as defined by E_O MAX and I_O MAX, respectively, in Table 1-3 for each model.

References in this manual to “BOP” apply to either Channel 1 or Channel 2. The two channels are functionally identical. References to “BOP 2X-DG” apply to the full rack chassis containing the two separate channels.

1.2 MODEL SUFFIX EXPLANATIONS

Suffixes appended to the model number are described below.

1.2.1 METERS (D SUFFIX): D suffix models are standard, and include two digital meters on the front panel used to display voltage and current for the channel selected by the Channel 1/Channel 2 select switch.

1.2.2 CAPACITIVE LOAD OPTIMIZATION (C SUFFIX): Models with a C suffix have been modified to be stable handling capacitive loads up to 10mF. A Modification Sheet describing changes and modified specifications can be downloaded from the Kepco web site at:
www.kepcopower.com/support/opmanls.htm#bop.

1.2.3 INDUCTIVE LOAD OPTIMIZATION (L SUFFIX): Models with an L suffix have been modified to be stable handling inductive loads up to 1 Henry. A Modification Sheet describing changes and modified specifications can be downloaded from the Kepco web site at:
www.kepcopower.com/support/opmanls.htm#bop.

1.2.4 GPIB (IEEE 488) INTERFACE (G SUFFIX): Models with a G suffix have two factory-installed BIT 4886 cards that allow digital remote control of each channel via GPIB using SCPI and IEEE 488 commands and queries.

1.2.5 LAN INTERFACE (E SUFFIX): Models with an E suffix have two factory-installed BIT 802E cards that allow digital control of each channel via LAN using a web browser. Operation of E suffix models are covered by a separate BOP 2X-DE (C)(L) Operator Manual (P/N 243-1405) that can be downloaded from the Kepco web site at:
www.kepcopower.com/support/opmanls.htm#bop.

1.3 GENERAL DESCRIPTION.

Kepco's Series BOP 2X-DG dual-channel BOP models are 200W and 400W power supplies. Each channel is a separate power supply, completely isolated, independent and functionally identical. The 200W units have two 100W channels and the 400W units have two 200W channels.

BOP 2X-DG Models include a separate GPIB interface used to access each channel (a tailored BIT 4886 Interface Card, also referred to as digital card) that is used for digital remote control of

the output. This interface is built-in and is not user-installable, however operation and calibration of the output channels is described in this manual.

Dual channel models come in a single full rack chassis capable of mounting in a standard 19-inch wide rack. The channels are completely isolated, but share two digital front panel meters (without affecting isolation) which display output voltage and current for the channel selected by the front panel Channel select switch. A single power cord powers both channels; separate on-off circuit breakers turn each channel on/off separately.

Each channel of Kepco Series BOP 2X-DG are fully dissipative, linear stabilizers for laboratory and systems applications. The BOP is an all solid-state design, featuring IC operational amplifiers in the control circuit section and silicon power transistors mounted on special fan-cooled heat sinks in the complementary power stage.

Each BOP channel is a linear stabilizer with two bipolar control loops (voltage or current mode), selectable and individually controllable by remote signals applied to the PC15 programming connector at the rear panel. These two principal control loops are each protected by bipolar limit circuits. The positive and negative current or voltage limit points can be remotely programmed simultaneously or Individually. Automatic crossover between each principal control loop and the limit loops is provided. Only one principle loop (voltage or current) can control the output at any one time.

All control and limit loops are connected to the complementary output stage via an EXCLUSIVE OR gate, so that only one main loop, either voltage or current, and one of the limit loops is in control of the BOP output at any one time. The BOP output and limits can be programmed via SCPI commands sent from a remote computer via either the GPIB or integral RS 232 interfaces included with Series BOP 2X-DG. The output can also be adjusted over its full output range by a ±10 volt signal applied to the Rear Programming Connector as described in Section 3 of this manual. The limit control loops can be controlled by 0 to +10 volt signals as described in Section 3 of this manual. Without any remote limit control applied to the Rear Programming Connector the limits are set to approximately 110% of E_O MAX and I_O MAX.

Two preamplifiers, with both inverting and non-inverting inputs, available at the Rear Programming Connector, can scale any programming source to the required internal control levels of the main loop. Zero adjustments for both preamplifiers are provided. By default the preamplifiers are set to an inverting configuration.

The BOP is a fast programmable power supply. The dynamic specifications for each channel, given in Table 1-3, are valid for resistive loads.

1.4 ELECTRICAL SPECIFICATIONS, GENERAL

See Table 1-1 for general specifications. These specification apply to each BOP channel except where indicated as applying to BOP 2X-DG, referring to the entire chassis comprised of two BOP channels.

TABLE 1-1. GENERAL SPECIFICATIONS

SPECIFICATION		RATING/DESCRIPTION (1)	CONDITION/COMMENT
Input Voltage (Nominal)		104/ 115/ 208/ 230Vrms	Range: ±9% Internal setting for the nominal input
Input Brown-out Voltage		Max 103.5Vrms	@ 115V/60Hz, Sourcing nominal output power
Input Frequency (Nominal)		50/ 60Hz	Range:± 3Hz
Input Specifications		Input current, input power (active), power factor and efficiency differ by model.	See Table 1-2.
Input Leakage Current		50uArms	@ 115V/ 60Hz, Sourcing nominal output power
Insulation Coordination		Installation Cat. 2 Over-voltage Cat. 2 Pollution Degree 2	Maximum 250V DC or PK between COM output terminal and chassis-ground
Type Of Stabilizer		Voltage-Current, 4-quadrant	
Topology		Linear, Half Bridge	Class AB
Output Voltage		0 to $\pm E_O$max (Volts DC or peak)	See Table 1-3 for $E_O$max values.
Output Current		0 to $\pm I_O$max (Amperes DC or peak)	See Table 1-3 for $I_O$max values.
Output Power (Nominal)			For long term operation in sink, reduce the complementary limit settings to half of nominal. See Table 1-3 for nominal output power values.
	Source Mode	$E_O$max x $I_O$max (Watts DC or peak)	
	Sink Mode	($E_O$max x $I_O$max)/2 (Watts DC or peak) (2)	
Output OFF/disabled level			The output OFF level can be calibrated using trimpots accessible from the top cover only if the digital card is disabled (see PAR. 3.10.1 to disable digital card) and by referring to the applicable Service manual.
	Voltage Mode	0V ±(0.01%)($E_O$max)V	
	Current Mode	0A ±(0.01%)($I_O$max)A	
Voltage, Analog Programming (V Mode)	DC Transfer Factor	(-$E_O$max/10) V/V (default), it can be changed thru rear programming connector to (+$E_O$max/10) V/V	Install components on rear programming connector for a different preamp transfer factor. Use rear programming connector to configure non-inverting operation. There is no internal calibration for the transfer factor in voltage mode. See Table 1-3 for default d-c transfer factor and output impedance values.
	Linearity	0.005% of $E_O$max	
	FS Tolerance	2% of $E_O$max with the default preamplifier's network, or 0.02% when using external 0.01% matched pair resistors for the preamplifier	
	Output Impedance	Model dependent	

(1) Specifications listed for ambient temperature = 20°C and apply to each BOP channel unless otherwise specified.
(2) Output power - Sink mode for BOP 5-20 = $E_O$max x $I_O$max (DC or peak).

TABLE 1-1. GENERAL SPECIFICATIONS (CONTINUED)

SPECIFICATION		RATING/DESCRIPTION (1)	CONDITION/COMMENT
Voltage Static Specifications (V mode)	Source effect	0.001% of $E_O$max	Input voltage = nominal ±9%
	Load effect	0.002% of $E_O$max	No load to (1.05 x $E_O$max)/$I_O$max
	Time effect (8h)	0.01% of $E_O$max	After 30 minutes of conditioning
	Temp. coefficient	0.01% of $E_O$max/ °C	Averaged for the operating temperature range.
	Ripple & noise (rms)	the greater of 0.01% of $E_O$max or 3mV	
Current, Analog Programming (C.mode)	DC Transfer Factor	(-$I_O$max/10)A/V (default), it can be changed thru rear programming connector to (+$I_O$max/10)A/V	Install components on rear programming connector for a different preamp transfer factor. Use rear programming connector to configure non-inverting operation. Transfer factor in current mode has an internal calibration. See Table 1-3 for d-c current transfer factor and output impedance values.
	Linearity	0.02% of $I_O$max	
	FS Tolerance	0.02% of $I_O$max	
	Output Impedance	Model dependent	
Current Static Specifications (C.mode)	Source effect	0.005% of $I_O$max	Input voltage = nominal ±9%
	Load effect	0.05% of $I_O$max	Short circuit to (0.95 x $E_O$max)/$I_O$max
	Time effect (8h)	0.02% of $I_O$max	After 30 minutes of conditioning
	Temp. coefficient	0.02% of $I_O$max/ °C	Averaged for the operating temperature range.
	Ripple & noise (rms)	0.03% of $I_O$max	
Analog Programming Type			
	Voltage Source	0 to ±10V (default) Inverting (default) or Non-inverting	Uses ±10V reference applied to Preamp A for voltage mode and Preamp B for current mode via the Rear Programming Connector (see Section 3 for application schematics). Contact factory for preconfigured rear programming connector adapted to specific applications.
	Current Source	Contact factory for details	
	External Resistance	10K Ohms (default)	See Section 3 for application schematics.

TABLE 1-1. GENERAL SPECIFICATIONS (CONTINUED)

SPECIFICATION		RATING/DESCRIPTION [1]	CONDITION/COMMENT
Output ON-OFF (enable-disable) Control		Using OUTP ON or OUTP OFF digital remote commands (see Appendix B, PAR. B.20).	Non-isolated 15V-CMOS signal and isolated TTL signal at rear programming connector can only be used if digital option is disabled. (see Table 1-5,for details, see PAR. 3.10.1 to disable digital card).
Voltage Mode/Current Mode Control		Remote TTL non-isolated signal at rear programming connector (see Table 1-5, pin 30 for details)	
Calibration		Using trim-pots.	
Dynamic Specs (V mode and C mode)		Model dependent	See Table 1-4
Special Dynamic Configurations	Main loop bandwidth reduction	Possible, with changes at rear programming connector, for voltage and/or current	
	Optimized for inductive loads	Models with L suffix	
	Optimized for capacitive loads	Models with C suffix	
Voltage/Current Limits (Pos & Neg)	Analog-remote adjustment range	0.2 to 110% of nominal output, using 0V to +10V programming. voltage applied to rear programming connector,	Symmetrical or independent (positive or negative) control. See Section 3 for application schematics.
Analog Readback For Voltage and Current	Range	0 to ±10V, for 0 to ± nominal output	The analog readback signals are available at the rear programming connector.
	Full Scale tolerance	0.02% of nominal monitoring output (10V)	
	Accuracy	0.1% of nominal monitoring output (10V)	
	Max current	1mA	
Kelvin Connection (remote error sensing)		Possible, with max 0.5V/ wire [3]	
Local Output Monitoring	Type	Two Digital Panel Meters (D suffix models), one each for output voltage and current	Front panel switch allows meters to show voltage/current for Channel 1 or 2.
	Accuracy	±(0.2% of full scale +2 counts) [4]	
Reference Voltages			Reference voltages can be used for external analog remote control schemes (see Section 3). They have internal calibration pots.
	Value	+10V, -10V	
	Tolerance	±1mV	
	Load Current	4mA max.	
	Source Effect	0.0005%	Input voltage: nominal ±9%
	Load Effect	0.0005%	No load - full load
	Time Effect	0.005%	8 hours after 1/2 hour conditioning
	Temp. Coefficient	0.005%/°C	Averaged for the operating temperature range.

(3) For BOP 2X5-20D: 0.25V/wire max.
(4) E.g, for a channel rated at 100V, at rated output the digital voltmeter reads 100.0V ± (0.002 x100 + 0.2)V= 100.0V ± 0.4V. For a channel rated at 20A, at rated output the digital ammeter reads 20.0A ± (0.002 x 20 + 0.2)A = 20.0A ± 0.2A.

TABLE 1-1. GENERAL SPECIFICATIONS (CONTINUED)

SPECIFICATION		RATING/DESCRIPTION (1)	CONDITION/COMMENT
Front Panel Lights			
	Channel 1, Channel 2	VOLTAGE MODE/VOLTAGE LIMIT (5)	Lit green when in Voltage mode Lit yellow when in Voltage Limit mode
	Channel 1, Channel 2	CURRENT MODE/CURRENT LIMIT (5)	Lit green when in Current mode Lit yellow when in Current Limit mode
Flag Signals NOTE: All flags available at rear programming connector (see Table 1-5)	Voltage Mode	Active in when in voltage mode	TTL signal, active “L”, non-isolated
	Current Mode	Active in when in Current mode	TTL signal, active “L”, non-isolated
	Voltage Limit	Active in when in Voltage limit	TTL signal, active “L”, non-isolated
	Current Limit	Active in when in Current limit	TTL signal, active “L”, non-isolated
	Output On-off	Output on-off	Not used since independent on-off control function is under digital control.
BOP 2X-DG Front and rear panel controls, indicators and terminals,		Front Panel: See Figure 2-1 and Table 2-1. Rear Panel: See Figure 2-2 and Table 2-2.	
Series Or Parallel Configurations of BOP channels	Type of connection	Automatic or Master-Slave	For implementation and limitations, see PAR. 3.10.1 for series configurations and PAR. 3.10.3 for parallel configurations
	Implementation	Configured by external wiring at rear programming connectors	
Protections		(a) Input overcurrent, with 110% threshold (b) Heat-Sink overtemperature (c) Channels or Multiple units interlock (d) Input power loss	Trips Input circuit breaker. Tripping of the circuit breaker for input power loss can be disabled by a modification to the rear programming connector (see PAR. 3.10.2).
Temperature	Operating	0 to +55° C	BOP 2X-DG
	Storage	-40 to +85° C	BOP 2X-DG
Humidity		0 to 95% RH	BOP 2X-DG, Non-condensing
Shock		20g/11ms/50% half sine	BOP 2X-DG, Non-operating
Vibration	5-10Hz	10mm, double amplitude	BOP 2X-DG, Non-operating
	10-55Hz	2g	
Altitude		Sea level to 10,000ft	BOP 2X-DG, For operating temperature range
Cooling	Type	BOP 2X-DG 200W, 400W: 1 fan per channel	Minimum 1U between stacked units.
	Air Flow	Front to rear	
	Air Intake	top, bottom, side	
Dimensions	BOP 2X-DG (200W, 400W)	5.22” H x 19” W x 26” D (132.6mm H x x482.6mm W x 558.8mm D)	
Weight	BOP 2X-DG 200W (BOP 2X-DG 400W	80 lbs (36.3 kg) 90 lbs (40.8 kg)	Unpacked
Connections	BOP 2X-DG Source	3-pin IEC connector	
	Load (power and sense) Grounding, GND network	Rear: 6-pin terminal block	
	Analog control	50-pin rear programming connector	
(5) Front panel lights always active for both channel 1 and channel 2, regardless of channel selection for meter display.			

TABLE 1-2. INPUT SPECIFICATIONS PER CHANNEL

MODEL (1)	INPUT CURRENT (2) (A)	ACTIVE INPUT POWER (2) (W)	POWER FACTOR (2)	EFFICIENCY (2) (%)
100 WATTS/CHANNEL				
BOP 2X5-20DG	4.7	460	0.84	22
BOP 2X20-5DG	2.9	270	0.81	37
BOP2X 50-2DG	2.9	270	0.81	37
BOP 2X100-1DG	2.9	270	0.81	37
200 WATTS/CHANNEL				
BOP 2X5-30DG	6.4	640	0.87	24
BOP 2X20-10DG	5.5	540	0.85	37
BOP 2X36-6DG	5.1	490	0.84	44
BOP2X 50-4DG	4.8	450	0.82	44
BOP 2X72-3DG	5.1	470	0.80	46
BOP 2X100-2DG	4.8	450	0.82	44

(1) Models with digital meters include suffix D.
(2) Specifications listed are per channel for Input @ 115V, 60Hz sourcing nominal power.

TABLE 1-3. OUTPUT RANGES, TRANSFER FACTOR AND OUTPUT IMPEDANCE PER CHANNEL

MODEL (1)	DC OR PEAK (2) OUTPUT RANGES		DC OR PEAK (2) OUTPUT POWER		d-c CLOSED LOOP (2) GAIN		OUTPUT IMPEDANCE (2) (3)			
					VOLTAGE CHANNEL	CURRENT CHANNEL	VOLTAGE MODE		CURRENT MODE	
	E_O MAX. V	I_O MAX A	SOURCE W	SINK W	G_V (V/V)	G_I (A/V)	SERIES R mΩ	SERIES L mH	SHUNT R kΩ	SHUNT C μF
100 WATTS/CHANNEL										
BOP 2X5-20DG	0 to ±5	0 to ±20	100	100	0.5	2.0	0.01	0.004	5.0	0.3
BOP 2X20-5DG	0 to ±20	0 to ±5	100	50	2.0	0.5	0.08	0.02	40	0.05
BOP2X 50-2DG	0 to ±50	0 to ±2	100	50	5.0	0.2	0.5	0.1	50	0.05
BOP 2X100-1DG	0 to ±100	0 to ±1	100	50	10.0	0.1	2.0	0.2	100	0.05
200 WATTS/CHANNEL										
BOP 2X5-30DG	0 to ±5	0 to ±30	150	75	0.5	3.0	0.01	0.004	5.0	0.5
BOP 2X20-10DG	0 to ±20	0 to ±10	200	100	2.0	1.0	0.04	0.05	20	0.1
BOP 2X36-6DG	0 to ±36	0 to ±6	216	108	3.6	0.6	0.12	0.05	36	0.1
BOP2X 50-4DG	0 to ±50	0 to ±4	200	100	5.0	0.4	0.25	0.1	50	0.05
BOP 2X72-3DG	0 to ±72	0 to ±3	216	108	7.2	0.3	0.48	0.2	72	0.05
BOP 2X100-2DG	0 to ±100	0 to ±2	200	100	10.0	0.2	1.0	0.2	100	0.05

(1) Models with digital meters include suffix D.
(2) Specifications listed are per channel.
(3) Output impedance is for basic models with D suffix. For models with L or C options (DL or DC suffix) see applicable Modification Sheets available from Kepco's website at **.http://www.kepcopower.com/support/opmanls.htm#bop**

.

TABLE 1-4. OUTPUT DYNAMIC SPECIFICATIONS PER CHANNEL

MODEL (1) (7)	3dB BANDWIDTH (2) (3) KHz		RISE & FALL TIME (2) (4) 10%-90% μsec		SLEW RATE (2) (5) (average)		RECOVERY A T STEP LOAD (2) (6) μsec	
	Mode		Mode		Mode		Mode	
	V	I	V	I	V V/μsec	I A/μsec	V	I
100 WATTS/CHANNEL								
BOP 2X5-20DG	11	5	40	70	0.2	0.46	40	100
BOP 2X20-5DG	17	13	20	30	1.6	0.27	25	10
BOP2X 50-2DG	17	13	20	30	4.0	0.11	25	10
BOP 2X100-1DG	18	11	17	22	9.4	0.07	40	25
200 WATTS/CHANNEL								
BOP 2X5-30DG	12	7	30	54	0.27	0.9	50	80
BOP 2X20-10DG	17	7	20	60	1.6	0.27	80	20
BOP 2X36-6DG	15	14	23	27	2.5	0.61	50	35
BOP2X 50-4DG	15	11	14	25	5.7	0.26	40	30
BOP 2X72-3DG	17	12	18	26	6.4	0.18	30	30
BOP 2X100-2DG	17	12	18	26	8.9	0.12	30	30

(1) Models with digital meters include suffix D.
(2) Specifications listed are per channel for Input @ 115V, 60Hz.
(3) Represents 3-dB bandwidth for large sine signal for nominal resistive load/peak output voltage value @ 50Hz, equal to $E_O$max in V mode and for short circuit/peak output current @ 50Hz, equal to $I_O$max in I mode.
(4) Represents the response timing of the pulsed output (@ 100Hz) for nominal resistive load/between $\pm E_O$max in V mode and for short circuit/between $\pm I_O$max in I mode.
(5) Defined as the ratio between (1.6)($E_O$max) or (1.6)($I_O$max) and the corresponding rise/fall time for the same conditions described in Note (4) above.
(6) Represents the 10-90% timing of a changing output when the load is switched between no load and nominal resistive load in V mode and between short circuit and nominal resistive load in I mode.
(7) Dynamic specifications are for basic models with D suffix. For models with L or C options (DL or DC suffix) see applicable Modification Sheets available from Kepco's website at
.http://www.kepcopower.com/support/opmanls.htm#bop

1.5 MISCELLANEOUS FEATURES

1.5.1 OUTPUT RANGE

Each BOP channel can be remotely programmed, from (–)100% to (+)100% of its specified d-c voltage and current range. The Class AB bipolar output stage permits operation as either a *source* or a *sink*. (See Figure 1-2).

NOTE: The BOP channel is operating as a *source* if the direction of its output voltage is the same as the direction of its output current. The BOP channel is operating as a *sink* if the direction of its output voltage is opposite that of its output current. An illustrative case is shown in Figure. 1-3, where the BOP channel is programmed to deliver a sine wave output and where the load produces a phase shift between the output voltage and current.

1.5.2 REAR PROGRAMMING FEATURES

The external Rear Programming Connector (See Table 2-2, Items 22 and 23 and Figure 3-32), is a 50-pin edge connector, providing access to all circuit features. The BOP 2X-DG is delivered with two (2) PC 15 connectors, one for each channel, which are prewired with jumpers used for standard operating schemes as described in Section 3, while allowing the user to adapt the configuration to fit other application needs. Spare connectors may be purchased from Kepco under the model number PC 12 for the unwired connector and PC 15 for a generically wired connector. The functions of the Rear Programming Connector terminals are listed below in Table 1-5. Contact Kepco for Rear Programming Connectors prewired for specific functions described in Section 3.

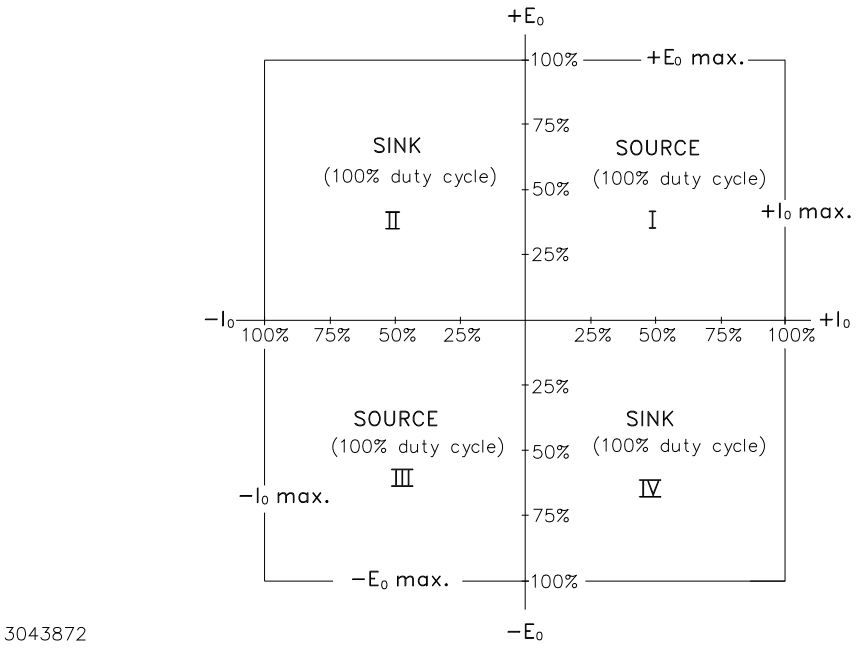


FIGURE 1-2. BOP CHANNEL OUTPUT CHARACTERISTIC

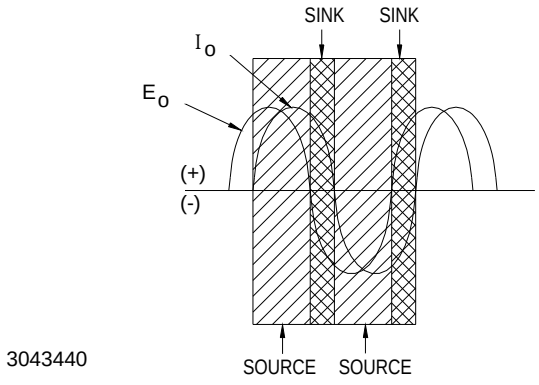


FIGURE 1-3. BOP OUTPUT WAVEFORMS WITH PHASE SHIFT

TABLE 1-5. REAR PROGRAMMING CONNECTOR (1 PER CHANNEL) TERMINAL FUNCTIONS

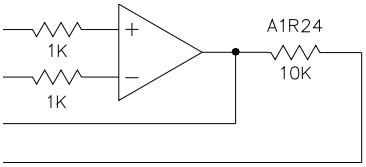
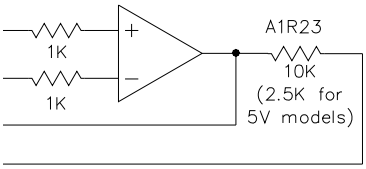
TERMINAL (Figure 3-32)	FUNCTION	DESCRIPTION
1	Current Limit Reference	Not used with digital card enabled.
3	Voltage Limit Reference	Not used with digital card enabled.
15 13 2 4	Preamp "B"  	Not used with digital card enabled.
11 9 5 7	Preamp "A"  	Not used with digital card enabled. NOTE: A1R23 = 2.5K for 5V models, 10K for all others
6	$+E_O$ Limit Input	Not used with digital card enabled.
8	$+I_O$ Limit Input	Not used with digital card enabled.
32	$-E_O$ Limit Input	Not used with digital card enabled.
34	$-I_O$ Limit Input	Not used with digital card enabled.
10	Current Monitoring	Analog output voltage (0 to ±10V, 1 mA max) proportional to the prevailing BOP output current for remote metering or monitoring.
20	Voltage Monitoring	Analog output voltage (0 to ±10V, 1 mA max) proportional to the prevailing BOP output voltage.
12, 14	Voltage Mode Compensating Terminals	Refer to PAR. 1.5.4
16, 18	Current Mode Compensating Terminals	Refer to PAR. 1.5.4
17	Voltage Comparison Amp Input	Analog input voltage of 0 to ±10V will produce 0 to ± full output voltage with unit set in voltage mode.
19	Current Comparison Amp Input	Analog input voltage of 0 to ±10V will produce 0 to ± full output current with unit set in current mode.
21	On-off Controlling Logic	NOTE: For use only if digital card is disabled (see PAR. 3.10.1). Establishes whether TTL "1" or "0" applied to pin 50, referenced to pin 48 turns the output on or off. No signal (default) or "1" (15V CMOS): "0" = ON, "1" = OFF "0" or short to pin 23: "1" = ON, "0" = OFF If no signal is applied between pin 50 and pin 48, pin 21 can be used as a non-isolated on-off control: no signal or "1" (15V CMOS) = output ON "0" or short to pin 23 = output OFF
22	–10V Reference Voltage 4mA Max	Not used with digital card enabled.
28	+10V Reference Voltage 4mA Max	Not used with digital card enabled.
23, 25, 27, 29, 31, 33	Sensing Common	
24	Circuit Breaker Control	Refer to PAR, 3.10.2
26, 36	Turn-off Signal Output	Refer to PAR, 3.10.2

TABLE 1-5. REAR PROGRAMMING CONNECTOR (1 PER CHANNEL) TERMINAL FUNCTIONS (CON-

TERMINAL (Figure 3-32)	FUNCTION	DESCRIPTION
30	Voltage Mode/Current Mode control	NOTE: For use only if digital card is disabled (see PAR. 3.10.1). Used to set either voltage mode or current mode Voltage mode = open or TTL “1” Current mode = short (referenced to sensing common, pin 23) or TTL “0”
35	Local/Digital Remote control	For use only if digital card is disabled (see PAR. 3.10.1).
37	+5V	Built-in TTL Logic supply, 5V @ 40mA max.
38	I_O Limit Flag	Not used with digital card enabled.
39	E_O Limit Flag	Not used with digital card enabled.
40	E_O Mode Flag	Not used with digital card enabled.
42	I_O Mode Flag	Not used with digital card enabled.
41	Non-isolated Turn-off Signal Input	Refer to PAR, 3.10.2
43, 44, 45	Not Used.	
46	/ON-OFF_FLAG	NOTE: For use only if digital card is disabled (see PAR. 3.10.1). Flags whether output set to ON or OFF. This isolated flag signal is the open collector of an optocoupler-transistor, with the emitter connected to pin 48. Optocoupler transistor is saturated when the output is ON (enabled). Current through the transistor should not exceed 1.6mA and supply voltage should not exceed 24V. OFF level: determined by (Vext) - (Rext) (Icoff) where: Vext = External positive voltage Rext = Vext/1.6mA (External current-limiting resistor). Icoff = 0.0001mA (optocoupler transistor maximum dark current) The flag’s rise time for either voltage mode with no load, or current mode with short circuit, is within 150μS max. ON level: +0.4V max. for 1.6mA current. The flag’s fall time for either voltage mode with no load, or current mode with short circuit, is within 150μS max.
47, 49	Optically Isolated Turn-off Signal Input	Refer to PAR, 3.10.2
48	ISO_GND	Return for ON-OFF_CTRL and for /ON_OFF_FLAG signals.
50	ON-OFF_CTRL	NOTE: For use only if digital card is disabled (see PAR. 3.10.1). Optically isolated signal that controls output to be either ON (enabled) or OFF (disabled). The logic of the control is established by pin 21 (see above). If pin 21 is not used (default status), the control levels for pin 50 are defined as follows: ON control (default): 0V to 0.4V or no connection (open circuit) at pin 50, referenced to pin 48. Turns ON within 100μS max (between 10% point of controlling signal and a) 10% point of output voltage for voltage mode with no load or b) 10% point of output current for current mode with short circuit). OFF control (default): 3.5V to 24V d-c at pin 50 referenced to pin 48; input current is 1mA ±0.3mA. Turns OFF within 100μS max (between 90% point of controlling signal and a) 90% point of output voltage for voltage mode with no load or b) 90% point of output current for current mode with short circuit).

1.5.3 REMOTE ON-OFF FEATURE

While digital control is active, output ON-OFF (Enable- Disable) control is performed remotely by the corresponding OUTP ON or OUTP OFF digital command (see PAR B.20). If the digital card is disabled, refer to the BOP 2X-D Operator Manual for details regrading the remote ON-OFF control feature. The BOP 2X-D Operator Manual is available on the Kepco web site at:

www.kepcopower.com/support/opmanls.htm#bop-2x

1.5.4 LOAD REACTANCE

1.5.4.1 DRIVING AN INDUCTIVE LOAD IN CURRENT MODE

BOP power supplies with option L (L suffix) are optimized for driving large inductor loads. Factory test were performed with loads up to 1H, however higher values are possible.

For basic units (without L suffix) when driving inductive loads higher than 0.5mH, the system may become unstable. There are two means of eliminating oscillation in the current loop operating with moderate inductive loads and moderate voltage and current:

1. Add capacitance in parallel with the output (inductive load). The value of capacitor can be from 0.1 μF to 1.0 μF depending on the value of the load inductance.

2. Add a series-connected resistor-capacitor network in parallel with the BOP output/inductive load. The value of resistor should be in the range of 100 to 500 Ohms and the value of capacitor should be 0.1 μF to 0.5 μF, depending on the load inductance value.

Both options are recommended when the BOP is used with very low frequency output into inductive loads.

CAUTION: When driving heavy inductive loads with high currents, it is possible for the BOP and the load to be damaged by an A-C input power loss which prevents the load's stored energy from being dissipated inside the BOP. To avoid possible damage from the inductance kick-back voltage observe the following:

- **Use UPS to supply the BOP.**
- **Set BOP output current and/or voltage to zero and turn off the BOP after the output current actually reaches a zero value.**
- **Use properly rated (voltage and current) bipolar transorbs or a properly rated bipolar crowbar element connected directly at the output of the BOP. An alternative is a Normally-Closed Contactor connected across the output of the BOP that will be controlled open during normal operation. The coil of the contactor will be supplied from the same input power source as the BOP. These options may also be combined.**

For both basic (without L suffix) and L suffix units, if the system becomes unstable at the crossover between voltage limit mode and current mode when driving an inductive load, the power supply can be further optimized by reducing the current loop bandwidth. This can be accomplished by connecting a ceramic capacitor across pins 16 and 18 of the Rear Programming Connector (see Figure 3-32 and Table 1-5). The value of the capacitor is correlated to the load's inductance. A value in the range of 0.0022μF to 0.1μF is recommended.

1.5.4.2 DRIVING A CAPACITIVE LOAD IN VOLTAGE MODE

BOP power supplies with option C (C suffix) are optimized for driving large capacitive loads (e.g., solar panels and solar cells). Factory test were performed with loads up to 10mF, however higher values are possible.

For basic units (without C suffix) the system can become unstable when driving a capacitive load higher than 0.2μF in voltage mode. Install an external ceramic capacitor rated for 50V across pins 12 and 14 of the rear programming connector for stable operation. This capacitor will reduce the voltage loop bandwidth. The value of the capacitor is correlated to the load's capacitance. A value in the range of 0.0047μF to 0.47μF is recommended. This method can also be used for C suffix models if the system becomes unstable at the crossover between voltage mode and current limit mode.

1.5.5 REMOTE TURN-OFF

Remote turn-off of the BOP, equivalent to tripping the front panel ON-OFF circuit breaker to OFF, can be accomplished by a contact closure, or by an optically isolated signal (see PAR. 3.10.2).

1.5.6 CIRCUIT BREAKER SHUTDOWN PREVENTION (AT POWER LOSS) CIRCUIT

See PAR. 3.10.2.

1.5.7 REFERENCES

While the digital card is enabled, the ±10V references can not be used.

1.5.8 SERIES OR PARALLEL OPERATION

See PAR. 3.10

1.5.9 MODE SETTING

While the digital card is enabled, the mode of operation, either voltage or current, is established by the FUNC:MODE digital remote command (see PAR. B.22).

1.5.10 VOLTAGE CONTROL LOOP

While the digital card is enabled, each channel's voltage loop is controlled by the corresponding reference generated by the digital card. The voltage limit references are fixed at approximately 10% above the nominal output voltage. The current limit references are accessible for remote digital programming using the CURR command while in voltage mode (See PAR. B.48).

1.5.11 CURRENT CONTROL LOOP

While the digital card is enabled, each channel's current loop is controlled by the corresponding reference generated by the digital card. The current limit references are fixed at approximately10% above the nominal output current. The voltage limit references are accessible for remote digital programming using the VOLT command while in current mode (See PAR. B.57).

1.5.12 MODE LIGHTS

Four (4) front panel (LED type) indicator lamps (two (2) for each channel) monitor the prevailing operating mode of the two BOP channels. These bicolor LEDs are always active, regardless of the position of the Channel 1/Channel 2 selector switch for meter displays. For each operating condition (Voltage Mode, ±Voltage Limit, Current Mode, ±Current Limit the corresponding indicator lamp energizes: green for Voltage mode or Current Mode, yellow for Voltage Limit or Current Limit.

1.5.13 STANDARDS

BOP models are designed and tested in accordance with NEMA Standard for Stabilized Power Supplies, d-c output, Publication No. PY-1.1972.

1.6 MECHANICAL SPECIFICATIONS

See Mechanical Outline Drawing, Figure 1-4 for dimensions and finish.

1.7 EQUIPMENT SUPPLIED

Equipment supplied is listed in Table 1-6.

TABLE 1-6. EQUIPMENT SUPPLIED

ITEM	PART NUMBER	QUANTITY
Rear Programming Connector (Local Control) - 50-terminal printed circuit connector for all rear programming connections and flag-signal outputs, wired for local (front panel) control of the BOP channel output.	PC 15	2
Power Cord, IEC connection to standard 115 VAC (USA),	118-0557	1
Mounting Flanges - for installation of the BOP into a standard (19 inch) equipment rack Right Left.	 128-0927 128-0926	 1 1

1.8 ACCESSORIES

Accessories (not supplied) are listed in Table 1-7.

TABLE 1-7. ACCESSORIES - NOT SUPPLIED

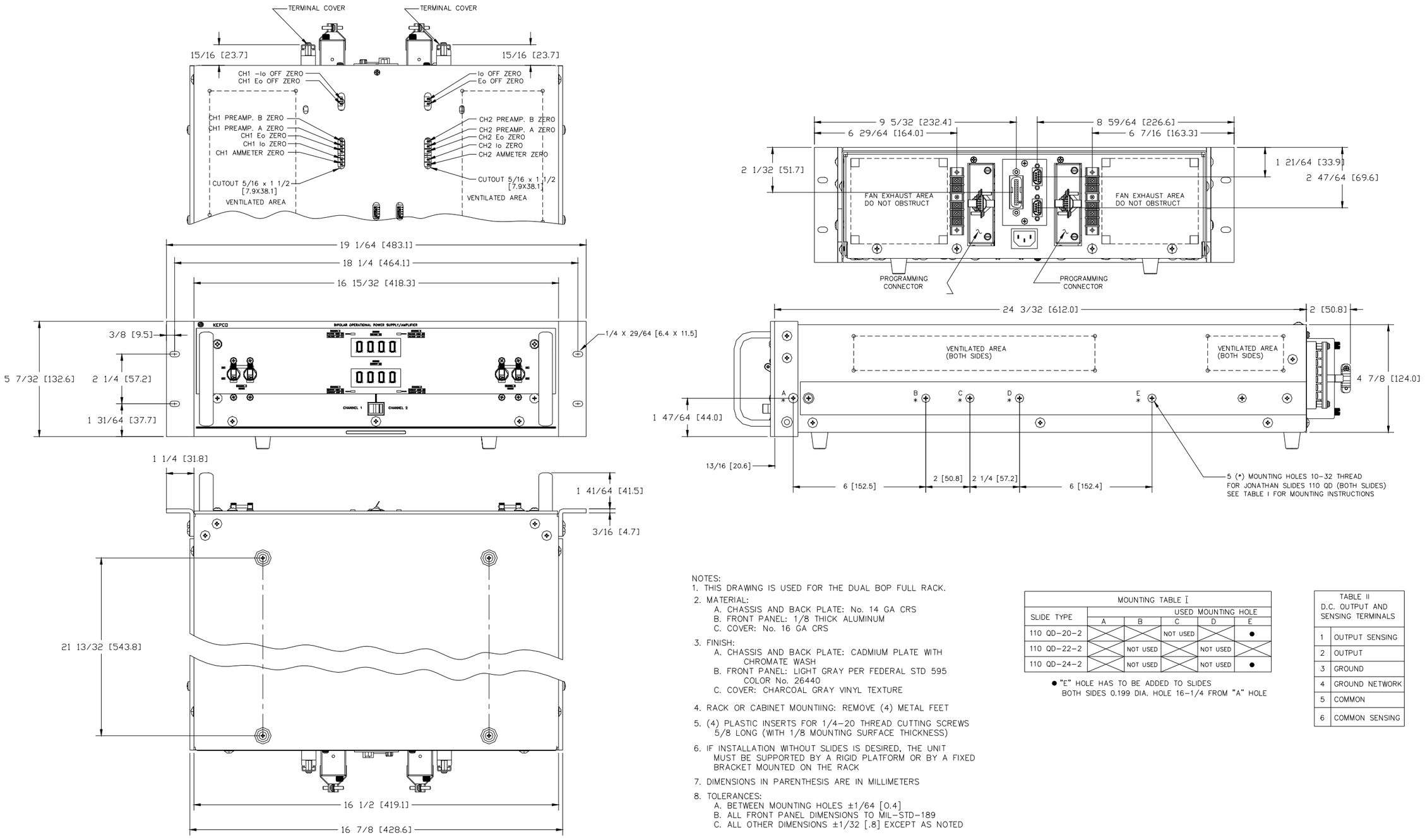
ITEM	FUNCTION	KEPCO PART NUMBER
Rear Programming Connector, Unwired	50-terminal printed circuit connector, unwired, for access to all rear programming connections and flag-signal outputs.	PC 12
Slides	Ease of access to adjustments through top cover	110 QD (Jonathan)
RS 232-to-USB Cable	Provides RS 232 connectivity to Windows computer via USB port. (one per channel required).	KIT 219-0630

1.9 SAFETY

Service must be referred to authorized personnel. Using the power supply in a manner not specified by Kepco. Inc. may impair the protection provided by the power supply. Observe all safety precautions noted throughout this manual. Table 1-8 lists symbols used on the power supply or in this manual where applicable.

TABLE 1-8. SAFETY SYMBOLS

SYMBOL	Meaning
	WARNING! RISK OF ELECTRIC SHOCK!
	CAUTION: REFER TO REFERENCED PROCEDURE.
WARNING	INDICATES THE POSSIBILITY OF BODILY INJURY OR DEATH.
CAUTION	INDICATES THE POSSIBILITY OF EQUIPMENT DAMAGE.



NOTES:
1. THIS DRAWING IS USED FOR THE DUAL BOP FULL RACK.
2. MATERIAL:
   A. CHASSIS AND BACK PLATE: No. 14 GA CRS
   B. FRONT PANEL: 1/8 THICK ALUMINUM
   C. COVER: No. 16 GA CRS
3. FINISH:
   A. CHASSIS AND BACK PLATE: CADMIUM PLATE WITH CHROMATE WASH
   B. FRONT PANEL: LIGHT GRAY PER FEDERAL STD 595 COLOR No. 26440
   C. COVER: CHARCOAL GRAY VINYL TEXTURE
4. RACK OR CABINET MOUNTIING: REMOVE (4) METAL FEET
5. (4) PLASTIC INSERTS FOR 1/4-20 THREAD CUTTING SCREWS 5/8 LONG (WITH 1/8 MOUNTING SURFACE THICKNESS)
6. IF INSTALLATION WITHOUT SLIDES IS DESIRED, THE UNIT MUST BE SUPPORTED BY A RIGID PLATFORM OR BY A FIXED BRACKET MOUNTED ON THE RACK
7. DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
8. TOLERANCES:
   A. BETWEEN MOUNTING HOLES ±1/64 [0.4]
   B. ALL FRONT PANEL DIMENSIONS TO MIL-STD-189
   C. ALL OTHER DIMENSIONS ±1/32 [.8] EXCEPT AS NOTED

MOUNTING TABLE I

SLIDE TYPE	USED MOUNTING HOLE				
	A	B	C	D	E
110 QD-20-2	X	X	NOT USED	X	●
110 QD-22-2	X	NOT USED	X	NOT USED	X
110 QD-24-2	X	NOT USED	X	NOT USED	●

● "E" HOLE HAS TO BE ADDED TO SLIDES BOTH SIDES 0.199 DIA. HOLE 16-1/4 FROM "A" HOLE

TABLE II
D.C. OUTPUT AND SENSING TERMINALS

1	OUTPUT SENSING
2	OUTPUT
3	GROUND
4	GROUND NETWORK
5	COMMON
6	COMMON SENSING

FIGURE 1-4. MECHANICAL OUTLINE DRAWING, (BOP 2X-DG SERIES MODELS)

SECTION 2 - INSTALLATION

2.1 UNPACKING AND INSPECTION.

This instrument has been thoroughly inspected and tested prior to packing and is ready for operation. After careful unpacking, inspect for shipping damage before attempting to operate. Perform the preliminary operational check as outlined in PAR. 2.6.1. If any indication of damage is found, file an immediate claim with the responsible transport service.

2.2 TERMINATIONS, CONTROLS AND INDICATORS

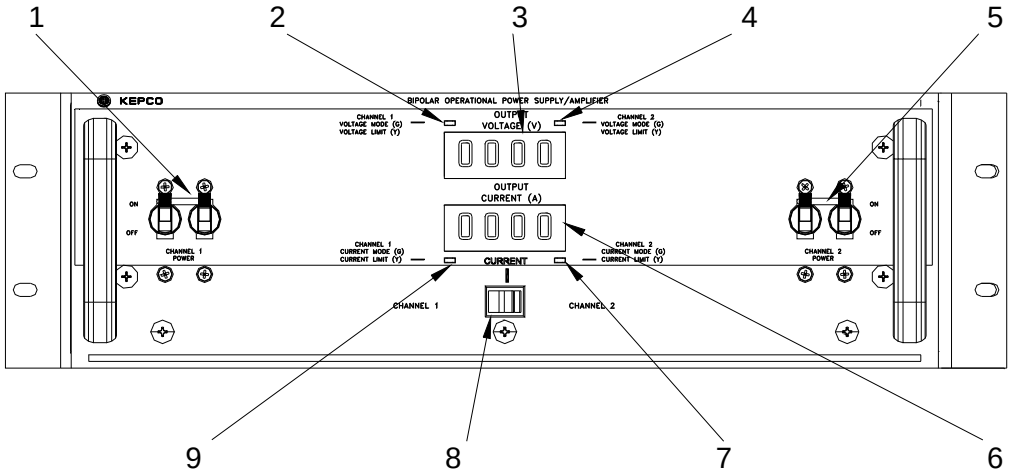
a) Front Panel: Refer to Figure 2-1. See Table 2-1 for an explanation of controls and indicators.

b) Rear Panel: Refer to Figure 2-2 and Table 2-2 for an explanation of terminations.

TABLE 2-1. BOP 2X-DG FRONT PANEL CONTROLS AND INDICATORS

FIG. 2-1 INDEX NO.	NAME OF TERMINATION OR CONTROL	FUNCTION
1	A-C POWER circuit breaker for Channel 1	A-C circuit breaker/power switch for Channel 1
2	VOLTAGE MODE/ VOLTAGE LIMIT indicator for Channel 1	Lights green when channel 1 is in voltage mode. Lights yellow when channel 1 is in voltage limit.
3	VOLTAGE meter	Digital output voltmeter displays d-c output voltage in Volts for Channel 1 or Channel 2 as determined by Channel 1/Channel 2 selector switch
4	VOLTAGE MODE/ VOLTAGE LIMIT indicator for Channel 2	Lights green when channel 2 is in voltage mode. Lights yellow when channel 2 is in voltage limit.
5	A-C POWER circuit breaker for Channel 2	A-C circuit breaker/power switch for Channel 2
6	CURRENT meter	Digital output current meter displays d-c output current in Amperes for Channel 1 or Channel 2 as determined by Channel 1/Channel 2 selector switch
7	CURRENT MODE/ CURRENT LIMIT indicator for Channel 2	Lights green when channel 2 is in current mode. Lights yellow when channel 2 is in current limit.
8	Channel 1/Channel 2 Meter select switch	Determines whether digital VOLT and CURRENT meters display output voltage and current, respectively, for Channel 1 or Channel 2.
9	CURRENT MODE/ CURRENT LIMIT indicator for Channel 1	Lights green when channel 1 is in current mode. Lights yellow when channel 1 is in current limit.

NOTE:
See Table 2-1 for functions.



3044255

FIGURE 2-1. BOP 2X-DG CONTROLS AND INDICATORS, FRONT PANEL

TABLE 2-2. BOP 2X-DG REAR PANEL TERMINATIONS AND CONTROLS

FIG. 2-2 INDEX NO.	NAME OF TERMINATION OR CONTROL	FUNCTION
1	REAR TERMINALS Channel 2	Rear barrier strip for channel 2 with output, output sensing, common, common sensing, grounding network and chassis ground terminals.
2	PC 15 for Channel 2	Rear Programming Connector for Channel 2. See Table 1-5 for details. NOTE: Unwired connector (PC 12) is available as an accessory (see Table 1-7).
3	GPIB Port	GPIB (IEEE 488) Port. Address 6 (default selects Channel 1., address 7 (default) selects channel 2. These addresses can be changed (see PAR. 2.5)
4	CH 1: BOP 2X-DG: RS 232 Port	RS 232 Port for Channel 1.
5	CH 2: BOP 2X-DG: RS 232 Port	RS 232 Port for Channel 2..
6	PC 15 for Channel 1	Rear Programming Connector for Channel 1. See Table 1-5 for details. NOTE: Unwired connector (PC 12) is available as an accessory (see Table 1-7).
7	REAR TERMINALS Channel 1	Rear barrier strip for channel 1 with output, output sensing, common, common sensing, grounding network and chassis ground terminals.
9	P201	A-C power input jack accepts 3-wire line cord (supplied) to supply input power to both channels 1 and 2.

NOTES:
1. See Table 2-2 for functions.

2. RS 232 port, pin 3, carries data from the Kepco power supply to the controller, pin 2 carries data from the controller to the Kepco power supply and pin 5 is the return for pins 2 and 3.

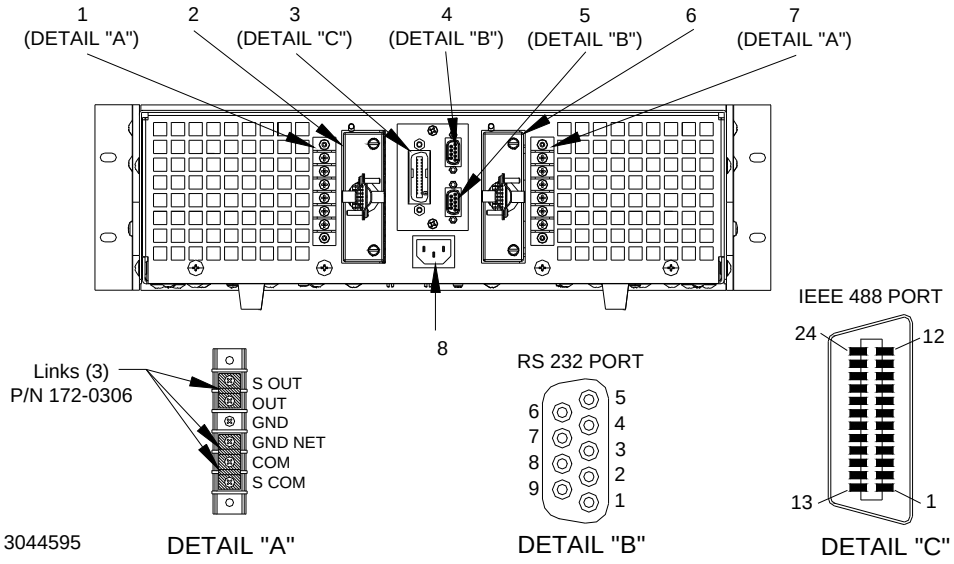


FIGURE 2-2. BOP 2X-DG TERMINATIONS AND CONTROLS, REAR PANEL

2.3 A-C INPUT REQUIREMENTS

Each channel is shipped wired for operation on a single phase, nominal 115V a-c line. For operation on other a-c source voltages, refer to FIG. 2-3. Determine your prevailing nominal source voltage and change the links on the barrier strip of main transformer T201 for both channel 1 and channel 2 according to the table provided in FIG. 2-3. The on-off circuit breakers for channels 1 and 2 remain equally effective at all input voltages.

CAUTION: Both channels of the BOP 2X-DG MUST be configured to use the same input voltage.

FOR AC SOURCE:	CONNECT LINKS BETWEEN TERMINALS:
104V	(3)—(4) and (7)—(8)
115V	(4)—(5) and (6)—(7)
208V	(2)—(3) and (8)—(9)
230V	(5)—(6)

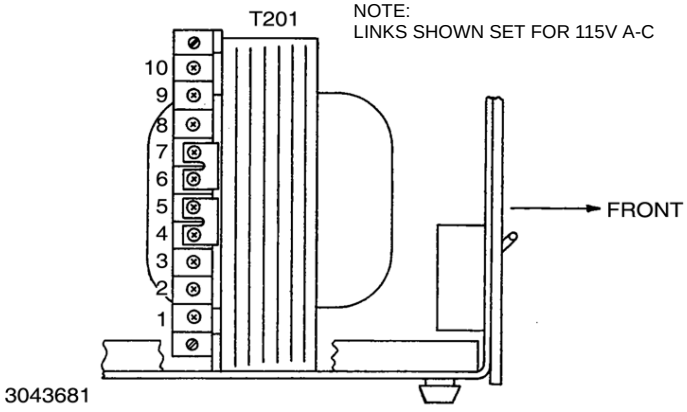


FIGURE 2-3. A-C SOURCE VOLTAGE SELECTION, EACH CHANNEL

2.4 COOLING

The components in the BOP power supply rely on forced air cooling for the maintenance of their operating temperature. Side panel openings and the top of the case must be kept clear from all obstructions to ensure air circulation. Periodic cleaning of the interior of the power supply is recommended. **If the BOP is rack-mounted or installed into confined spaces, care must be taken that the ambient temperature (the temperature immediately surrounding the power supply) does not rise above 55°C (~131°F).**

2.5 SET START-UP DEFAULT (SEE FIGURE 2-4)

The start-up default, consisting of the Device Address (GPIB address) is initially set by means of DIP switches as described in PAR. 2.5.1. The default power supply identification is described in PAR. 2.5.3.

2.5.1 SET GPIB (DEVICE) ADDRESS

The GPIB (Device) Address for channel 1 and 2 is set by means of DIP switches CH1S1 and CH2S1, respectively, positions 1 through 5, accessible through the top cover (see Figure 2-4). The Device Address is the permanent Listener or Talker address of the interface card on the GPIB. It is factory preset to address 6 for channel 1 and 7 for channel 2. If a different Device Address is required in your system, there are 30 (1-30) possible choices (See Table 2-3 and Figure 2-4).

The factory default GPIB address settings are:

Channel 1): 6
Channel 2): 7

NOTE: If both channel 1 and 2 DIP switches are set to the same address, GPIB communication will be disabled on both channels.

2.5.2 SET BAUD RATE FOR RS 232 INTERFACE

When using the RS 232 interface, the GPIB address (see PAR. 2.5.1) sets baud rate:

GPIB address 1 or 9 selects 19200,
GPIB address 3 or 11 selects 38400,
all other addresses select 9600 (default).

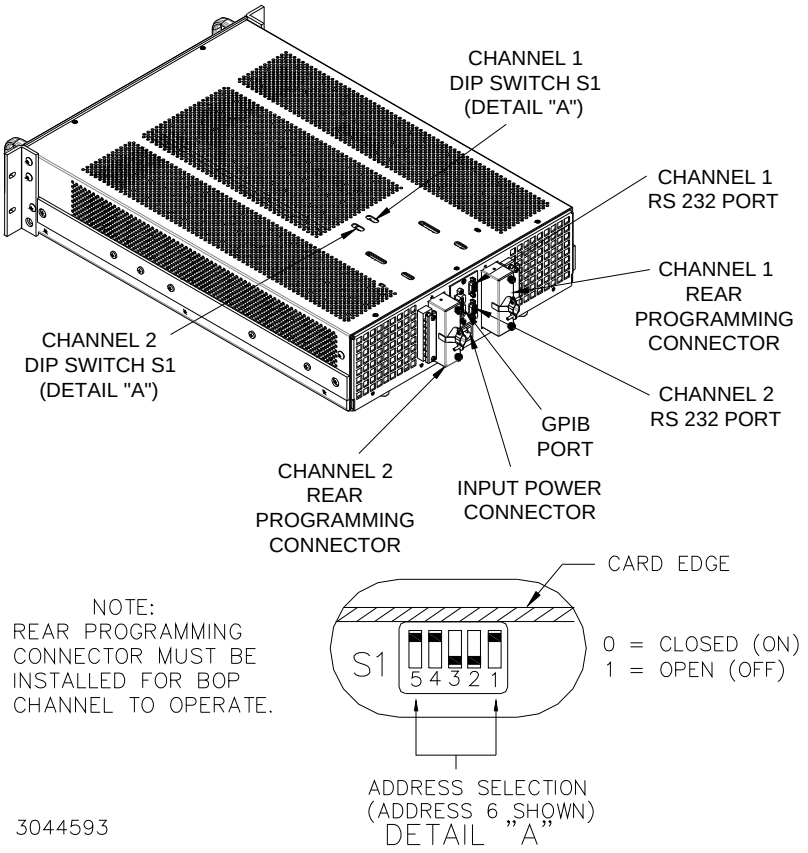


FIGURE 2-4. SETTING THE GPIB ADDRESS

TABLE 2-3. GPIB (DEVICE) ADDRESS SELECTION

DECIMAL ADDRESS	SELECTOR SWITCH S1 SECTION (SIGNAL LINE)				
	A5	A4	A3	A2	A1
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0

TABLE 2-3. GPIB (DEVICE) ADDRESS SELECTION (CONTINUED)

DECIMAL ADDRESS	SELECTOR SWITCH S1 SECTION (SIGNAL LINE)				
	A5	A4	A3	A2	A1
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0

NOTES:
1. 0 = CLOSED (ON) (Towards printed circuit board)
   1 = OPEN (OFF) (Away from printed circuit board)
2. GPIB address 31 (all 1's) is invalid and will cause the unit to beep 8 times to indicate non-volatile memory has been cleared.

2.5.3 POWER SUPPLY IDENTIFICATION

Each internal channel interface card has been factory set to report power supply voltage as $E_O$max (Volts) and current as $I_O$max (Amperes) for each model as listed in Table 1-3. The interface card has nonvolatile random access memory which is used to store these values. Other commands can be used to establish other operating characteristics such as the language and DCL operation (see PAR. 3.9).

2.6 PRELIMINARY CHECKOUT

CAUTION:The PC 15 connector must be connected for the channel to operate.

A simple operating check after unpacking, and before permanent installation, is advisable to ascertain whether the BOP has suffered damage in shipment. Refer to Figures 2-1 and 2-2 and Tables 2-1 and 2-2 for the location and function of the operating controls and terminals.

- Refer to PAR. 2.6.1 and perform the preliminary checkout using either the GPIB or RS 232 interface.

2.6.1 PRELIMINARY CHECKOUT USING GPIB OR RS 232 INTERFACE

Proceed as follows:

1. Install PC 15 rear panel programming connector at rear panel of channel under test (see Figure 2-2). This is required for proper operation of the channel under test.

2. Set the CHANNEL 1 - CHANNEL 2 selector switch to CHANNEL 1.

3. If using the GPIB interface, connect BOP to the GPIB bus using a standard GPIB cable connected to the 24-pin GPIB connector at the rear panel. Use GPIB address 6 to connect to channel 1, address 7 to connect to channel 2.

4. If using the RS 232 interface, at the RS 232 port corresponding to the channel under test proceed as follows:

   a. Use either a standard RS 232 cable to connect to any RS-232 computer port or use a standard RS 232-to-USB cable to connect to a USB port of a Windows computer. A USB adapter (Kepco P/N 118-1394) is provided as an accessory in KIT 219-0630 (1 per channel required) along with installation instructions. Note that the RJ45 patch cord and RJ45 to DB9 adapter mentioned in the KIT are inside the BOP 2X-DG series; ignore KIT installation instructions referring to these items. Instructions for KIT 219-0630 can be downloaded from the Kepco web site at:
   www.kepcopower.com\support\opmanls.htm#0630

   b. Set the desired baud rate using the GPIB address switch for channel under test: 1 or 9 selects 19200, 3 or 11 selects 38400, all other addresses select 9600 (default).

5. Refer to Safety instructions (PAR. 2.8) and connect BOP to 115V a-c source; see PAR 2.3 for different source voltage.

6. Apply power to BOP channel by setting corresponding AC POWER circuit breaker to ON. . The BOP channel will beep for less than 1 second, then will be ready for use.

7. Send the *IDN? query to channel under test via either the GPIB or RS 232 port.

8. For BOP 2X20-10DG channel 1 as an example, verify that the unit responds with KEPCO,BOP2X 20-10DG,E1233456A 11/10/98,1.81-1.81. Channel 2 responds with KEPCO,BOP2X 20-10DG,E1233456B 11/10/98,1.81-1.81.

9. Repeat steps 1 through 8 as needed, with connections to the other channel. Ensure that the PC 15 programming connector is installed for channel under test and, if using RS 232, the connection is made to the RS 232 port of the channel under test.

2.7 INSTALLATION OF BOP 2X-DG SERIES MODELS.

NOTE: For all installations into confined spaces, care must be taken that the temperature immediately surrounding the unit does not exceed the maximum specified ambient temperature (55° C).

Refer to the Mechanical Outline Drawing, Figure 1-4. BOP 2X-DG Series models are delivered with the Fixed Angle Brackets and the Chassis Side Support Bars installed. For bench operation, both of these components may be removed.

The BOP 2X-DG may be installed into standard (19-inch) equipment racks, or onto any other flat surface after removing the four (4) bottom aluminum feet. For rack installation of multiple units, 1U spacing is required between units.

If chassis slides are not used, the BOP 2X-DG may be installed onto any flat surface by means of the provided plastic mounting inserts on the bottom of the BOP 2X-DG chassis.

NOTE: If slide installation (as described below) is not desired, other means of supporting the unit in the rear must be provided (additional rear brackets or a solid platform, for example) since the front angle brackets alone are not sufficient to support the full weight of the unit.

2.7.1 SLIDE INSTALLATION.

The Chassis Slide Support Bars of the BOP 2X-DG are pre-drilled to accept Jonathan Series 110 QD slides. The slides listed in Table 2-4 can be accommodated.

TABLE 2-4. CHASSIS SLIDE MOUNTING HOLES

MODEL	KEPCO P/N	MOUNTING HOLES USED				
		A	B	C	D	E
110 QD-20-2	108-0067	X	X	O	X	DR
110 QD-22-2	108-0156	X	O	X	O	X
110 QD-24-2	108-0192	X	O	X	O	DR

KEY:X = Hole Used
O = Hole Not Used
DR = Hole must be drilled into chassis part of slide, each side, 0.199 in. dia., 16-1/4 in. from “A” hole.

To install the slides, refer to Figure 2-5 and proceed as follows:

1. Remove screws “A” through “E” from left and right chassis slide support bars. (Do not discard).

2. Determine the required hole-pattern for the selected slide model from Table 2-4.

3. Drill the “E” hole into both left and right chassis sections of slide if required.

4. Mount the chassis section of the slide to the left and right side of the BOP 2X-DG as shown in Figure 2-5. Use the same 10-32 x 1/2” mounting screws removed in step 1.

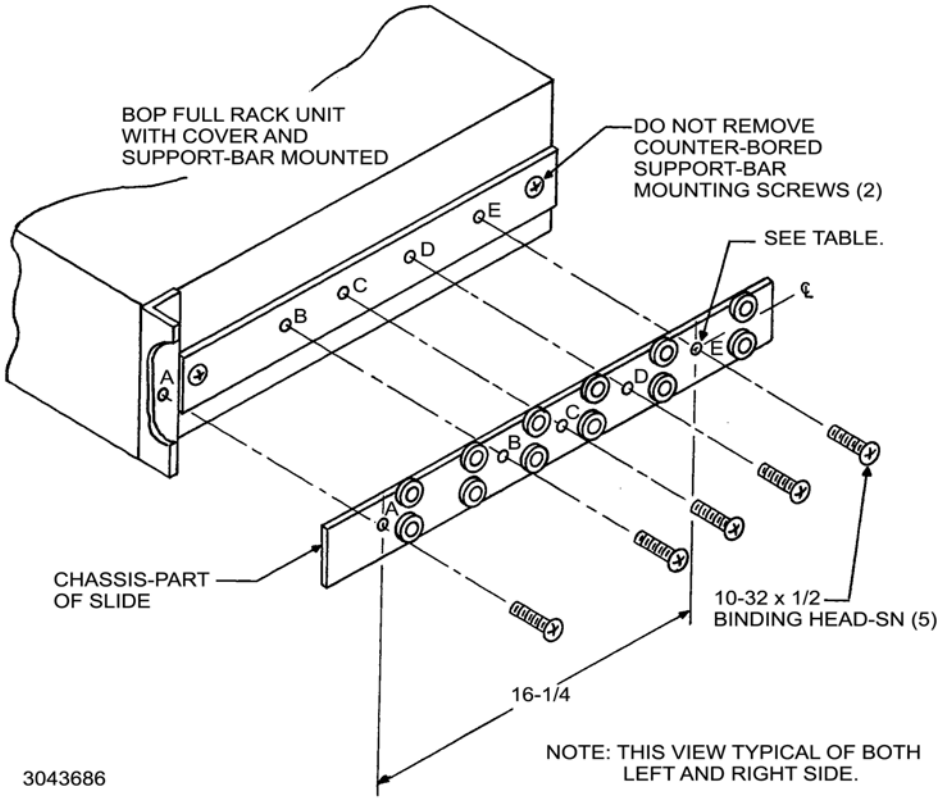


FIGURE 2-5. SLIDE INSTALLATION ON BOP 2X-DG UNITS

2.8 SAFETY PRECAUTIONS

Exercise care in making all connections to and from the BOP terminals.

WARNINGS

1. Remove a-c power from the BOP 2X-DG before making any connections.

2. Wires and/or cables, connected from the BOP terminals to external components or programming devices must be properly insulated and securely terminated on both sides to make accidental touch impossible.

3. The BOP 2X-DG chassis and cover must be safety-grounded to a reliable a-c source ground. A safety-ground may be established by using a grounded a-c power outlet or, if the latter is not available, by means of a separate wire, from the provided GND terminal of the rear panel terminal block to a reliable a-c source ground point.

4. THE COM OUTPUT TERMINALS OF THE BOP 2X-DG MAY EITHER FLOAT OR BE GROUNDED. Leave the output terminal floating when an externally grounded device is used at either the programming input or the output, or for a grounded load. For noise reduction ground a single return terminal on either the programming device, the BOP or the load configuration. See PAR. 2.9.4 for the maximum allowable floating voltage for the output terminals.

2.9 LOAD CONNECTION

Load connections to each BOP channel are achieved via the OUTPUT and COMMON terminals located on the rear panel.

2.9.1 LOAD CONNECTION, LOCAL SENSING

The basic interconnections between a BOP channel and the load using local sensing are shown in Figure 2-6.

The load wire size for the 2-wire connection shown should be rated for the nominal output current of the BOP channel and tied together to keep parasitic inductance low and to reduce possible “pick-up” from stray magnetic fields. The basic 2-wire connection is useful where the voltage drop in the load wires is small or of minor consequence, as for example, operation in constant current mode.

2.9.2 LOAD CONNECTION, REMOTE SENSING

The basic interconnections between each BOP channel and the load using remote sensing are shown in Figure 2-7. This is the recommended load connection for all applications requiring minimum load effect across a remote load. A twisted or shielded pair of wires (No. 22 AWG minimum) are connected from the BOP sensing

NOTE: The following explanation of AC and DC grounding can apply to any Kepco power supply even though this text applies specifically to BOP 2X-DG Series.

2.9.3 A-C GROUND

The metal cases of electronic equipment which are operated from the a-c power line must be kept at ground potential at all times. If the 3-wire line cord with 3-prong safety plug (supplied with this equipment) is used in combination with a properly grounded a-c power outlet, this is taken care of automatically. If a non-grounded outlet is used, the case must be earth-grounded separately. A separate GROUND terminal for each channel is provided for this purpose on the rear panel of the BOP 2X-DG.

The GROUND terminal of each channel is connected to chassis of the unit and to the local earth-ground potential through the A-C power line cord. It can be used a) as a grounding point for the output of the unit (COMMON terminal), and b) for doubling the a-c power cord earth-ground connection if necessary by providing a separate connection to the local earth-ground point.

CAUTION: Never connect a BOP output terminal (or the load terminal tied to the output terminal) to earth-ground. Otherwise, if the controlling device is grounded, the BOP channel can be damaged by the output limit current flowing inside the BOP along the programming signal return path.

CAUTION: Do not connect both the load and the programming device return (common) to earth-ground. Otherwise, if the common power connection between the BOP channel and the load is lost, the BOP can be damaged by output current flowing inside the BOP along the programming signal return path.

CAUTION: Do not use the programming return terminal as a tap point for the load return because the BOP can be damaged by output current flowing inside the BOP along the programming signal return path.

In conclusion, if the application needs to have its common point tied to earth ground, make the common terminal to earth-ground connection for only ONE of the following: a) BOP channel in use, b) load, c) a device connected to the BOP output, d) a programming device.

2.9.4 D-C GROUND

Specified ripple and noise figures listed in Table 1-1 for operational power supplies apply to each channel and are valid only with the COMMON side of the output/load circuit returned to an earth-ground point as shown in Figures 2-6 and 2-7.

CAUTION: If the programming device return is tied to earth-ground, leave the BOP output floating. Otherwise, instability and inaccuracy will occur.

If the application requires, the COMMON of the BOP may be floated up to 250V d-c (or peak) off ground. For this case, due to the common mode noise current flowing through a higher impedance, the resulting ripple/noise will be larger than published specifications. If the application must be grounded, connect a single return/common point of the application (including programming device, BOP and load) to earth-ground. The exact location of the best return earth-ground point must be carefully selected for minimum ripple/noise output.

The GROUNDING NETWORK terminal is tied to GROUND (chassis) terminal through a series capacitor-resistor network. Connecting the GROUNDING NETWORK terminal to the COMMON terminal reduces common noise current flowing through the load.

2.10 GPIB CONNECTIONS

To use the GPIB interface connect BOP channel to the GPIB bus using a standard GPIB cable connected to the 24-pin GPIB connector at the rear panel (see Figure 2-2, index 3). Use GPIB address 6 to connect to channel 1, address 7 to connect to channel 2. See PAR. 2.5.1 if necessary to change the GPIB address.

2.11 RS 232 CONNECTIONS

Use either a) a standard RS 232 cable to connect to any RS-232 computer port or b) a standard RS 232-to-USB cable to connect to a USB port of a Windows computer. A USB adapter (Kepco P/N 118-1394) is provided as an accessory in KIT 219-0630 (1 per channel required) along with installation instructions. Note that the RJ45 patch cord and RJ45 to DB9 adapter mentioned in the KIT are inside the BOP 2X-DG series; ignore KIT installation instructions referring to these items. Instructions for KIT 219-0630 can be downloaded from the Kepco web site at:

www.kepcopower.com\support\opmanls.htm#0630

Refer to PAR. 2.5.2 to use the GPIB address switches to set the baud rate for the channel to be used.

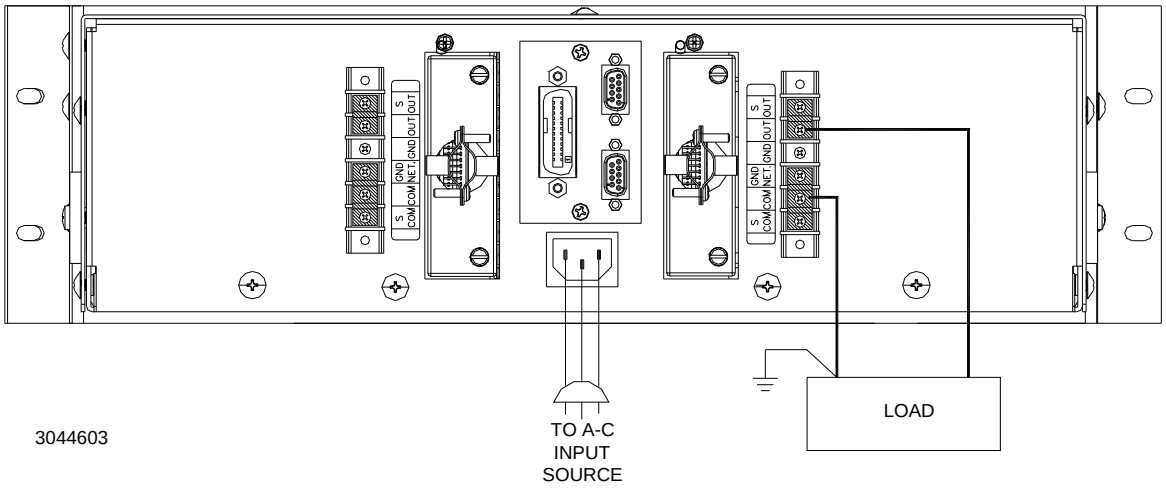
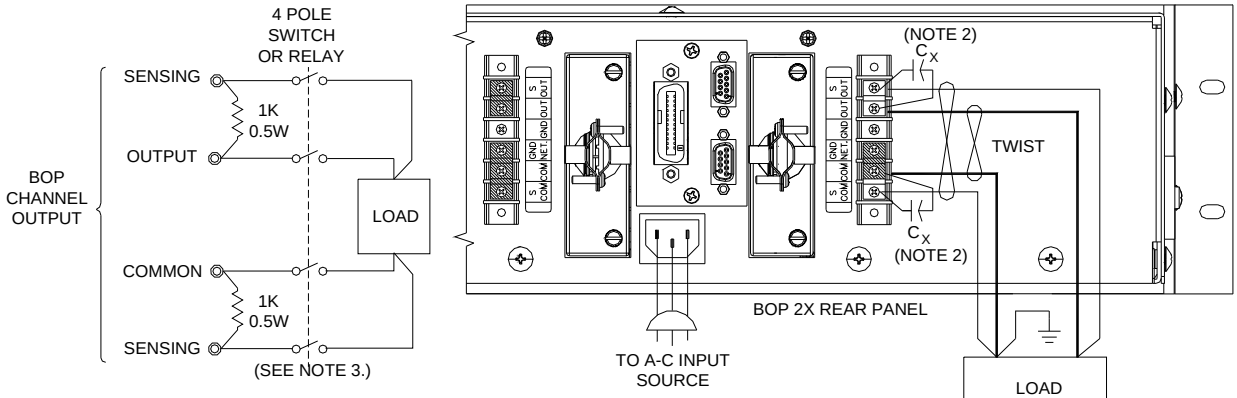


FIGURE 2-6. LOAD CONNECTION (CHANNEL 1, TYP.) USING LOCAL SENSING



NOTES:
1. Local sensing links between COM and COM S and between OUT and OUT S must be removed.
2. For applications requiring very long connections, additional external capacitors (C_X) should be connected as shown to compensate for the lead inductance of the connecting wires.
3. For applications requiring manually switched or relay driven output connections, external resistors must be connected as shown. All connections must be made and broken simultaneously.
4. Use separate properly rated twist pair wires for output power and output sensing connections.

3044604

FIGURE 2-7. LOAD CONNECTION USING REMOTE SENSING

SECTION 3 - OPERATION

3.1 INTRODUCTION

Each channel of the BOP 2X-DG Bipolar Operational Power Supply can be used in a great variety of applications. As a precision voltage or current source, the BOP output can be controlled remotely by either the built-in GPIB or RS 232 interface or by means of external voltage signals. Independently adjustable, remotely programmable limit circuits for both output voltage and output current protect a sensitive load from any overvoltage or overcurrent.

NOTE: Before using the BOP in any application, please refer to Section 2 to get acquainted with the operating controls, a-c power requirements and information on load connections and grounding. Front panel components identified by index numbers in parentheses are shown in Figure 2-1.

3.2 POWER SUPPLY BASICS

CAUTION: DO NOT repeatedly toggle the circuit breaker/switch as this may cause unit to fault. To turn a channel on, set applicable Power ON/OFF circuit breaker/switch on front panel to ON. If actuator does not lock when released, wait a few seconds before trying again. The circuit breaker is “trip-free” design; if overload exists, contacts cannot be held closed by actuator.

NOTE: Upon power up, the BOP channel output is set to off, VOLTage and CURRent are set to 0. VOLTage and CURRent commands will not change the BOP channel output until the command OUTPut ON is sent to the BOP channel in use.

3.2.0.1 FACTORY DEFAULTS

The factory default GPIB address settings are listed below

- Channel 1): 6 (see PAR. 2.5.1 to change it if needed).
- Channel 2): 7 (see PAR. 2.5.1 to change it if needed).

Note that when using the RS 232 interface, the GPIB address controls the baud rate:

- 1 or 9 selects 19200,
- 3 or 11 selects 38400,
- all other addresses select 9600 (default).

3.2.1 ANALOG CONTROL

As a Bipolar Amplifier, each BOP channel responds to such input signals as sine, square or triangular waves. A ±10 volt input signal will program the BOP output (voltage or current) through its rated output ranges. Built-in preamplifiers, for the voltage, as well as the current loop of the BOP, permit amplification of the control signals to the required amplitude and provide the interface with high, as well as low, impedance signal sources.

Detailed examples of the more popular applications are described in the following paragraphs. Before actual operation, however, the following safety precautions must be carefully considered.

WARNING

Exercise care in making all connections to and from the BOP terminals. Remove a-c power from the BOP 2X-D before making any connections.

Wires and/or cables connected from the BOP terminals to external components or programming devices must be properly insulated and securely terminated on both sides to make accidental touch impossible.

The BOP chassis must be safety-grounded to a reliable a-c source earth-ground. A safety-ground may be established by using a grounded a-c power outlet or, if the latter is not available, by means of a separate wire from the provided GROUND terminal for each channel to a reliable a-c source earth-ground point.

The COMMON output terminal of a BOP channel can be either grounded or floating, depending on the application. Additional precautions must be taken to make any user access to the output terminals impossible.

A BOP channel must always have sensing configured between OUT and OUT S and between COM and COM S. For voltage mode choose either local or remote sensing as desired. For current mode use local sensing.

Application and test set up diagrams on the following pages show the symbolic and simplified representation of four basic programmable circuits of a BOP channel:

- Voltage Control Loop: see Figure 3-1 and PAR. 3.3.
- Current Limit Circuits: See Figure 3-2 and PAR. 3.3.5.
- Current Control Loop: see Figure 3-3 and PAR. 3.4.
- Voltage Limit Circuits: see Figure 3-4 and PAR. 3.4.5.

The complete simplified diagram is shown in Figure 3-31 at the end of this section. The four circuits can be programmed separately as described in the examples in this section. Of these four circuits, only three can be used at one time, voltage loop with current limit and voltage limit loops or current loop with current limit and voltage limit loops. However, most of the time either voltage loop current limit loop or current loop with voltage limit loop are being exercised. Figures 3-31 and 3-32 also show the complete wiring configuration of the unmodified (default) PC 15 Rear Programming Connector.

The main control loop (either voltage or current) can be selected by using pin 30 of the Rear Programming Connector, however do not use pin 30 for voltage/current mode control if an optional BIT card for digital control of the channel is installed.

- Voltage mode = open or TTL “1” (high)

- Current mode = short (referenced to sensing common, pin C) or TTL “0 (low)

3.2.2 PC 15 REAR PROGRAMMING CONNECTOR

The associated PC 15 Rear Programming Connector MUST be installed in order for a BOP channel to function. The numbered pins in the simplified diagrams of this section correspond to the pins of the Rear Programming Connector PC 15. Modifications to the existing PC 15 connections are noted on the simplified diagrams for the application examples. The pin designated C (Common) corresponds to pin numbers 23, 25, 27, 29, 31 or 33 on the PC 15 Connector. See Figure 3-32 at the end of this section for additional details regarding the PC 15 rear programming connector.

Although schematically and functionally identical, there are some small differences in the physical connections for the two PC 15 versions, P/N 195-0039 which uses wire jumpers and P/N 195-0121 which uses printed circuit traces. Connections shown on the diagrams in this section are for P/N 195-0121. Wire jumpers may be used to make needed connections for both PC 15 versions.

- **Pins 1, 8, and 34:** wire jumpers use pin 1 as a tie point, with wire jumpers to pins 8 and 34. Printed circuit traces use pin 34 as a tie point, with printed circuit trace connections to pins 8 and 1.

- **Pins 3, 6, and 32:** wire jumpers use pin 32 as a tie point, with wire jumpers to pins 3 and 6. Printed circuit traces use pin 3 as a tie point, with printed circuit trace connections to pins 32 and 6.

3.2.3 EXTERNAL PROGRAMMING RESISTORS.

External programming resistors should be high-quality units, with low-temperature coefficients. Resistors should be selected carefully, since in most applications, the limitations for stability and drift are due not to the BOP, but to the programming resistors. Selection criteria for resistors are:

- Temperature coefficient

- Leakage (in values above 100k ohm)

- Humidity effects

- Drift with time

- Self-heating (power dissipation)

- Tolerance.

NOTE: When an external 10K Ohm resistor is specified, it should be 0.001% tolerance to meet published specifications for the power supply channel.

For variable resistors (potentiometers or rheostats) similar selection criteria apply. In addition, such specifications as listed below should be carefully considered if the application requires:

- End resistance
- Linearity
- Capacitive and inductive parasitic effects

3.2.4 ACTIVE PROGRAMMING SOURCES

External programming sources (Signal Generators, etc.) or reference sources should have temperature coefficients and drift specifications comparable to (or better than) the BOP 2X-D Power Supply.

CAUTION: A-C source-operated programming sources must have their output isolated from the case.

3.2.5 EXTERNAL LEADS

Shielded (preferably twisted) lead pairs are recommended for all input connections to the BOP control loops of each channel. The shield should be connected (single-ended) to the chosen signal ground point. Shielded leads should be held as short as practicable. Output leads must be voltage-rated for at least the maximum BOP output voltage and current. Twisted pair wires are recommended for output power and output sense wiring.

3.2.6 EXTERNAL VOLTAGE MONITOR

Use Voltage Monitoring signal at pin 20 of the Rear Programming Connector. The range of this signal is 0 to ±10V equivalent to output voltage of 0 to $\pm E_O$max. The Voltage Monitoring signal is buffered by an op-amp. This stage is protected against short-circuit by the intrinsic protection of the op-amp. The return for the Voltage Monitoring signal is referred to as pin C, and can be any one of the signal return (signal common) pins (23, 25, 27, 29, 31 or 33) of the rear programming connector.

3.2.7 EXTERNAL CURRENT MONITOR

Use the Current Monitoring signal at pin 10 of the Rear Programming Connector for external current monitoring. The range of this signal is 0 to ±10V, equivalent to output current of 0 to $\pm I_O$max. An op-amp buffers the Current Feedback/Sensing signal in order to create the Current Monitoring signal. This stage is protected against short-circuit by the intrinsic protection of the op-amp generating this signal. The return for the Current Monitoring signal is referred to as pin C, and can be any one of the signal return (signal common) pins (23, 25, 27, 29, 31 and 33) of the rear programming connector.

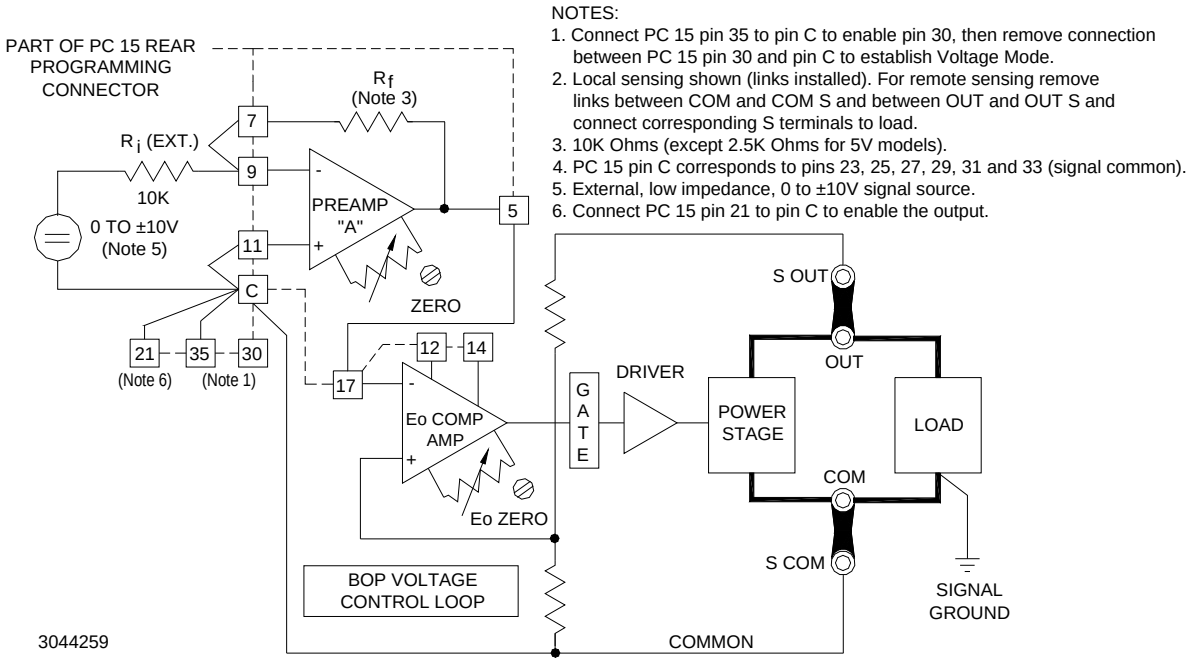


FIGURE 3-1. BOP BASIC VOLTAGE CONTROL LOOP (LOCAL SENSING)

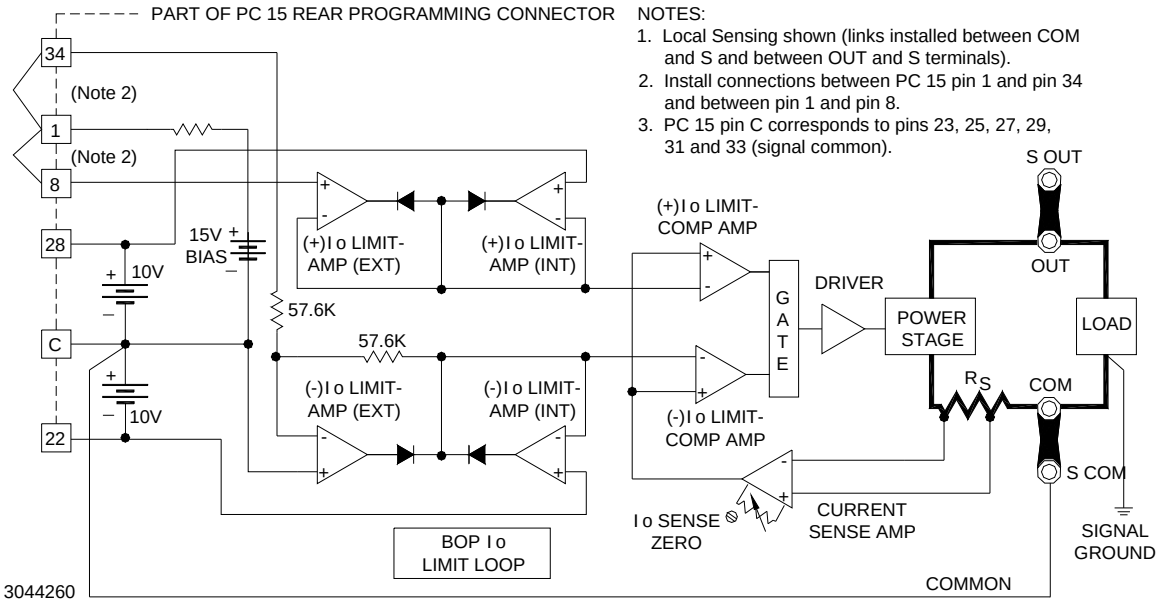


FIGURE 3-2. BOP BASIC (±) CURRENT LIMITING CIRCUIT (LOCAL SENSING)

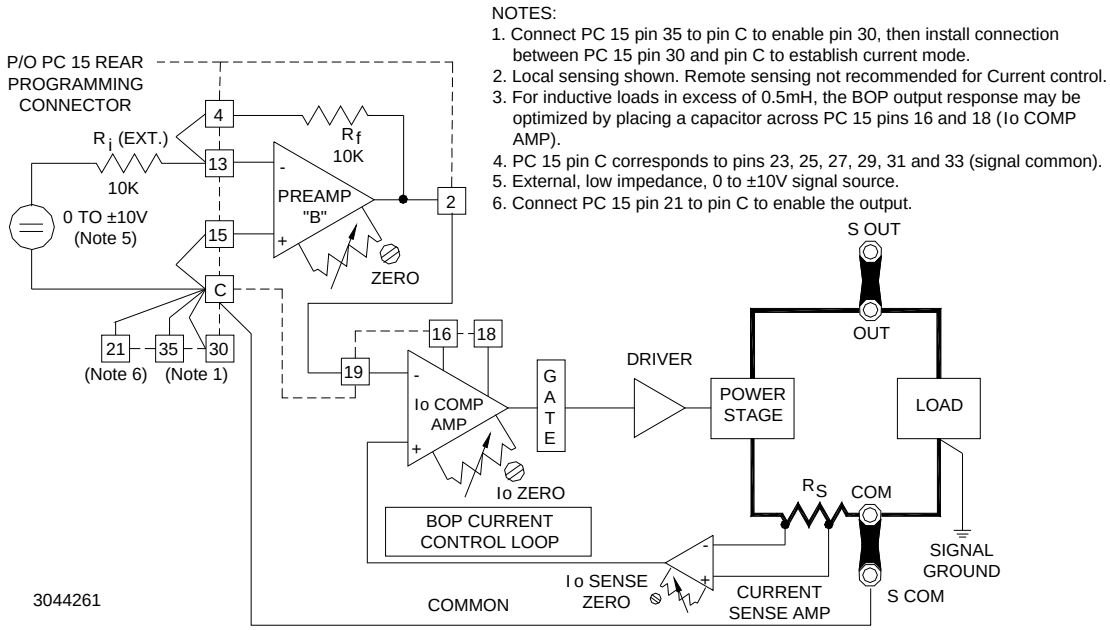


FIGURE 3-3. BOP BASIC CURRENT CONTROL (LOCAL SENSING)

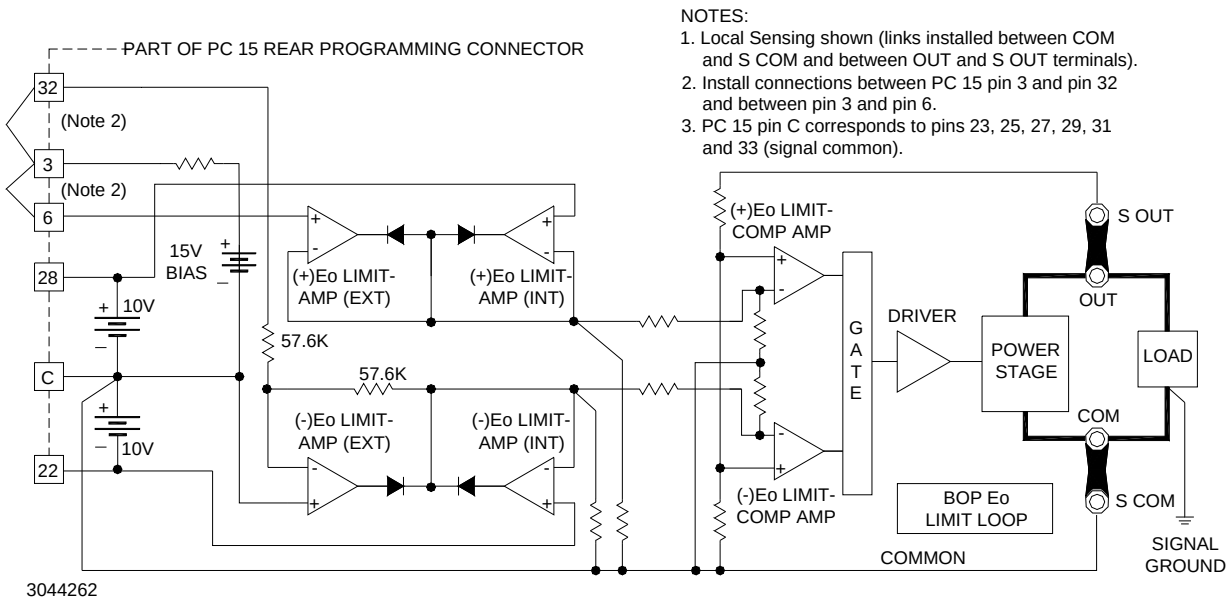


FIGURE 3-4. BOP BASIC (±) VOLTAGE LIMITING CIRCUIT (LOCAL SENSING)

3.3 REMOTE ANALOG CONTROL OF VOLTAGE LOOP

Control of the voltage loop is accomplished by using an external configuration to apply control voltage(s) to the applicable pins of the PC 15 Rear Programming Connector.

3.3.1 REMOTE VOLTAGE CONTROL USING RESISTANCE (POTENTIOMETER)

When configured as shown in Figure 3-5, Preamp “A” with the built-in feedback and input resistors has a gain of one and the fixed gain of all BOP models is designed so that a d-c input signal of zero to (±)10 volts will control the BOP output through its full positive and negative range. This is accomplished be the external Remote Voltage Control potentiometer shown in Figure 3-5.

With no additional external wiring ±current limiting is automatically set to 110% of the nominal BOP output current. Full user control of current limits is accomplished using the external configurations described in PAR. 3.3.5.

NOTE: During normal operation in Voltage mode, The front panel Voltage Mode/Voltage Limit indicator lights green, and the Current Mode/Current Limit indicator is off. If the output current exceeds the preadjusted value at any time, the Voltage Mode/Voltage Limit indicator will go off and the Current Mode/Current Limit indicator will light yellow. After the cause of overcurrent is eliminated, the indicators will return to their initial status.

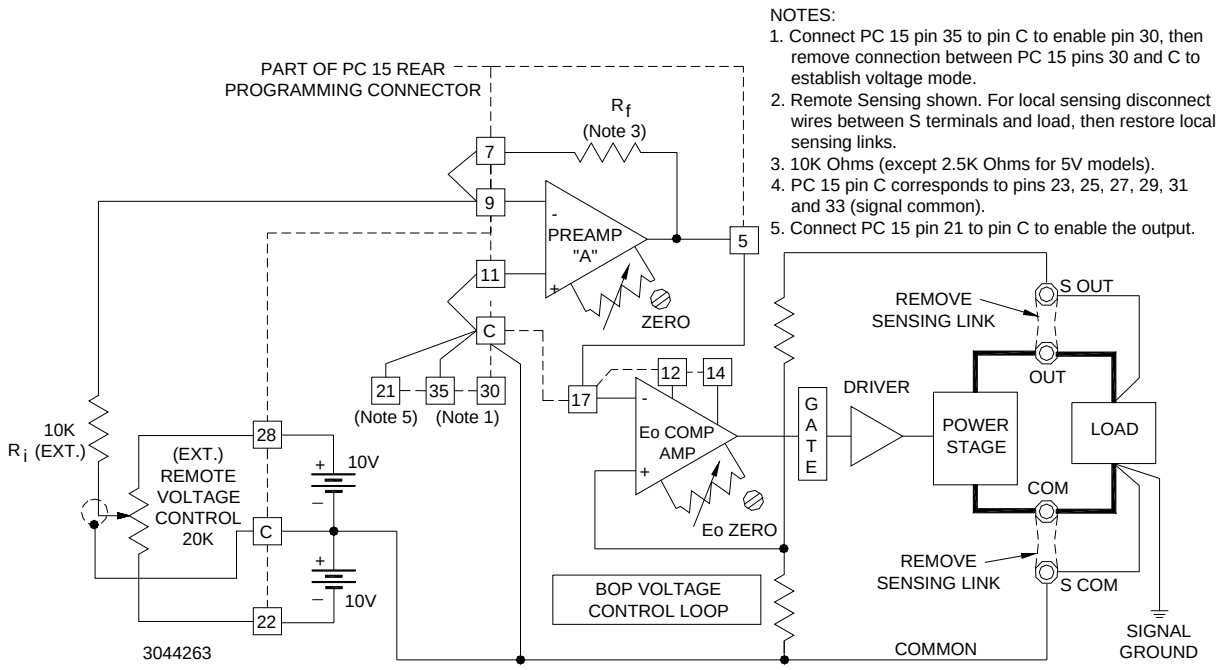


FIGURE 3-5. REMOTE POTENTIOMETER CONTROL OF D-C OUTPUT VOLTAGE

3.3.2 REMOTE VOLTAGE CONTROL USING RESISTANCE (DECADE BOX)

Two-terminal resistance control of the d-c voltage (for example, by means of a Decade Box (make-before-break type) or any variable resistance) can be exercised as shown in Figure 3-6.

The external switch (S1), connected across the (±) reference voltage, is used to pre-select either positive or negative output by setting it to the applicable reference voltage polarity.

PREAMP “A” functions here as an inverter. Its output can be expressed by the equation:

$E_{O\ (PREAMP\ “A”)} = -E_{ref}\ (R_f/R_i)$. where

R_f = Decade Resistor, 0 - 10K Ohms (except 0 - 2.5K Ohms for 5V models)
R_i = Ext. Input Resistor

The BOP output voltage will vary linearly with the change in decade resistance.

This programming system can be calibrated very accurately by setting the external Remote Voltage Control (Decade) to zero, and adjusting for zero output volts using the screwdriver-adjusted E_O ZERO control while monitoring the BOP output with a digital voltmeter. The maximum output voltage is calibrated by making all or part of R_i a rheostat and adjusting it with the Remote Voltage Control at maximum resistance, to the maximum BOP output voltage.

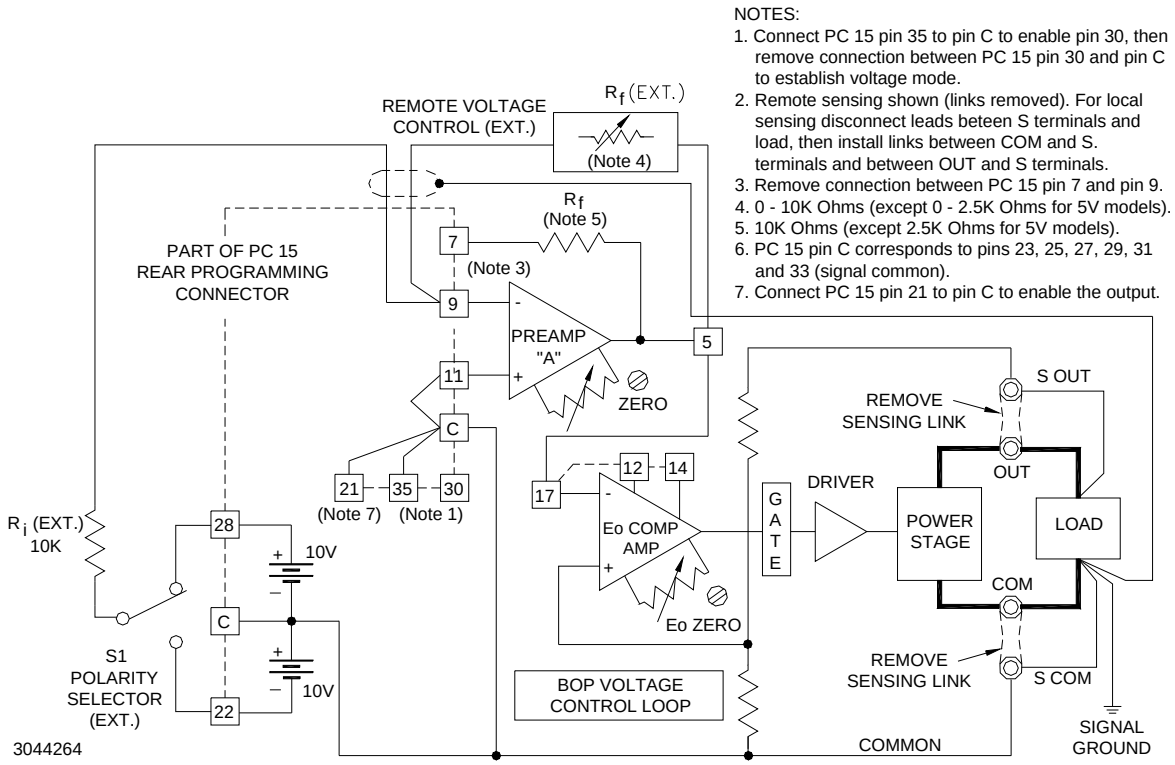


FIGURE 3-6. REMOTE D-C OUTPUT VOLTAGE CONTROL USING TWO-TERMINAL RESISTANCE (DECADE)

3.3.3 REMOTE VOLTAGE CONTROL USING D-C SIGNAL VOLTAGE

The BOP d-c output voltage can be controlled directly by a ±10V d-c signal, applied to the rear programming connector pin 9 referenced to common. An interesting example of how this can be implemented is the use of a Kepco SN Digital Programmer to convert digital input to the analog ±10V d-c signal. The digital input to the SN programmer can be provided by a computer, or manually by means of a keyboard. The analog output from the SN is from zero to ± 10 volts and constitutes the input programming voltage for the BOP.

The connections between the SN Programmer and the BOP are illustrated in Figure 3-7. This programming system provides a resolution of 12-bits with a linearity of (±)1/2 LSB (0.0122%). The digital and the power supply grounds are isolated (optical isolation) to 1000 volts. Calibration of the system is performed with the provided controls on the SN Programmer.

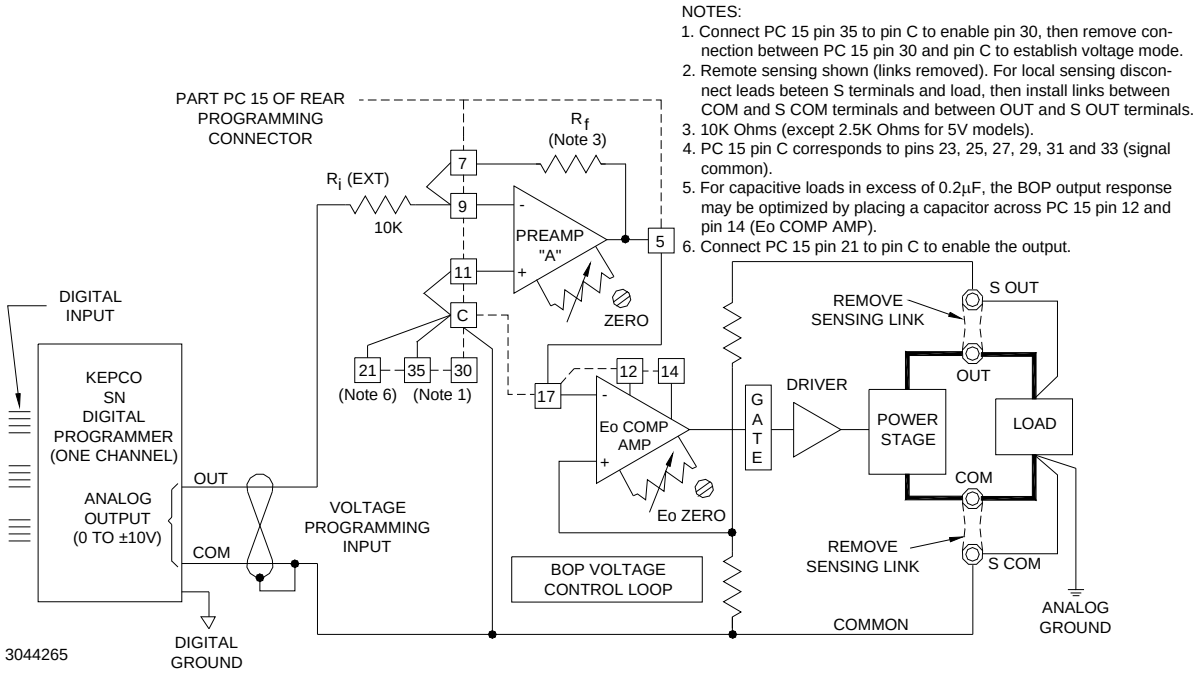


FIGURE 3-7. EXTERNAL DIGITAL CONTROL OF THE D-C OUTPUT VOLTAGE

Since all terminals of Preamp “A” are available at the Rear Programming Connector, d-c output voltage control can be exercised in many other ways. Preamp “A” can be treated as an uncommitted operational amplifier, with its applicable transfer functions. Its initial offsets can be zeroed with the provided control. The basic principle of d-c output voltage control of the BOP is that an Input control voltage of (±)10 volts able to deliver 1mA applied to rear programming connector pin 9 (through an external 10k Ohm resistor), referenced to common, will produce the full d-c output voltage swing of the BOP. If the selected control signal does not have the required amplitude or if the required control current cannot be supplied, the Preamp “A” configuration can be altered to provide the proper interface for the application.

For example, if the BOP d-c output is to be controlled by means of a bipolar, 1 volt, high impedance source, it can be connected to the BOP as shown in Figure 3-8.

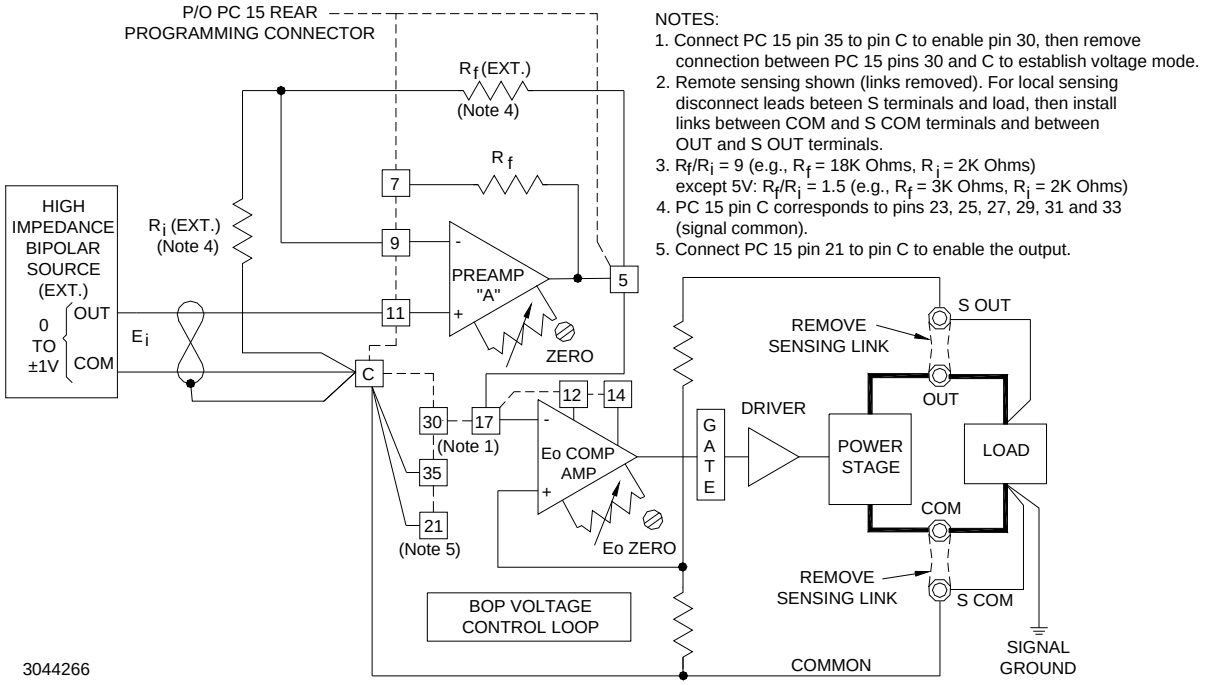


FIGURE 3-8. D-C OUTPUT VOLTAGE CONTROL WITH NON-INVERTING CONFIGURATION AND HIGH IMPEDANCE ±1V SIGNAL SOURCE

As seen in Figure 3-8, the original input feedback components of Preamp “A” have been bypassed and are replaced by external resistors (metal film or wire-wound type resistors may be used and connected directly to the Rear Programming Connector terminals as shown). Preamp “A” is used in the non-inverting configuration, so the output voltage will have the same polarity as the input programming signal. Its output voltage (the necessary BOP control voltage) as shown in the examples below is given by:

$$E_{O\ (PREAMP\ “A”)} = E_i \frac{R_f + R_i}{R_i}$$

all models except 5V: $E_{O\ (PREAMP\ “A”)} = (\pm 1V)\frac{18KOhms + 2KOhms}{2KOhms} = \pm 10V$

$E_{O(unit)} = (\pm 10V)\ (G) = \pm E_O nom\ (V)$
where G = Closed loop Gain V/V(see Table 1-3)

5V models: $E_{O\ (PREAMP\ “A”)} = (\pm 1V)\frac{3KOhms + 2KOhms}{2KOhms} = \pm 2.5V$

$E_{O(unit)} = (\pm 2.5V)\ (2V/V) = \pm 5V$

Consequently, as the bipolar input voltage (E_i) is controlled from –1V through zero to +1V, the BOP output voltage will follow over its specified range. Calibration of the system can be performed at the programming source, or the Preamp “A” ZERO and E_O ZERO controls can be used for the zero adjustment, while the feedback resistor (R_f) can be trimmed to calibrate full scale.

3.3.4 USING THE BOP AS A VOLTAGE AMPLIFIER

If, instead of the d-c control signal (see PAR. 3.3.3), an a-c signal voltage is applied to the Voltage Programming Input terminals, the BOP functions as a bipolar amplifier. As an amplifier, the BOP voltage gain is set on each BOP model so that (as with the d-c control signal) a bipolar a-c input signal with an amplitude of 0V to 20V (peak-to-peak) will drive the BOP output through its specified (±) output voltage range. All other dynamic specifications are given in Table 1-4.

The BOP can be used to amplify, sum or scale a variety of waveshapes, some of which are illustrated in Figure 3-9. All input signals are shown 180° out of phase with their corresponding output waveshapes. The latter can be readily produced in phase (i.e. with the output wave taking the same direction as the input signal) by addressing the non-inverting input of Preamp “A.”

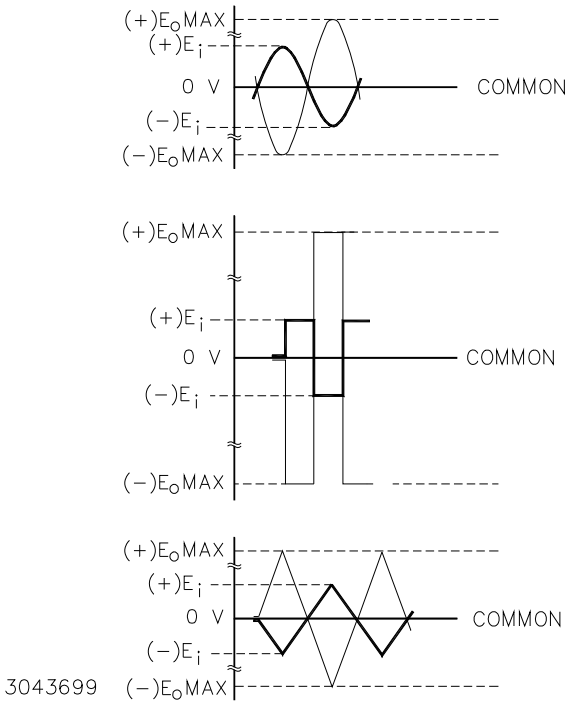


FIGURE 3-9. GRAPHS OF POSSIBLE INPUT/OUTPUT WAVESHAPES

The basic programming circuit used to produce the waveforms shown in Figure 3-9 is shown in Figure 3-10. Since all input signals of Figure 3-9 are shown with a ±10 volts amplitude, the basic programming circuit in Figure 3-10 must be modified if the external signal source cannot produce 10 volts and if the full BOP output voltage swing is required.

If the external programming source does not have sufficient amplitude to drive the BOP over its full output range, the gain of Preamp “A” must be changed from the built-in value to suit the application. To calculate the required components for the new gain requirement, the output equation for Preamp “A” in the inverting configuration is used:

$E_{O\ (PREAMP\ “A”)} = -E_i\ (R_f/R_i)$

where E_O (Preamp "A") = ±10V (±2.5V for 5V models), and the values of R_f and R_i depend on the available amplitude of the programming source. If, for example, a ±1 volt source able to deliver 0.1mA is available, the ratio R_f/R_i must be 10 (2.5 for 5V models), and the two resistor values can be R_i = 10K and R_f = 100K (R_i = 10K and R_f = 25K for 5V models) respectively. R_i = 10K can be retained, and only R_f must be replaced with a 100K (25K for 5V models) metal film, 1/2 watt resistor. The necessary connections are illustrated in Figure 3-11. Gain control can be exercised by making R_f a rheostat instead of a fixed resistor.

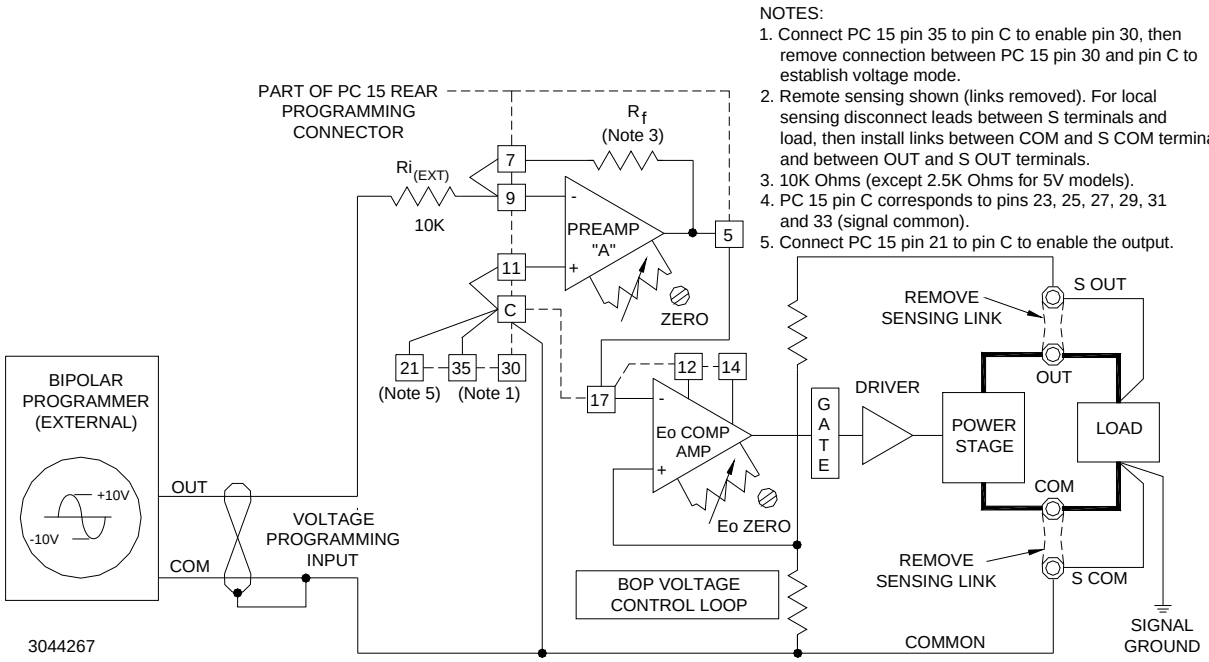


FIGURE 3-10. BASIC PROGRAMMING CIRCUIT FOR USE OF BOP AS BIPOLAR AMPLIFIER (VOLTAGE MODE)

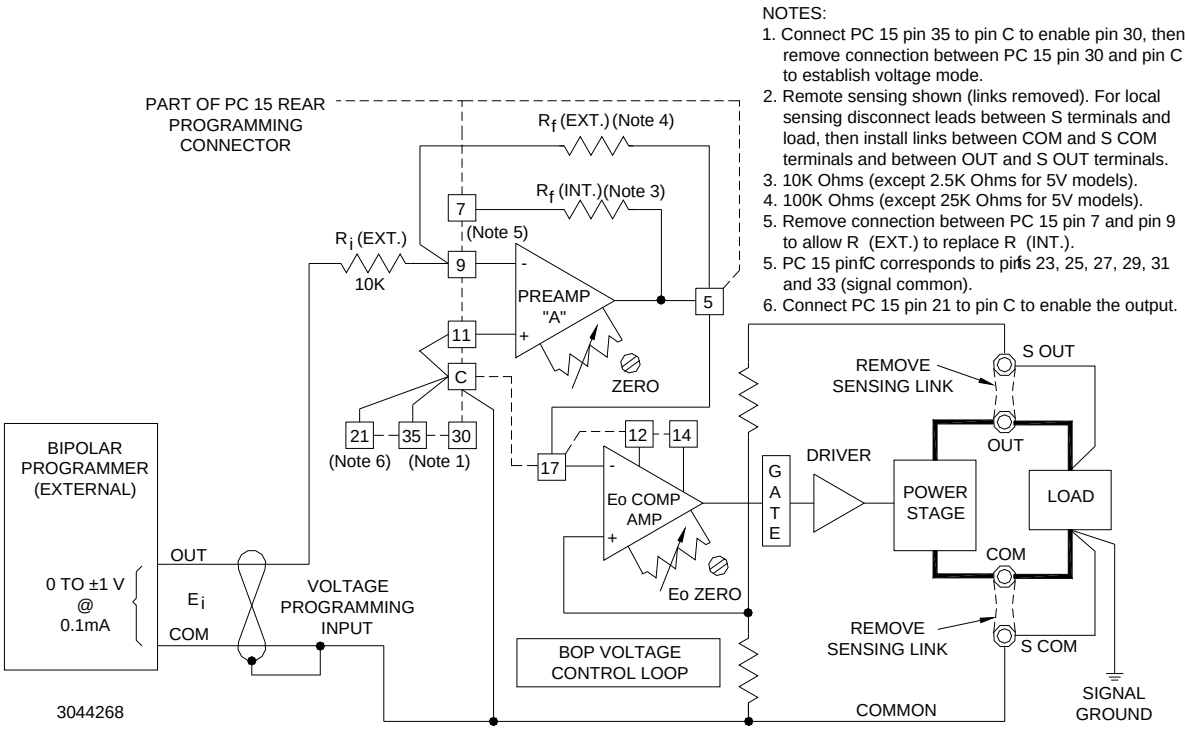


FIGURE 3-11. PROGRAMMING CIRCUIT FOR DRIVING OUTPUT VOLTAGE WITH ±1V INPUT PROGRAMMING VOLTAGE

3.3.5 REMOTE ANALOG CONTROL OF CURRENT LIMITING

For remote control of current limiting, the ± I_O Limit Amplifiers (EXT) are disconnected from the 15V Bias source by removing connections between pin 1 and pin 34 and between pin 34 and pin 8 at the Rear Programming Connector (see Figure 3-12). An external 0 to +10V control signal is applied to pins 8 and 34, with reference to common. This control signal programs both the negative and positive current limit simultaneously and symmetrically over the BOP output current range.

If individual control of the ± current limits is desired, the jumper between pin 8 and pin 34 at the Rear Programming Connector must be removed and separate control sources must be connected: 0 to +10 volts to pin 8 for the + I_O limit, and 0 to +10 volts to pin 34 for the – I_O limit, both with reference to common (see Figure 3-13).

NOTE: If a control voltage is not applied to pins 8 and 34 of the Rear Programming Connector, the ± I_O LIMIT value is automatically set to 110% of the rated nominal current of the BOP. This is a backup of the external programming of current limits described above.

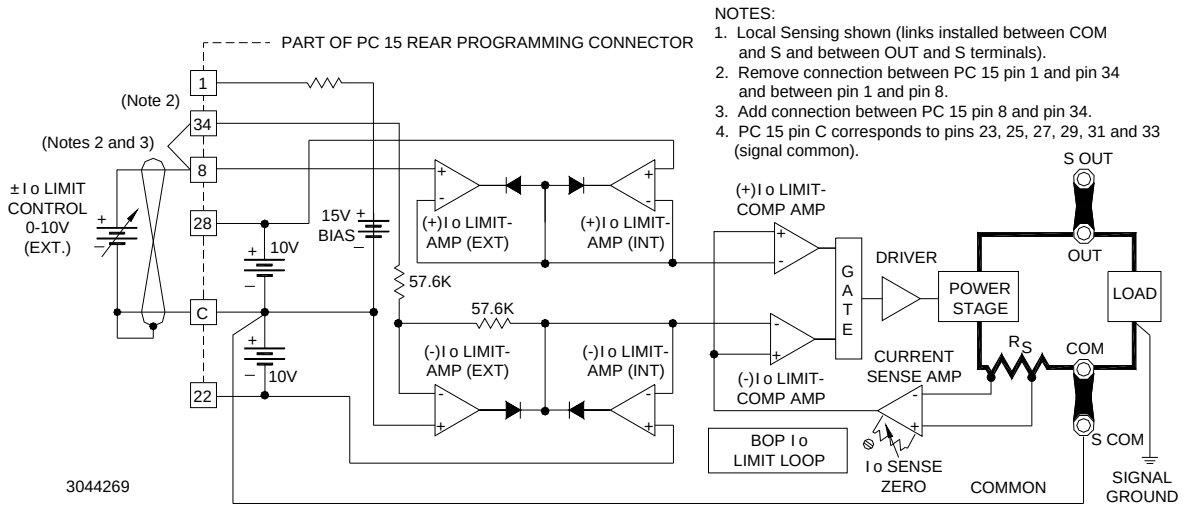


FIGURE 3-12. SYMMETRICAL REMOTE CONTROL OF THE CURRENT LIMITS

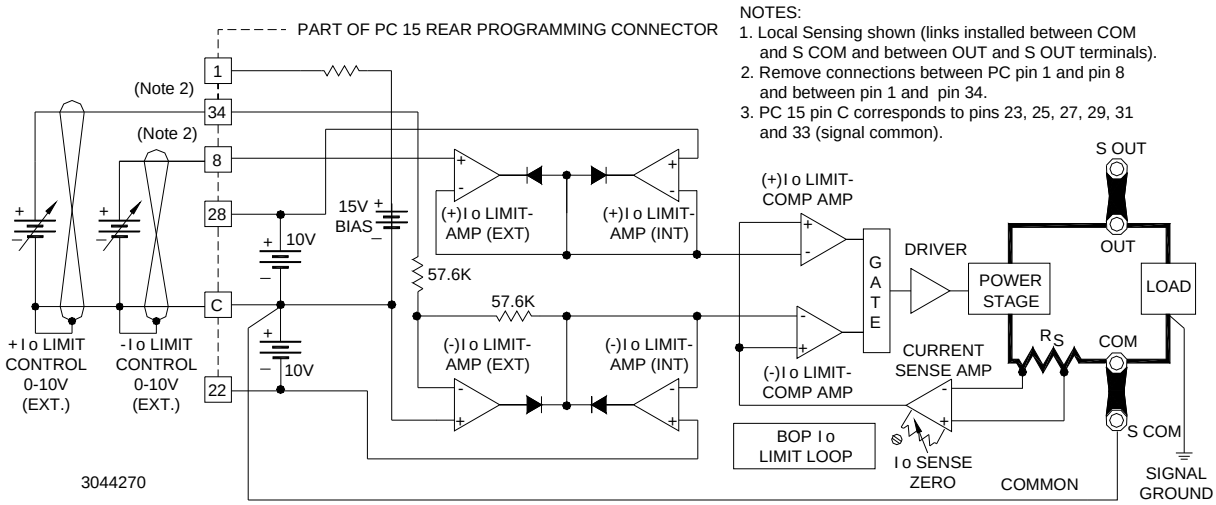


FIGURE 3-13. INDEPENDENT REMOTE CONTROL OF + I_O AND – I_O LIMITS

3.4 REMOTE CONTROL OF CURRENT CONTROL CHANNEL

Since the input signal requirements for the control of the BOP output current are the same as for programming the output voltage, and since the control circuitry of the current channel is almost identical to that of the voltage channel, all programming circuit descriptions for programming the output voltage of the BOP can be applied to current programming. The current programming circuits are illustrated in the following diagrams. Any comments specifically applicable to current channel programming are noted on the diagrams.

NOTE: Adjust the zero output current point by means of the built-in ZERO control of Preamp "B" or that of the I_O Comp. Amp.

3.4.1 REMOTE CURRENT CONTROL USING RESISTANCE (POTENTIOMETER)

Similar to remote voltage control mode described in PAR. 3.3.1, the BOP output current can be controlled over the full output range by means of an external resistance as shown in Figure 3-14. The control potential (zero to (±)10V) is applied via Preamp “B,” operating with unity gain, to the I_O Comp. Amp which drives the Power Stage with a fixed gain as listed in Table 1-3. A d-c control signal from zero to (±)10V will, therefore, control the BOP output current through its specified range.

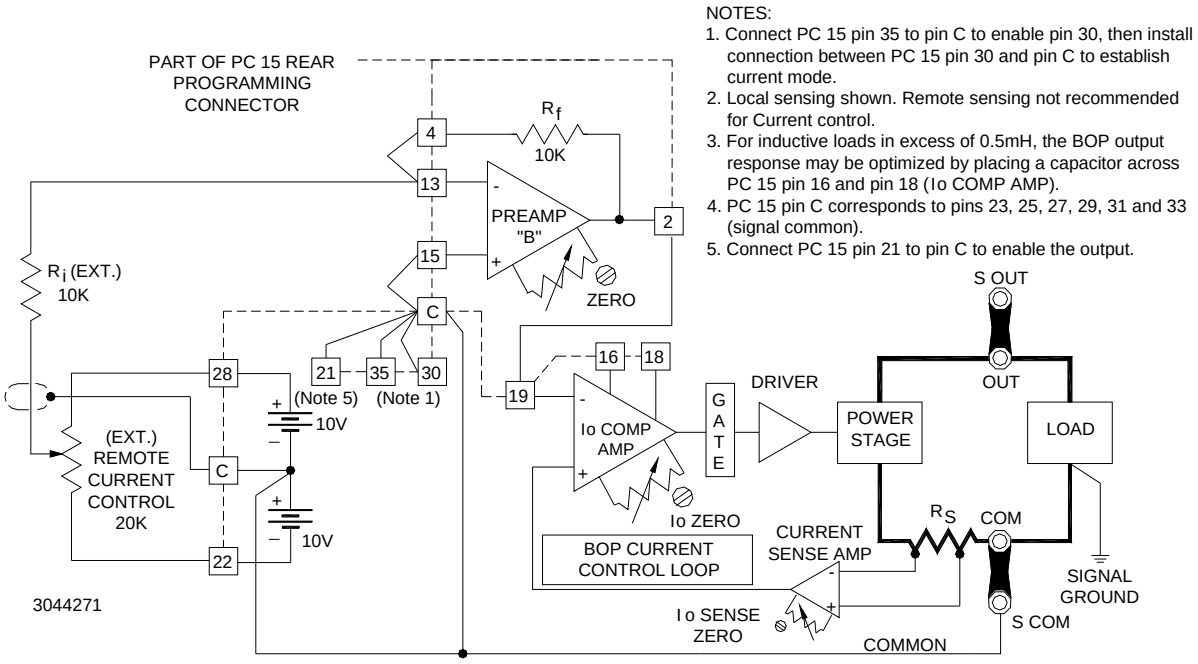


FIGURE 3-14. REMOTE POTENTIOMETER CONTROL OF THE BOP OUTPUT CURRENT

3.4.2 REMOTE CURRENT CONTROL USING RESISTANCE (DECADE BOX)

See Figure 3-15 for remote current control using a two-terminal resistance (decade box)

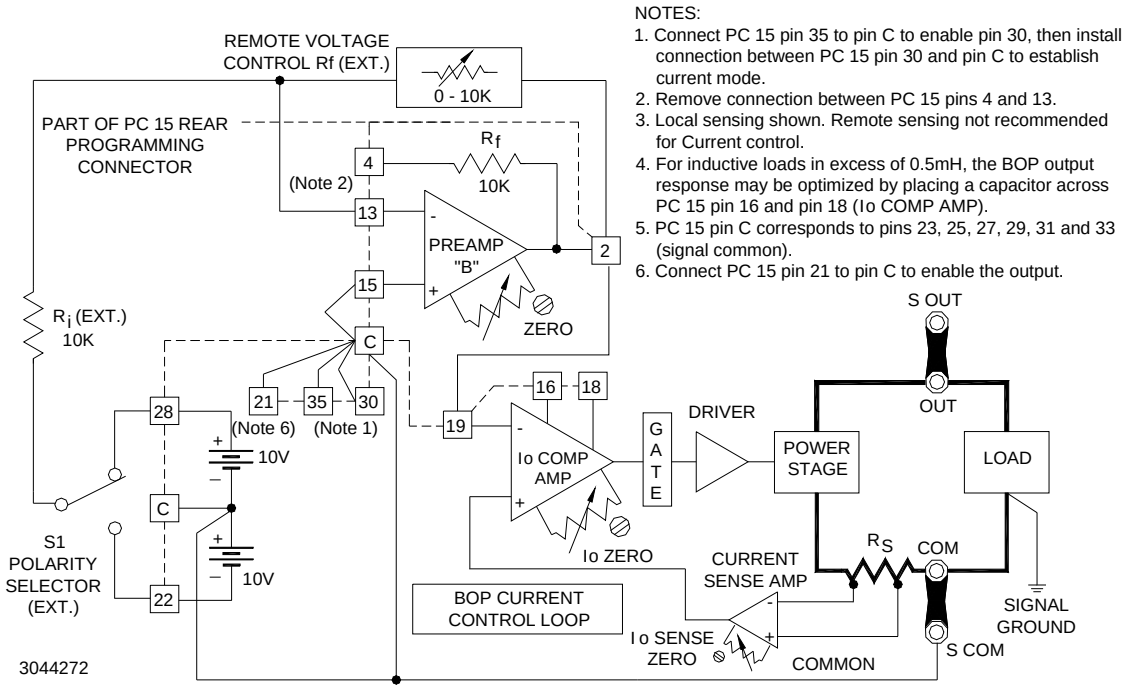


FIGURE 3-15. REMOTE D-C OUTPUT CURRENT CONTROL USING TWO-TERMINAL RESISTANCE (DECADE)

3.4.3 REMOTE CURRENT CONTROL USING D-C SIGNAL VOLTAGE

See Figure 3-16 for remote current control using a d-c signal voltage.

See Figure 3-17 for remote current control using a high impedance ±1V voltage. Preamplifier “B” is used in the non-inverting configuration, so the output current will have the same polarity as the input programming signal.

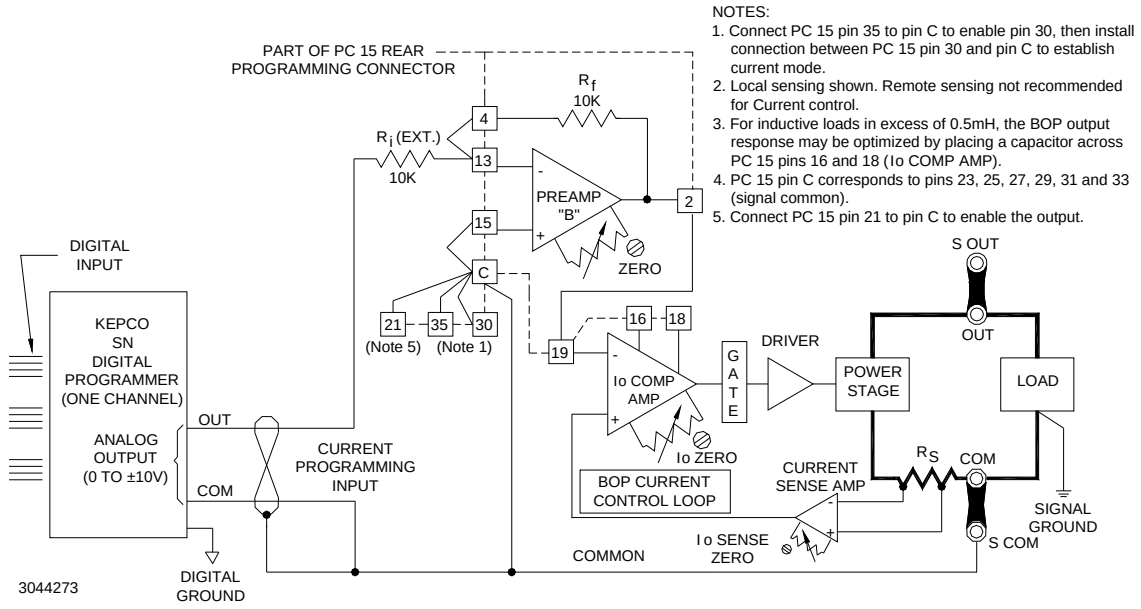


FIGURE 3-16. DIGITAL CONTROL OF OUTPUT CURRENT

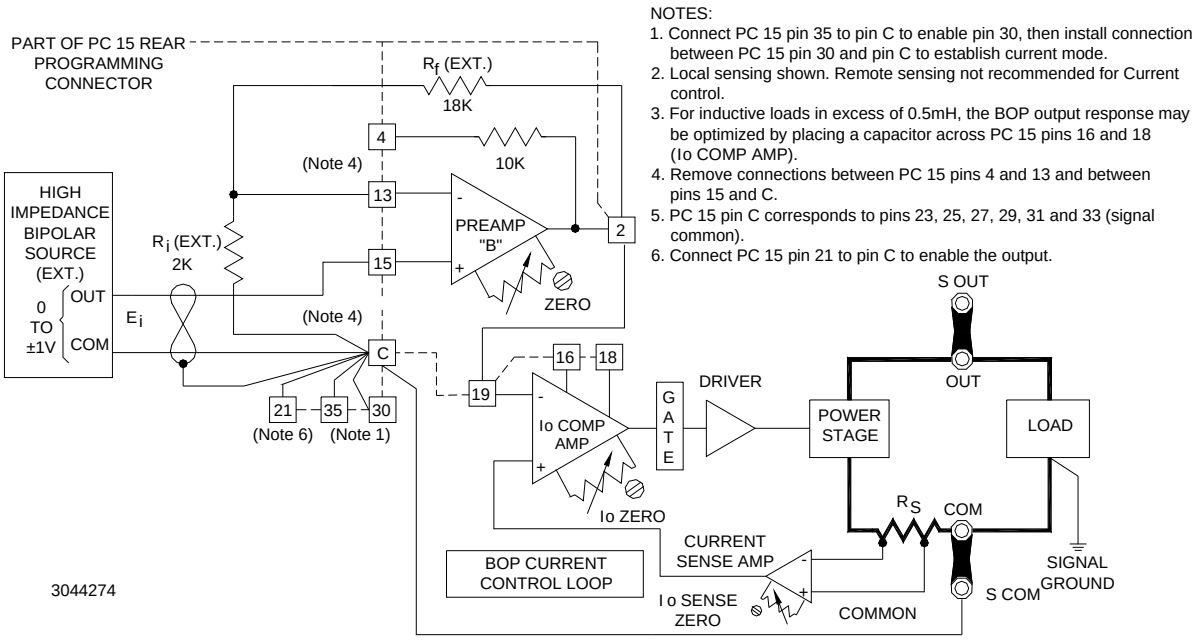


FIGURE 3-17. BOP OUTPUT CURRENT CONTROL WITH NON-INVERTING CONFIGURATION AND HIGH IMPEDANCE (±)1 VOLT SIGNAL SOURCE

3.4.4 USING THE BOP AS A CURRENT AMPLIFIER

See Figure 3-18 to configure the BOP as a current-stabilized bipolar amplifier.

See Figure 3-19 to drive output current using a bipolar signal voltage less than ±10V.

See Figure 3-20 to drive output current using a high impedance source.

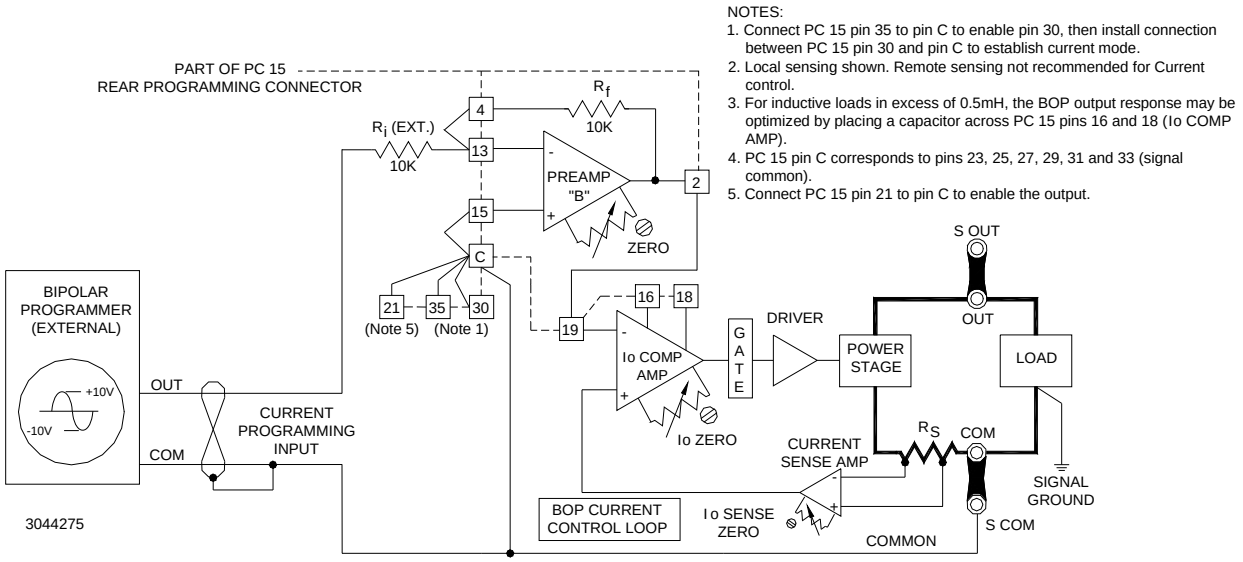


FIGURE 3-18. BASIC PROGRAMMING CIRCUIT FOR BOP AS BIPOLAR CURRENT-STABILIZED AMPLIFIER

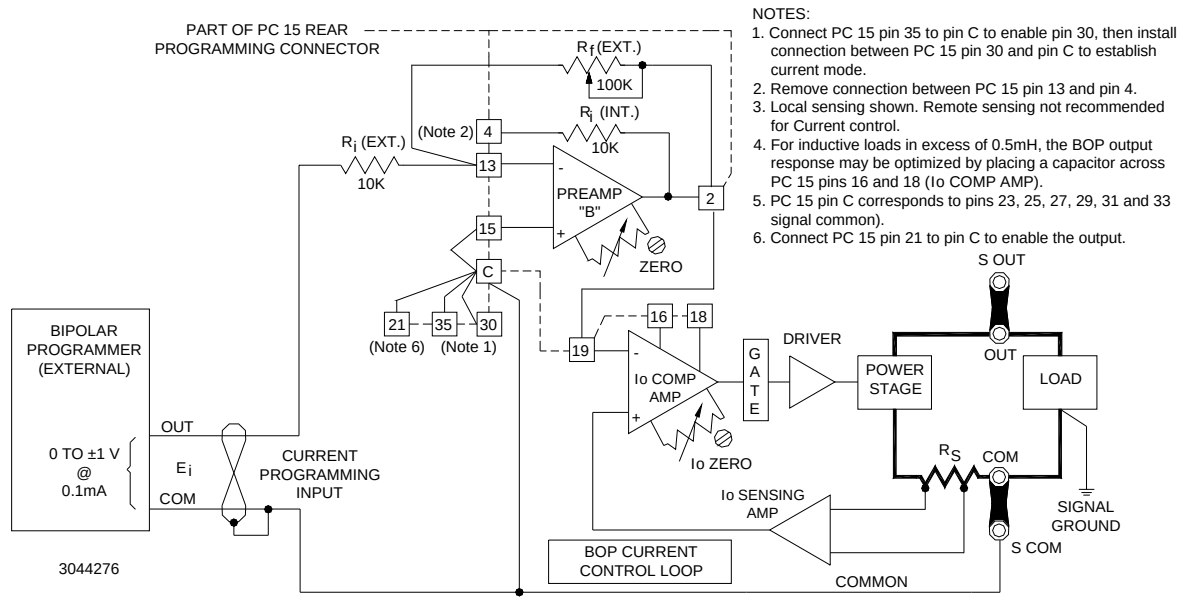


FIGURE 3-19. PROGRAMMING CIRCUIT FOR DRIVING OUTPUT CURRENT WITH ±1V INPUT PROGRAMMING VOLTAGE

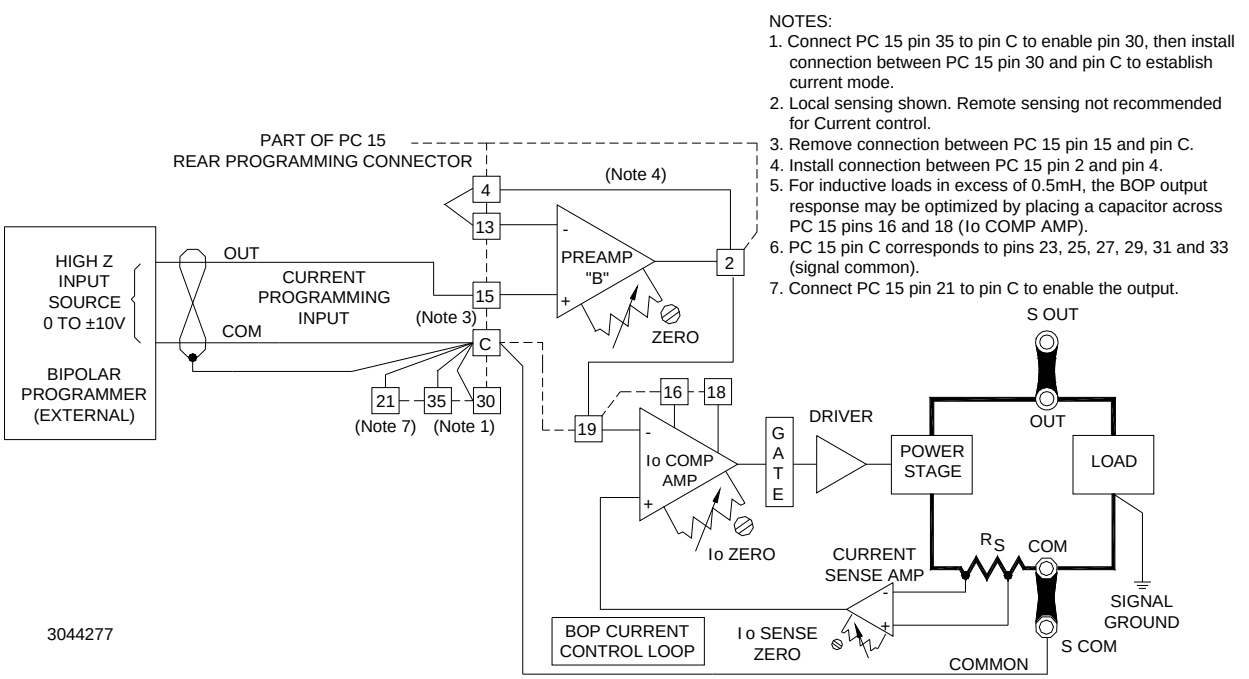


FIGURE 3-20. PROGRAMMING CIRCUIT FOR DRIVING OUTPUT CURRENT WITH A HIGH IMPEDANCE SOURCE

3.4.5 REMOTE CONTROL OF VOLTAGE LIMITING

The E_O Limit circuit can be configured to use a single external 0 to 10V d-c programming source to symmetrically control ± Voltage Limits as shown in Figure 3-21. The circuit functions in a manner similar to the I_O Limit circuit (refer to Paragraph 3.3.5), except that the voltage limit feedback is derived by means of a voltage divider across the BOP output.

The + and – voltage limit can be controlled independently by individual 0 to 10V d-c sources, as illustrated in Figure 3-22.

NOTE: If a control voltage is not applied to pins 6 and 32 of the Rear Programming Connector, the ± E_O LIMIT value is automatically set to 110% of the rated nominal voltage of the BOP. This is a backup of the external programming of voltage limits described above.

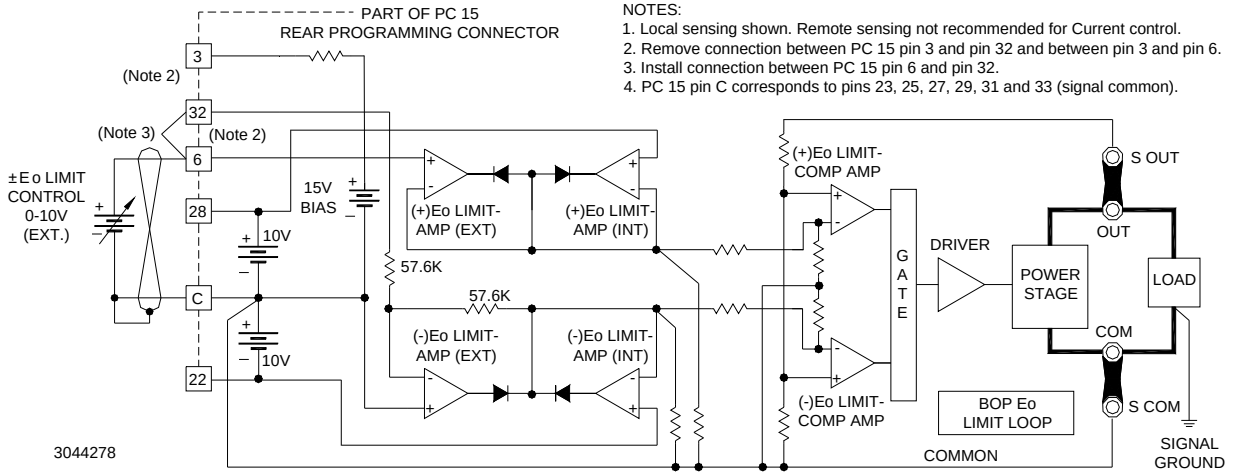


FIGURE 3-21. SYMMETRICAL REMOTE CONTROL OF VOLTAGE LIMIT

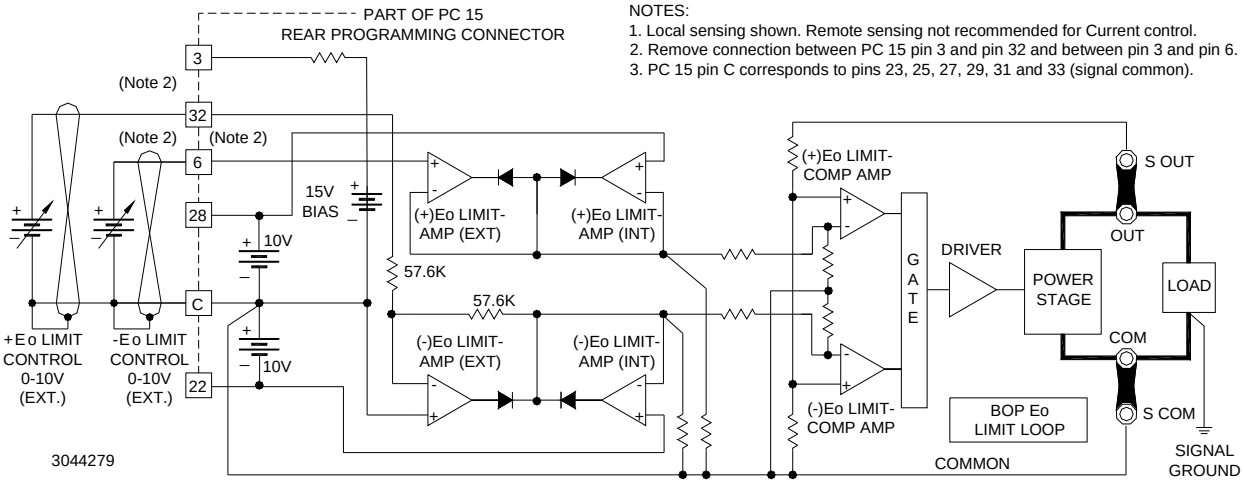


FIGURE 3-22. INDEPENDENT REMOTE CONTROL OF $+E_O$ AND $-E_O$ VOLTAGE LIMIT

3.5 OPERATING A CHANNEL USING REMOTE DIGITAL INTERFACE

To use the GPIB interface connect the channel to the GPIB interface as specified in PAR. 2.10 and set the GPIB address to address the desired channel (see PAR. 2.5.1).

To use the RS 232 interface connect the channel to the RS 232 interface as specified in PAR. 2.11 and set the GPIB address to select the desired baud rate (see PAR. 2.5.2).

Each channel of the Kepco BOP 2X-DG Power Supply may be programmed by sending IEEE 488 and SCPI (Standard Commands for Programmable Instruments) commands via either the GPIB or RS 232 interface. SCPI provides a common language used in an automatic test sys-

tems (see Appendix A and B for details). Additional remote programming information can be found in the following paragraphs, as well as PAR's 3.8 and 3.9.

3.5.1 PROGRAMMING TECHNIQUES TO OPTIMIZE PERFORMANCE

Proper programming techniques can offer significant response time improvement and reduce undesirable transients at the channel output. The key to performance optimization is to minimize mode changes (voltage mode/current limit to current mode/voltage limit or vice versa). Mode changes should be limited to changes in load conditions (to which the channel will respond automatically), or by programming the limit parameter when required by the user application.

The proper way to program the channel is to initially program the operating parameter to zero and the complementary limit parameter to the desired maximum value. Subsequent commands should change only the operating parameter. (The operating parameter is the parameter that controls the output, e.g., voltage controls the output in voltage mode, current in current mode.) The complementary limit parameter should be programmed only when there is a need to change it.

3.5.1.1 SETTING BOP VOLTAGE AND CURRENT LIMITS

The GPIB or RS 232 interface controls the channel output via two control loops: the main control loop, which is defined by the operating mode, and the limit control loop. In voltage mode, the main control loop handles output voltage and the limit control loop handles current limit. In current mode the main control loop handles output current and the limit control loop handles voltage limit.

- If the channel is operating in voltage mode, sending VOLT sxxxx controls the main control loop, and thus the output voltage (where s = sign, + or – and xxxx = the absolute value of output voltage). Sending CURR sxxxx controls current limit (where s = sign, + or – and xxxx = the absolute value of current limit.

- If the channel is operating in current mode, sending CURR sxxxx controls the main control loop, and thus the output current (where s = sign, + or – and xxxx = the absolute value of output current). Sending VOLT sxxxx controls voltage limit (where s = sign, + or – and xxxx = the absolute value of voltage limit.

3.5.1.2 AUTOMATIC RANGE OPERATION.

The channel has low and high ranges which are changed automatically when the main control loop set point is changed from a value less than the 1/4 of the units operating capability to a value that is higher than 1/4 of the operating capability. The automatic gain change is enabled at power up, or by either the *RST command or a mode change command (FUNC:MODE).

An automatic range change can result in undesirable transients when passing through the quarter scale point as result of a gain change, depending upon the load at the output of the channel. For example, with a 50V channel in voltage mode, the quarter scale gain change is at 12.5V. When the unit is programmed from 12.5V to 12.6V, the gain of the channel changes and a spike may be seen at the output. A transient may also occur when changing from –5V volts to +50V as –5 volts is in the region between –12.5V and +12.5V and +50V is beyond the quarter scale region. **When the unit is programmed to remain in full scale, the gain change does not occur and this eliminates the transient.**

3.5.1.3 USING THE GPIB OR RS 232 INTERFACE TO PRODUCE A SOFTWARE-TIMED RAMP AT THE OUTPUT

If the channel is intended to operate in a test application that requires a set of specific output voltages in a specific order and the number of points is more than the 1000 points available using the LIST command, a computer-controlled ramp can be used.

The recommended way to run a voltage ramp is to set the current limit to maximum plus at least 2% (or unit limit for max absolute value of the load). While in voltage mode, current limit is the absolute value of the programmed current.

Before running the ramp it is recommended that the scale be set to full scale using VOLT:RANG 1 to avoid any transients as explained in PAR. 3.5.1.2.

When installed, the Labview interactive example available on our web site creates a KpDCpwr directory in Labview's user.lib. This sub-directory contains various programs such as KepcoDCPwr Software Timed Ramp.vi. This vi is a software-timed linear ramp that is an example of using both the range command and the limits in generating a ramp. There is also a KepcoDCPwr Software Timed Current Ramp.vi for the same functionality in current mode.

3.5.2 MAKING SURE THE PREVIOUS COMMAND IS COMPLETE

Some SCPI commands require a flash memory update and can take an indeterminate amount on time to complete. These commands are:

- *SAV
- MEM:PACK
- MEM:UPD
- CAL:COPY
- CAL:SAVE
- SYST:PASS:NEW
- SYST:SEC:IMM

When sending these commands via Telnet, these commands require a query to be added to the command string to verify the previous command is complete. When the command is complete, the unit updates the status byte and indicates MAV (Message Available, bit 4 - see Table A-3) is true. MAV indicates that there is a response to be received by the computer, so when it becomes set, the unit is ready for its next command after reading back the data from the query that was added to the command string.

The *ESR? query is ideal to check if the previous command is complete since it returns either a 1 or 0. It is important that it be sent as a part of the same string as the command that causes a flash update. As an example, sending CAL:SAVE 12/31/2005;:*esr? or *esr?;:CAL:SAVE 12/31/2005 are valid command strings. Sending the commands separately will not verify that the previous command is complete. Figure 3-23 is a program written in Visual C, a trademark of Microsoft corporation, incorporating these techniques. The Visual C project for this example is part of the Sample VISA programs for Visual Studio file that can be downloaded from Kepco's Website (see **www.kepcopower.com/drivers/drivers-dl3.htm#bit4886)**.

Failure to provide an adequate delay can result in:

- Commands that are not processed,
- The following command may be received in error, causing an error in the transmission,
- Unit lock-up requiring power cycling of the unit.

3.5.3 PASSWORD SETUP.

To change a password send:

```
SYST:PASS:CEN DEFAULT

SYST:PASS:NEW DEFAULT new_password
```

If the password is lost, please contact the factory.

3.6 BUILT IN TEST

The GPIB interface is designed to perform periodic testing of itself, power up testing and commanded testing. The testing is reported to the operator both by errors placed in the SCPI error queue and by issuing audible beeps.

The CVI driver and demonstration soft panel can be downloaded from the Kepco website at: **www.kepcopower.com/drivers.htm**. The driver and demonstration panel provide an easy way to verify GPIB interface performance. A description of the driver is included as an Adobe Acrobat ™ PDF file.

3.6.1 POWER-UP TEST

The power up testing of the GPIB interface is similar to the *TST command (PAR. A.16). Portions of the interface are verified on power-up and errors are indicated by the beep codes listed in Table 3-1. The failed test is repeated indefinitely.

```
/*********************************************************************/
/* Kepco Sample Program using National Instruments VISA            */
/* note : visa32.lib must be included in your project               */
/*********************************************************************/
/*              Read and Write to an Instrument Example              */
/*                                                                   */
/* This code demonstrates synchronous read and write commands to a  */
/* GPIB, serial or message-based VXI instrument using VISA.         */
/*                                                                   */
/* The general flow of the code is                                  */
/*      Open Resource Manager                                       */
/*      Open VISA Session to an Instrument                          */
/*      Write the Identification Query Using viWrite                */
/*      Try to Read a Response With viRead                          */
/*      Close the VISA Session                                      */
/*********************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];

/*
* In every source code or header file that you use it is necessary to prototype
* your VISA variables at the beginning of the file. You need to declare the VISA
* session, VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
* VISA session that need this prototyping. As an example, above retCount has been
* prototyped as a static variable to this particular module.   It is an integer of
* bit length 32. If you are uncertain how to declare your VISA prototypes refer
* to the VISA help under the Section titled Type Assignments Table. The VISA
* help is located in your NI-VISA directory or folder.
*/

int main(void)
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle.  We will store this handle in defaultRM.
     */
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
      exit (EXIT_FAILURE);
   }
```

FIGURE 3-23. PROGRAMMING EXAMPLE TO VERIFY PREVIOUS COMMAND HAS COMPLETED. (SHEET 1 OF 3)

```
 /*
  * Now we will open a VISA session to a device at Primary Address 6.
  * You can use any address for your instrument. In this example we are
  * using GPIB Primary Address 6.
  *
  * We must use the handle from viOpenDefaultRM and we must
  * also use a string that indicates which instrument to open.  This
  * is called the instrument descriptor.  The format for this string
  * can be found in the NI-VISA User Manual.
  * After opening a session to the device, we will get a handle to
  * the instrument which we will use in later VISA functions.
  * The two parameters in this function which are left blank are
  * reserved for future functionality.  These two parameters are
  * given the value VI_NULL.
  *
  * This example will also work for serial or VXI instruments by changing
  * the instrument descriptor from GPIB0::2::INSTR to ASRL1::INSTR or
  * VXI0::2::INSTR depending on the necessary descriptor for your
  * instrument.
  */
status = viOpen (defaultRM,  "GPIB0::6::INSTR", VI_NULL, VI_NULL, &instr);
if (status < VI_SUCCESS)
{
     printf ("Cannot open a session to the device.\n");
     goto Close;
}

 /*
  * Set timeout value to 5000 milliseconds (5 seconds).
  */
status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 5000);

 /*
  * At this point we now have a session open to the instrument at
  * Primary Address 6.  We can use this session handle to write
  * an ASCII command to the instrument.  We will use the viWrite function
  * to send the string "*IDN?", asking for the device's identification.
  */
strcpy(stringinput,"*IDN?");
status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
if (status < VI_SUCCESS)
{
   printf("Error writing to the device\n");
   goto Close;
}

 /*
  * Now we will attempt to read back a response from the device to
  * the identification query that was sent.  We will use the viRead
  * function to acquire the data.  We will try to read back 100 bytes.
  * After the data has been read the response is displayed.
  */

status = viRead (instr, buffer, 100, &retCount);
if (status < VI_SUCCESS)
{
   printf("Error reading a response from the device\n");
}
else
{
   printf("%*s\n",retCount,buffer);
}
```

FIGURE 3-23. PROGRAMMING EXAMPLE TO VERIFY PREVIOUS COMMAND HAS COMPLETED (SHEET 2 OF 3)

```
      /*** Set Output Volt & Curr Values & enable output ***/

   strcpy(stringinput, "volt 5;:curr 1;:outp on");
   status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
   if (status < VI_SUCCESS)
   {
       printf("Error writing to the device\n");
       goto Close;
   }
   printf("ready - hit a key to continue\n");  getchar();

   /*** send  measure volt & curr cmnds & get response ***/

   strcpy(stringinput, "meas:volt?;:meas:curr?");
   status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
   if (status < VI_SUCCESS)
   {
       printf("Error writing to the device\n");
       goto Close;
   }

   status = viRead(instr, buffer, 100, &retCount);
   if (strchr(buffer, 10)) strchr(buffer, 10)[0] = 0; // terminate the buffer so we don't get
trailing garbage
   if (status < VI_SUCCESS)
   {
       printf("Error reading a response from the device\n");
   }
   else
   {
       printf("Volt;Curr : %*s\n", retCount, buffer);
   }

   printf("ready - hit a key to continue\n");  getchar();

   /*** send  a reset ***/

   printf("resetting\n"); strcpy(stringinput, "*rst");
   status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
   if (status < VI_SUCCESS)
   {
       printf("Error writing to the device\n");
       goto Close;
   }

   printf("ready - hit a key to continue\n");  getchar();

   /*
    * Now we will close the session to the instrument using
    * viClose. This operation frees all system resources.
    */
Close:
   printf("Closing Sessions\nHit enter to continue.");
   fflush(stdin);
   getchar();
   status = viClose(instr);
   status = viClose(defaultRM);

   return 0;
}
```

FIGURE 3-23. PROGRAMMING EXAMPLE TO VERIFY PREVIOUS COMMAND HAS COMPLETED (SHEET 3 OF 3)

3.6.2 CONFIDENCE TEST

The SCPI command *TST? Is used to perform a confidence test of the interface. It verifies the microprocessor memory, the timer, optical buffer and reference of the GPIB interface. It does not affect the output of the channel. This command returns a 0 for pass and number from 1 through 1023 to indicate the cause of the error. The test executes each of the subtests even when any one fails. If any test fails, a bit is set in the error code which is returned to the user. The error codes returned are listed in Table 3-1

TABLE 3-1. BUILT-IN TEST ERROR CODES

TEST ACTIVE		ERROR CODES			MEANING
*tst?	DIAG:TST?	BIT	VALUE	BEEPS	
Yes	Yes		0		Pass (No error)
Yes	Yes	0	1	1	ROM error
Yes	Yes	1	2	1	RAM error
Yes	Yes	2	4	2	FLASH error -- Perform calibration to correct
Yes	Yes	3	8	3	Optical Buffer Error‘
Yes	Yes	4	16	6	Digital Pot error
No	Yes	5	32	5	Loop Back Test error
No	Yes	6	64		Max Voltage Output error
No	Yes	7	128		Min Voltage Output error
No	Yes	8	256		1/4 Scale Voltage error
No	Yes	9	512		1/4 Scale Voltage Readback error

3.6.3 CHANNEL TEST

The DIAG:TST? Command performs a test of the Channel. The test includes the GPIB interface internal tests of the DAC (Digital to Analog Converter), Limit and Op Amp circuits. If successful it programs the channel in voltage mode to the positive and negative limits of the device. It also verifies the Limit operation of the channel in current mode by setting the unit for the rated maximum current delivery and verifies that the voltage limit set at one volt operates correctly. It also tests the readback voltages in both high and low scales.

CAUTION: TO AVOID DAMAGE TO THE LOAD, DISCONNECT THE LOAD BEFORE ISSUING THIS COMMAND. (DURING THE SELF-TEST, THE BOP IS PROGRAMMED TO FULL SCALE POSITIVE AND FULL SCALE NEGATIVE OUTPUT.)

This command returns a 0 for pass, or a number from 1 through 1023 to indicate the cause of the error. The test executes each of the subtests even when any one fails. If any test fails, a bit is set in the error code which is returned to the user. The error codes returned are listed in Table 3-1.

3.7 RS 232 SERIAL INTERFACE

The RS 232 serial interface behaves like the GPIB interface in that the command is parsed after receiving a control character of either a Line Feed or Carriage Return. The serial interface supports six special control characters. The six special control characters are:

Escape ($1B_H$)	Causes the input buffer to be cleared. This character is used to ensure that the buffer is empty when the host powers on since it is possible that the Interface Card was previously powered on and received some characters prior to the initialization of the host computer.
Backspace (08_H)	Causes the last character in the input buffer to be removed from the input buffer queue.
Carriage Return ($0D_H$)	Causes the input buffer to be parsed by the BOP.
Line Feed ($0A_H$)	Causes the input buffer to be parsed by the BOP.

3.7.1 RS 232 IMPLEMENTATION

The following paragraphs are provided to help the user understand how the RS 232 serial interface is implemented in the BIT 4886. Since the RS 232 protocol does not use a parity bit, the echo mode is the default method used to ensure reliable communication between the command originator (computer) and the BIT 4886, thus avoiding a more complex “handshake” protocol.

When the BIT 4886 is in the RS 232 echo mode it returns all data sent to the host controller. The BIT 4886 provides one additional option that allows handshake communication: the XON XOFF method. In standard echo mode the controller must verify that each character is echoed back by the BIT 4886. As shown in Figure 3-24, there are times when the BIT 4886 does not echo back the character from the controller, requiring that the controller resend the character. By using the handshake option (XON XOFF) the host controller can ensure that serial data interrupts occurring after parsing of the incoming message do not result in lost data.

Figure 3-24 illustrates the default echo mode and the XON XOFF method described in the following paragraphs.

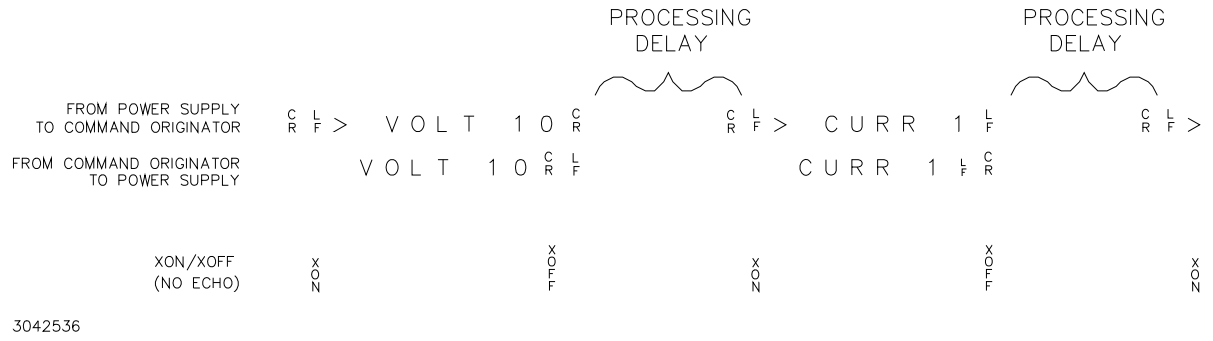


FIGURE 3-24. RS 232 IMPLEMENTATION

Only seven control characters (characters between 00_H and $1F_H$) are acknowledged by the BIT 4886:

- Carriage Return (CR, $0D_H$)
- Line Feed (LF, $0A_H$)
- Back Space (BS, 08_H)
- Escape (ESC, $01B_H$)

- XON (011_H)
- XOFF (013_H)
- CAN (018_H)

BS deletes the last character entered, with the exception of CR or LF characters. Either the CR or LF character acts as the line terminator, initiating parsing of the ASCII data sent to the BIT 4886 by the command originator. When the line is parsed and the commands are sent to the analog processor, the BIT 4886 sends the line terminator sequence CR LF to the command originator.

The ESC character is used for synchronization, causing the BIT 4886 to reset its input buffer and return a CR LF sequence.

The XON character enables the transmitter if XON/XOFF flow control is enabled (see PAR. 3.7.1.2).

The XOFF character stops data transmission if XON/XOFF flow control is enabled (see PAR. 3.7.1.2).

The CAN character resets the receive and transmit pointers and queues.

CAUTION: When the serial port has received an XOFF, the error message *-400, QUE error* will be placed in the queue to indicate the loss of transmitted information due to a received XOFF character. When XON is received, the unit will transmit all data in it's buffer followed by the exclamation character (!). This (!) character is not part of any message from the BIT 4886 and indicates the transmission buffer has been cleared and the BIT 4886 is idle.

All non-control characters are sent via the serial port of the command originator. The control character BS is echoed as BS Space BS. Only the first control character is returned in response to either a CR LF or LF CR character sequence (see Figure 3-24).

3.7.1.1 ECHO MODE

Echo mode is one method of ensuring data is transferred without errors. This mode should only be enabled when errors in operation are detected. Refer to PAR. B.76 to change the setting.

Each byte (character) is echoed back to the sender where it is verified as the same character that was just sent. If the wrong character is echoed back, sending the ESC character clears the line to allow retransmission of the character. It is important that CR and LF characters are NOT sent until the verification process is complete.

When working in echo mode, it is possible to receive the NAK (15 hex) from the BIT 4886. This indicates an unknown quantity of echoed characters have been lost due to a queue overflow problem. The error queue will also contain the *-400, QUE error* message, To prevent this, please insure the received data string does not exceed 127 characters between line terminators and no more than four queries are sent between line terminators in SCPI mode of operation

All non-control characters are sent via the serial port of the command originator.

3.7.1.2 XON XOFF METHOD

The XON XOFF method allows the BIT 4886 to control when the command originator is allowed to send data. The command originator can only send data after the XON (transmission on) character (011_H) has been received; the command originator stops sending data after receiving the XOFF (transmission off) character (013_H), and waits until the XON character is received before sending additional data. Refer to PAR. B.78 to change the setting.

Control characters, either CR or LF, are returned as XOFF CR if echo mode is on, and as XOFF if echo mode is off. XOFF stops data from the command originator and the BIT 4886 returns the normal sequence of CR LF (if echo mode is enabled).

3.7.1.3 ISOLATING RS 232 COMMUNICATION PROBLEMS

A Loop Back test can be executed to aid in isolating RS 232 communication problems. The test is executed via a SCPI command sent over the GPIB interface.

1. Send the command DIAG:LBT? to the unit via the GPIB interface with the Loop Back Test connector (Kepco P/N 195-0111) NOT installed. The response sent over the GPIB will be **FAILED**.

2. Install the Loop Back Test connector (Kepco P/N 195-0111) into the RS 232 port; if this connector is not available, skip this step and proceed to step 3. Send DIAG:LBT? again and read back the answer. if the response is PASSED, the power supply is operating properly. If the response is FAILED, the unit requires repair.

3. To test the integrity of the cable assembly connecting the BIT 4886 RS 232 port to the computer, remove the Loop Back test connector from the BIT 4886 RS 232 port and connect the cable in its place. With the DB9 adapter installed on the opposite end of the cable, connect a short jumper wire between pins 2 and 3 of the adapter connector and repeat the test of step 2 above. If the response is FAILED, the cord is either an improper type (not null modem) or defective. If the response is PASSED, the cable is correct; remove the jumper and reconnect the cable to the computer.

4. If the loop back connector was not available and steps 1 And 3 were completed successfully, contact Kepco to obtain loop back connector P/N 195-0111 and rerun the test. If each of the above steps is completed successfully, the problem lies in the computer hardware and/or software. If the serial communication is monitored at the computer end, every DIAG:LBT? command will cause the IDN response to be sent to the computer.

3.7.2 USING SCPI COMMANDS FOR RS 232 COMMUNICATION.

The unit must be in remote mode before RS 232 commands affecting the output can be executed (e.g., VOLT 10;OUTP ON). This can be accomplished by sending SYST:REM ON prior to sending any commands that affect the power supply output. (See PAR. B.87 and Figure B-7)

3.8 SCPI PROGRAMMING

SCPI (Standard Commands for Programmable Instruments) is a programming language conforming to the protocols and standards established by IEEE 488.2 (reference document *ANSI/ IEEE Std 488.2, IEEE Standard Codes, Formats, Protocols, and Common Commands*). SCPI commands are sent to the Interface Card as output strings within the selected programming language (PASCAL, BASIC, etc.) in accordance with the appropriate TCPIP function for port number 5025.

3.8.1 SCPI MESSAGES

There are two kinds of SCPI messages: program messages from controller to power supply channel, and response messages from the power supply channel to the controller. Program messages consist of one or more properly formatted commands/queries and instruct the channel to perform an action; the controller may send a program message at any time. Response messages consist of formatted data; the data can contain information regarding operating parameters, channel state, status, or error conditions.

3.8.2 COMMON COMMANDS/QUERIES

Common commands and queries are defined by the IEEE 488.2 standard to perform overall channel functions (such as identification, status, or synchronization) unrelated to specific channel operation (such as setting voltage/current). Common commands and queries are preceded by an asterisk (*) and are defined and explained in Appendix A. Refer also to syntax considerations (PARs. 3.8.2 through 3.8.5).

3.8.3 SCPI SUBSYSTEM COMMAND/QUERY STRUCTURE

Subsystem commands/queries are related to specific channel functions (such as setting output voltage, current limit, etc.) Figure 3-25 is a tree diagram illustrating the structure of SCPI subsystem commands used in the Interface Card with the “root” at the left side, and specific commands forming the branches. The following paragraphs introduce the subsystems; subsystem commands that are defined and explained in Appendix B.

3.8.3.1 INITIATE SUBSYSTEM

This subsystem enables the trigger system. When a trigger is enabled, the triggering action will occur upon receipt of a GPIB <GET>, *TRG or TRIGger command. If a trigger circuit is not enabled, all trigger commands are ignored.

3.8.3.2 MEASURE SUBSYSTEM

This query subsystem returns the voltage and current measured at the channel's output terminals.

3.8.3.3 [SOURCE:]VOLTAGE AND [SOURCE:]CURRENT SUBSYSTEMS

These subsystems program the output voltage and current of the channel.

3.8.3.4 OUTPUT SUBSYSTEM

This subsystem enables/disables the channel output. Voltage and current are determined by the combination of SOURCE subsystem VOLTage and CURRent commands and the load.

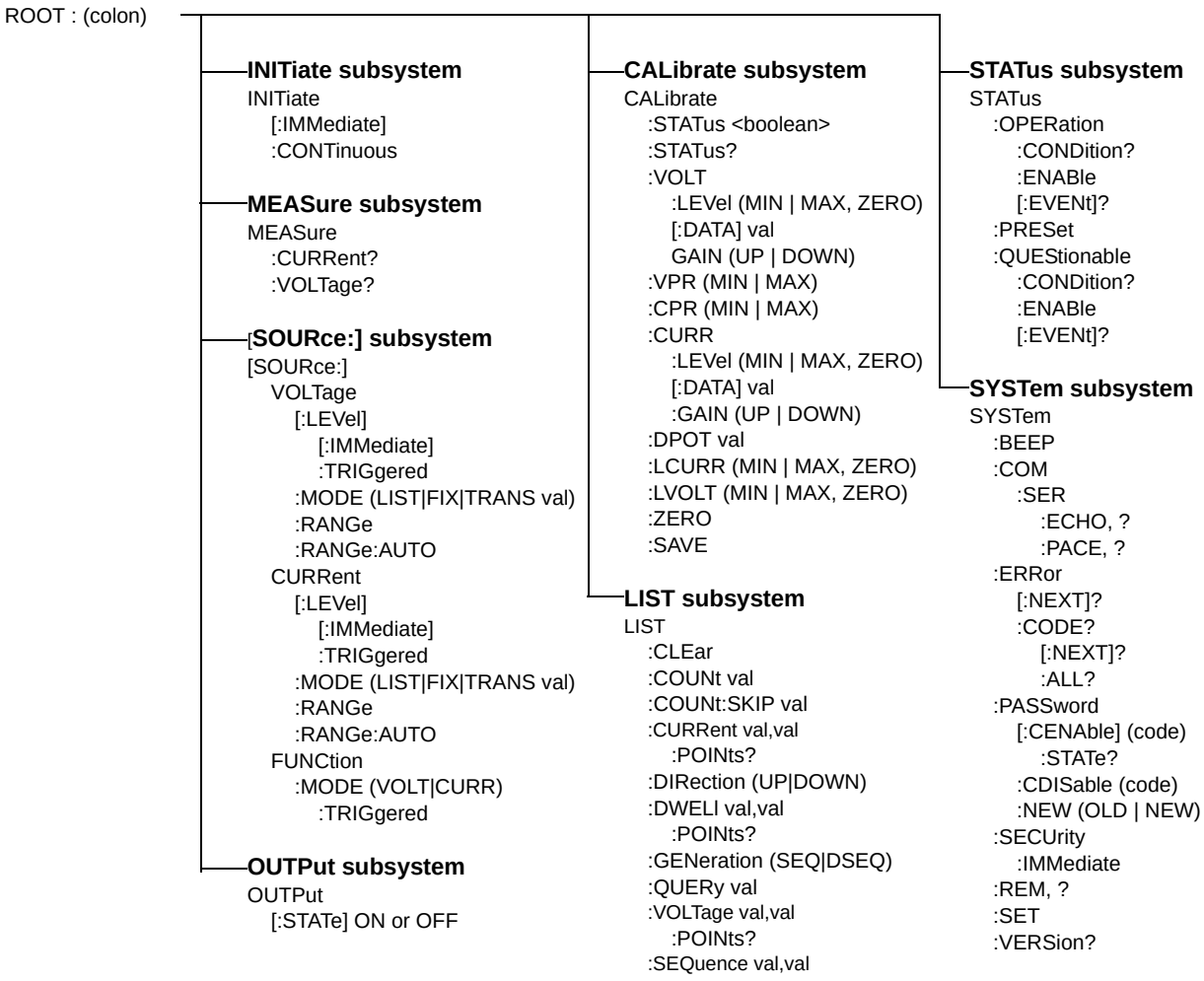


FIGURE 3-25. TREE DIAGRAM OF SCPI COMMANDS USED WITH BIT 4886 INTERFACE CARD

3.8.3.5 LIST SUBSYSTEM

The LIST subsystem is used to generate transients, waveforms and execute a series of steps repeatedly. Up to 1002 channel settings can be stored temporarily. Each setting consists of either a voltage or current value (depending on whether the channel has been set to Voltage or Current mode), and a corresponding dwell time (the duration those settings are in effect). These settings may be executed in sequence as they are entered, or executed in a user-determined sequence that also allows individual settings to be repeated more than once. In addition, the entire sequence may be repeated for a specific number of times, or run indefinitely until commanded to stop. The sequence can also be run in reverse order to produce inverted waveforms. The following paragraphs provide guidance for using the list commands.

3.8.3.5.1 REQUIRED LIST COMMANDS

There are only five LIST commands, plus either the VOLT:MODE or CURR:MODE command, that are needed to create and execute a list. Use of these required commands is illustrated in Figure B-2 which shows the creation of sawtooth and triangular waveforms. LIST commands are not accepted while a list is running. Send VOLT:MODE FIX (PAR. B.59) or CURR:MODE FIX (PAR. B.50) to stop the list. A list remains in memory until the power is cycled or the LIST:CLEAR command is processed. Therefore, if the original list is unchanged, additional commands can be added to the end of the list without resending all the commands. However, to change parameters of one or more commands within the list, send LIST:CLEAR and then resend the entire list including the changed parameters. To execute the list again, either VOLT:MODE LIST or CURR:MODE LIST must be sent again.

LIST:VOLT (PAR. B.45) or LIST:CURR (PAR. B.31). These commands establish the points (steps) of a list which program output voltage or current. A list can only be either a voltage list or a current list, so the points in a list must be made with either LIST:VOLT XXXX or LIST:CURR XXXX. Mixing of these commands within a list is not allowed.

LIST:CLEAR (PAR. B.26). Always precede a new list with this command. A list remains in memory until the power is cycled or the LIST:CLEAR command is processed.

LIST:DWELL (PAR. B.36). Defines the dwell time for each point in a list. In many instances it is easier to use one dwell time and repeat a specific point multiple times to generate longer duration pulses. Remember, if more than one LIST:DWELL is sent, there must be a LIST:DWELL for each voltage or current point in the list.

LIST:COUNT (PAR. B.27) Defines how many times a list is executed. 0 equals indefinite; when a count of 0 is used, either *RST, VOLT:MODE FIX or CURR:MODE FIX must be used to stop the list. *RST will cause the output to be set off and the unit is set to Voltage mode. VOLT:MODE FIX or CURR:MODE FIX stops the list immediately; the point being executed when the list is stopped will be present at the BOP output.

3.8.3.5.2 OTHER REQUIRED COMMANDS

VOLT:MODE LIST (PAR. B.59) or CURR:MODE LIST (PAR. B.50). These commands start the list and VOLT:MODE FIX (PAR. B.59) or CURR:MODE FIX (PAR. B.50) stop the list. LIST commands are not accepted while a list is running. If LIST:COUNT is between 1 and 255, when the count decrements to 0, the list stops and the unit automatically moves to VOLT:MODE FIXED or CURR:MODE FIXED state. To execute the list again, either VOLT:MODE LIST or CURR:MODE LIST must be sent again.

3.8.3.5.3 OTHER USEFUL COMMANDS

LIST:COUNT:Skip (PAR. B.29). This command provides the ability to execute the initial points only once whenever a list is run. It is used to set initial preconditions prior to running a repetitive sequence.

LIST:DIR (PAR. B.34) This command changes the order of the list.

LIST:VOLT:POINTS? (PAR. B.47) or LIST:CURR:POINTS? (PAR. B.33) These queries return the number of points in a list and provide a simple way to insure that all points entered were properly processed and as intended.

3.8.3.5.4 OPTIONAL COMMANDS

Most commands have associated Queries (?) These are useful for troubleshooting/debugging lists but are not needed in most cases.

The LIST:SEQ command is provided for backward compatibility. It is not recommended to be used in any new designs as it may be eliminated at some point in the future.

3.8.3.6 STATUS SUBSYSTEM

This subsystem programs the channel status register. The channel has two groups of status registers: Operation and Questionable. Each group consists of three registers: Condition, Enable, and Event.

3.8.3.7 SYSTEM SUBSYSTEM

This subsystem is used to establish system settings Program Message Structure.

3.8.4 PROGRAM MESSAGE STRUCTURE

SCPI program messages (commands from controller to power supply channel) consist of one or more ***message units*** ending in a ***message terminator*** (required by Kepco power modules). The message terminator is not part of the syntax; it is defined by the way your programming language indicates the end of a line (such as a “newline” or “end-of-line” character). The message unit is a keyword consisting of a single command or query word followed by a message terminator (e.g., CURR?<NL> or TRIG<end-of-line>). The message unit may include a data parameter after the keyword separated by a space; the parameter is usually numeric (e.g., CURR 5<NL>), but may also be a string (e.g., OUTP ON<NL>). Figure 3-26 illustrates the message structure, showing how message units are combined. The following subparagraphs explain each component of the message structure.

NOTE: An alternative to using the message structure for multiple messages defined in the following paragraphs is to send each command as a separate line. In this case each command must use the full syntax shown in Appendix B.

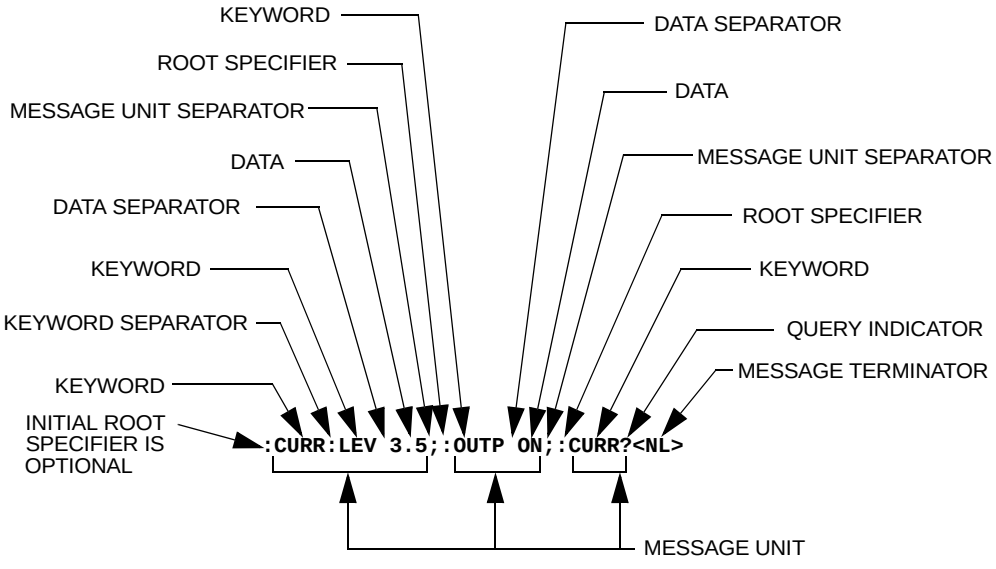


FIGURE 3-26. MESSAGE STRUCTURE

3.8.4.1 KEYWORD

Keywords are instructions recognized by a decoder within the Interface Card, referred to as a "parser." Each keyword describes a command function; all keywords used by the Interface Card are listed in Figure 3-25.

Each keyword has a long form and a short form. For the long form the word is spelled out completely (e.g. STATUS, OUTPUT, VOLTAGE, and TRIGGER are long form keywords). For the short form only the first three or four letters of the long form are used (e.g., STAT, VOLT, OUTP, and TRIG).

To identify the short form and long form in this manual, keywords are written in upper case letters to represent the short form, followed by lower case letters indicating the long form (e.g., IMMediate, EVENt, and OUTPut). The parser, however, is not sensitive to case (e.g., outp, OutP, OUTPUt, ouTPut, or OUTp are all valid).

You must use the rules above when using keywords. Using an arbitrary short form such as ENABL for ENAB (ENABLE) or IMME for IMM (IMMEDIATE) will result in an error. Regardless of which form chosen, you must include all the letters required by that form.

3.8.4.2 KEYWORD SEPARATOR

If a command has two or more keywords, adjacent keywords must be separated by a colon (:) which acts as the keyword separator (e.g., **`CURR:LEV:TRIG`**). The colon can also act as a root specifier (paragraph 3.8.4.7).

3.8.4.3 QUERY INDICATOR

The question mark (?) following a keyword is a query indicator. This changes the command into a query. If there is more than one keyword in the command, the query indicator follows the last keyword. (e.g., **`VOLT?`** and **`MEAS:CURR?`**).

3.8.4.4 DATA

Some commands require data to accompany the keyword either in the form of a numeric value or character string. Data always follows the last keyword of a command or query (e.g., **`VOLT:LEV:TRIG 14`** or **`SOUR:VOLT? MAX`**.

3.8.4.5 DATA SEPARATOR

Data must be separated from the last keyword by a space (e.g., **`VOLT:LEV:TRIG 14`** or **`SOUR:VOLT? MAX`**.

3.8.4.6 MESSAGE UNIT SEPARATOR

When two or more message units are combined in a program message, they must be separated by a semicolon (;) (e.g., **`VOLT 15;MEAS:VOLT?`** and **`CURR 12;CURR:TRIG 12.5`**).

3.8.4.7 ROOT SPECIFIER

The root specifier is a colon (:) that precedes the first keyword of a program message. This places the parser at the root (top left, Figure 3-25) of the command tree. Note the difference between using the colon as a keyword separator and a root specifier in the following examples:

VOLT:LEV:IMM 16 The root specifier is not necessary for the first keyword and has been omitted. Both colons are keyword separators.

:CURR:LEV:IMM 4 Even though it is not required, it is still correct to include the root specifier (first colon). The other two are keyword separators.

VOLT:LEV 6;:CURR:LEV 15 The root specifier for VOLT is not necessary because it is the first keyword in the string and has not been included. The second colon is the root specifier for CURR and is required; if it is missing an error will result. The first and third colons are keyword separators.

:INIT ON;:TRIG;:MEAS:CURR?;VOLT? The first three colons are root specifiers. The fourth colon is a keyword separator.

3.8.4.8 MESSAGE TERMINATOR

The message terminator defines the end of a message. Three message terminators are permitted:

- new line (**<NL>**), ASCII 10 (decimal) or 0A (hex)
- carriage return (**<CR>**), ASCII 13 (decimal) or 0D (hex)
- both of the above (**<CR> <NL>**). If both are sent, the second is null and has no effect.

Your GPIB interface card software will automatically send a message terminator. For example, the HP BASIC OUTPUT statement inserts **<NL>** after the last data byte. When binary data is exchanged, **<END>** must be used. The combination **<NL><END>** terminator can be used for all data except binary data.

NOTE: Kepco power modules ***require*** a message terminator at the end of each program message. The examples shown in this manual assume a message terminator will be added at the end of each message. Where a message terminator is shown it is represented as **<NL>** regardless of the actual terminator character.

3.8.5 UNDERSTANDING THE COMMAND STRUCTURE

Understanding the command structure requires an understanding of the subsystem command tree illustrated in Figure 3-25. The “root” is located at the top left corner of the diagram. The parser goes to the root if:

- a message terminator is recognized by the parser
- a root specifier is recognized by the parser

Optional keywords are enclosed in brackets [ ] for identification; optional keywords can be omitted and the channel will respond as if they were included in the message. The root level keyword [SOURce] is an optional keyword. Starting at the root, there are various branches or paths corresponding to the subsystems. The root keywords for the Interface Card are **:INITiate**, **:MEASure**, **:OUTPut**, **[:SOURce]**, **:STATus**, and **:SYSTem**. Because the **[SOURce]**

keyword is optional, the parser moves the path to the next level, so that **VOLTage**, **CURRent**, and **FUNCtion** commands are at the root level.

Each time the parser encounters a keyword separator, the parser moves to the next indented level of the tree diagram. As an example, the **STATus** branch is a root level branch that has three sub-branches: **OPERation**, **PRESet**, and **QUEStionable**. The following illustrates how SCPI code is interpreted by the parser:

STAT:PRES<NL>
The parser returns to the root due to the message terminator.

STAT:OPER?;PRES<NL>
The parser moves one level in from **STAT**. The next command is expected at the level defined by the colon in front of **OPER?**. Thus you can combine the following message units **STAT:OPER?** and **STAT:PRES;**

STAT:OPER:COND?;ENAB 16<NL>
After the **OPER:COND?** message unit, the parser moves in one level from **OPER**, allowing the abbreviated notation for **STAT:OPER:ENAB**.

3.8.6 PROGRAM MESSAGE SYNTAX SUMMARY

- Common commands begin with an asterisk (*).
- Queries end with a question mark (?).
- Program messages consist of a root keyword and, in some cases, one or more message units separated by a colon (:) followed by a message terminator. Several message units of a program message may be separated by a semicolon (;) without repeating the root keyword.
- If a program message has more than one message unit, then a colon (:) must precede the next keyword in order to set the parser back to the root (otherwise the next keyword will be taken as a subunit of the previous message unit).

  e.g., the command **meas:volt?;curr?** will read output voltage and output current since both **volt?** and **curr?** are interpreted as subunits of the **meas** command.
- Several commands may be sent as one message; a line feed terminates the message. Commands sent together are separated by a semicolon (;). The first command in a message starts at the root, therefor a colon (:) at the beginning is not mandatory.

  e.g., the command **meas:volt?;curr?** will read output voltage and output current, however the command **meas:volt?;:curr?** will read actual output voltage and *programmed current* since the colon preceding **curr?** indicates that **curr?** is not part of the **meas** command and starts at the root.
- UPPER case letters in mnemonics are mandatory (short form). Lower case letters may either be omitted, or must be specified completely (long form)
  e.g., **INSTrument** (long form) has the same effect as **INST** (short form).
- Commands/queries may be given in upper/lower case (long form)
  e.g., **SoUrCe** is allowed.
- Text shown between brackets [] is optional.
  e.g., **[SOUR:]VOLT[:LEV]:TRIG** has the same effect as **VOLT:TRIG**

3.8.7 SCPI PROGRAM EXAMPLE

Figure 3-27 is an example of a program written in Visual C using SCPI commands to program a BOP 2X-DG Power Supply channel. The Visual C project for this example is part of the Sample VISA programs for Visual Studio file that can be downloaded from Kepco's Website (see **www.kepcopower.com/drivers/drivers-dl3.htm#bit4886**). If it does not compile due to undefined functions, the issue is related to the importing of the visa.lib file in your project. It may be located in a different place than our supplied project and must be corrected.

The program illustrated is for a configuration using a National Instruments GPIB interface card. (It will be necessary to consult the manufacturer's data to achieve comparable functions with an interface card from a different manufacturer.) This program sets output voltage (Voltage mode) or voltage limit (Current mode) to 5V, and current limit (Voltage mode) or output current (Current mode) to 1A, then reads the measured (actual) voltage and current, then prints the measurements.

```
/* Kepco Sample Program using National Instruments VISA           */
/* note : visa32.lib must be included in your project              */
/*****************************************************************/
/*              Read and Write to an Instrument Example            */
/*                                                                 */
/* This code demonstrates synchronous read and write commands to a  */
/* GPIB, serial or message-based VXI instrument using VISA.        */
/*                                                                 */
/* The general flow of the code is                                 */
/*      Open Resource Manager                                      */
/*      Open VISA Session to an Instrument                         */
/*      Write the Identification Query Using viWrite               */
/*      Try to Read a Response With viRead                         */
/*      Close the VISA Session                                     */
/*****************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];

/*
* In every source code or header file that you use it is necessary to prototype
* your VISA variables at the beginning of the file. You need to declare the VISA
* session, VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
* VISA session that need this prototyping. As an example, above retCount has been
* prototyped as a static variable to this particular module.   It is an integer of
* bit length 32. If you are uncertain how to declare your VISA prototypes refer
* to the VISA help under the Section titled Type Assignments Table. The VISA
* help is located in your NI-VISA directory or folder.
*/

int main(void)
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle.  We will store this handle in defaultRM.
     */
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
```

FIGURE 3-27. TYPICAL EXAMPLE OF INTERFACE CARD PROGRAM USING SCPI COMMANDS (SHEET 1 OF 4)

```
 /*
  * First we must call viOpenDefaultRM to get the resource manager
  * handle.  We will store this handle in defaultRM.
  */
status=viOpenDefaultRM (&defaultRM);
if (status < VI_SUCCESS)
{
   printf("Could not open a session to the VISA Resource Manager!\n");
   exit (EXIT_FAILURE);
}

 /*
  * Now we will open a VISA session to a device at Primary Address 6.
  * You can use any address for your instrument. In this example we are
  * using GPIB Primary Address 6.
  *
  * We must use the handle from viOpenDefaultRM and we must
  * also use a string that indicates which instrument to open.  This
  * is called the instrument descriptor.  The format for this string
  * can be found in the NI-VISA User Manual.
  * After opening a session to the device, we will get a handle to
  * the instrument which we will use in later VISA functions.
  * The two parameters in this function which are left blank are
  * reserved for future functionality.  These two parameters are
  * given the value VI_NULL.
  *
  * This example will also work for serial or VXI instruments by changing
  * the instrument descriptor from GPIB0::2::INSTR to ASRL1::INSTR or
  * VXI0::2::INSTR depending on the necessary descriptor for your
  * instrument.
  */
status = viOpen (defaultRM,  "GPIB0::6::INSTR", VI_NULL, VI_NULL, &instr);
if (status < VI_SUCCESS)
{
     printf ("Cannot open a session to the device.\n");
     goto Close;
}

 /*
  * Set timeout value to 5000 milliseconds (5 seconds).
  */
status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 5000);
/*
  * At this point we now have a session open to the instrument at
  * Primary Address 6.  We can use this session handle to write
  * an ASCII command to the instrument.  We will use the viWrite function
  * to send the string "*IDN?", asking for the device's identification.
  */
strcpy(stringinput,"*IDN?");
status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
if (status < VI_SUCCESS)
{
   printf("Error writing to the device\n");
   goto Close;
}
```

FIGURE 3-27. TYPICAL EXAMPLE OF INTERFACE CARD PROGRAM USING SCPI COMMANDS (SHEET 2 OF 4)

```
     /*
      * Now we will attempt to read back a response from the device to
      * the identification query that was sent.  We will use the viRead
      * function to acquire the data.  We will try to read back 100 bytes.
      * After the data has been read the response is displayed.
      */

    status = viRead (instr, buffer, 100, &retCount);
    if (status < VI_SUCCESS)
    {
       printf("Error reading a response from the device\n");
    }
    else
    {
       printf("%*s\n",retCount,buffer);
    }

    /*** Set Output Volt & Curr Values & enable output ***/

    strcpy(stringinput, "volt 5;:curr 1;:outp on");
    status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
    if (status < VI_SUCCESS)
    {
        printf("Error writing to the device\n");
        goto Close;
    }
    printf("ready - hit a key to continue\n");  getchar();

    /*** send  measure volt & curr cmnds & get response ***/

    strcpy(stringinput, "meas:volt?;:meas:curr?");
    status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
    if (status < VI_SUCCESS)
    {
        printf("Error writing to the device\n");
        goto Close;
    }

    status = viRead(instr, buffer, 100, &retCount);
    if (strchr(buffer, 10)) strchr(buffer, 10)[0] = 0; // terminate the buffer so we don't get
trailing garbage
    if (status < VI_SUCCESS)
    {
        printf("Error reading a response from the device\n");
    }
    else
    {
        printf("Volt;Curr : %*s\n", retCount, buffer);
    }

    printf("ready - hit a key to continue\n");  getchar();
```

FIGURE 3-27. TYPICAL EXAMPLE OF INTERFACE CARD PROGRAM USING SCPI COMMANDS (SHEET 3 OF 4)

```
    /*** send  a reset ***/

    printf("resetting\n"); strcpy(stringinput, "*rst");
    status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput), &writeCount);
    if (status < VI_SUCCESS)
    {
        printf("Error writing to the device\n");
        goto Close;
    }

    printf("ready - hit a key to continue\n");  getchar();

    /*
     * Now we will close the session to the instrument using
     * viClose. This operation frees all system resources.
     */
Close:
    printf("Closing Sessions\nHit enter to continue.");
    fflush(stdin);
    getchar();
    status = viClose(instr);
    status = viClose(defaultRM);

    return 0;
}
```

FIGURE 3-27. TYPICAL EXAMPLE OF INTERFACE CARD PROGRAM USING SCPI COMMANDS (SHEET 4 OF 4)

3.9 ENHANCED OPERATION

The following paragraphs describe options that will permanently modify the behavior of the remote interface and associated BOP channel. Although these options use SCPI commands, they are not intended to be used during normal programming of the BOP output.

The commands described in the following paragraphs affect only RAM variables. The power up state of the variables is retrieved from the FLASH EEPROM. The command MEMORY:UPDATE can be used to copy this RAM information to the FLASH EEPROM. The WINDOW variable is stored using the command MEM:UPDATE SHUTDOWN and the other variables, including the DIAG:SAV command, use the command MEMORY:UPDATE INTERFACE

3.9.1 ERROR DETECTION

The interface card has the ability to alter the way the unit responds to detection of load and settings errors. The default behavior is for the interface to measure the main channel and verify that it is within the specified channel tolerances of the BOP channel, setting the appropriate bits in the Status Questionable register. This default behavior is established by the SYST:SEC:IMM command and the factory default window of 32 LSBs.

The default behavior can be modified using the DIAG:ERR, DIAG:ONL, DIAG:OFFL AND DIAG:WIN commands listed in Table 3-2 so that device errors are generated for conditions such as window exceeded, output current limit at setting and output current exceeding a hard limit. These errors can also be used to force the channel to go to an output off state upon detecting the errors. The behavior is altered, the new setting is stored in RAM and it remains in effect until

a power off/power on cycle. The settings can be saved for power up by sending the MEMORY:UPDATE SHUTDOWN command. Error responses are listed in Table 3-2.

3.9.2 LIMIT CHANNEL CONTROL

The interface supports three limit settings for the output off state. The standard configuration is for the limits to be set to the minimum levels which causes the output not to vary beyond the BOX limits defined in the BOP manual. The limits can also be set to maximum or 110% of rated voltage or current. This is useful for battery operation. These limit settings are established by the DIAG:OUTPut command. With other types of devices, it is possible to set the output off limit channels to be at any level by using the DIAG:OFFLimit commands to establish the limit for the output off state. DIAG:OUTP n_H (see Table 3-2) controls the limit DAC values when the output is set to OFF; n = 0 is the default. This sets both voltage and current limit channels to 0 when the output is off. n = 1 sets voltage limit to maximum when the output is off. n = 2 sets the current limit to the maximum when the output is off. Bits 2 and 3 of the hex character allow the DIAG:OFFLimit commands to establish a specific voltage limit and current limit, respectively, when the output is off. Bits 0 and 2 or bits 1 and 3 should not be set at the same time since the maximum setting will override the value specified by DIAG:OFFLimit. The SYST:SEC:IMM command sets DIAG:OUTP n = 0. The variables are stored in non-volatile EEPROM by the command MEMORY:UPDATE INTERFACE.

3.9.3 USING AND SAVING SYSTEM VARIABLES

The DIAG:SAV command prepares the variables established by the SYST:SET command see (PAR. B.90) to be stored and used for subsequent power up cycles. However, executing SYST:SEC:IMM (PAR. B.89) restores the following configuration for subsequent power up cycles: LF0, DC0, RO0. In addition to initializing all the variables (except password) to the default state, SYST:SEC:IMM clears the SAV setup area and performs a PACK of the memory partitions (see PAR. 3.9.4).

Saving LF1 means that the unit will provide a line feed if the buffer is empty and a read is performed. Saving LF0 means that each string sent to the GPIB port causes a null string to be transmitted. The null string is a single Line feed character. This command is useful during program debug as a read without a query will not hold up the bus.

Saving DC0 means that DCL and SDC have no effect on the device (power supply channel) except to clear internal status registers. Saving DC1 means that DCL and SDC commands will function the same as the *RST command: the output is set to 0 Volts, 0 Amperes, voltage mode is selected and the output is set to OFF (unless RO1 is also set).

Saving RO0 causes the *RST command to set the output to 0 Volts and 0 Amps and sets output to OFF, requiring an OUTP ON command to be sent before voltage/current commands are present at the output. Saving RO1 causes *RST to set the output to ON and sets the output to 0 Volts and 0 Amps. Subsequent VOLT and CURR commands affect the output directly without having to send OUTP ON. The OUTP ON and OUTP OFF commands function normally to turn the output on and off.

3.9.4 FLASH MEMORY EEPROM OPERATION

The unit's configuration, voltage and current, saved setups (*SAV and *RCL command) and Calibration values are stored in Flash Memory. Since the Flash EEPROM cannot be modified by writing a single data byte, the block must be erased and then the data written into the correct locations.

The GPIB interface accomplishes this by partitioning the Flash memory. As the amount of Flash memory used becomes larger with each subsequent data update, the Flash memory can fill up and needs to be compressed. The compression of the Flash EEPROM, called packing, is automatically handled by the GPIB interface microprocessor's code. The packing process can take a half a second to accomplish. Because it is automatically executed, it can occur after any *SAV, Memory:UPDATE, or save of Calibration. When the memory is out of space, the internal microprocessor copies the live data to a backup flash area, erases the main flash data area, copies the current control variables into this clean space and then erases the backup flash area. The pack process can take 500 milliseconds to complete. Packing is automatically invoked during power up initialization if the GPIB interface finds that any storage area is over 90% utilized.

The following statistics are a guide as to when a MEMORY PACK will occur.

- Save area - Approximately 300 *SAV operations can be completed before a PACK will occur automatically.
- Variable area - around 30 updates can be completed prior to an automatic PACK occurring. Variable areas also include serial number updates and password updates.
- Calibration - 16 calibrations can be saved before a PACK is required.

If the memory is over 80% full, the user can also initiate packing by sending the command MEMORY:PACK. Issuing a MEMORY:PACK command if the memory is less than 80% full will have no effect. Because this command may take a long time to complete, it is recommended that the user send MEMORY:PACK;*ESR?. The computer can then perform serial polls to wait for the command to complete. The command will be complete when the serial poll byte indicates there is data in the output buffer. (bit 4 set in the returned status byte).

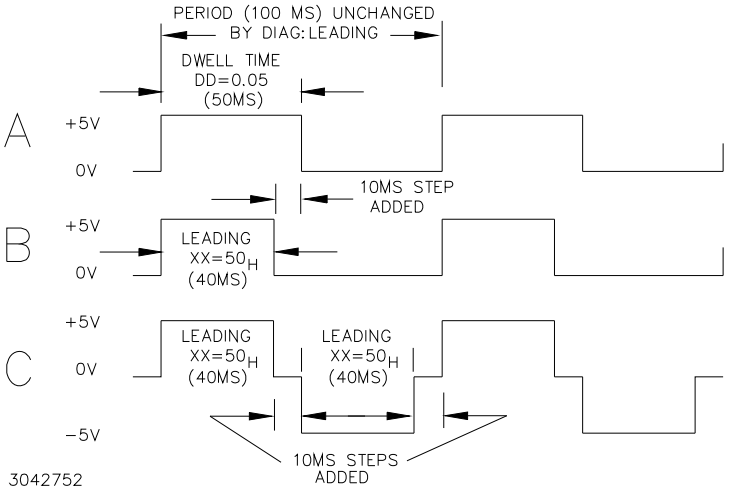
TABLE 3-2. ENHANCED OPERATION - ERROR RESPONSE

COMMAND	QUERY	DESCRIPTION
DIAG:ERR:CURR xx (xx =hex values from 00 to FF) Factory default value: 0	DIAG:ERR:CURR?	Allows both current protect and current limit functionality. Bit 7 - when set will establish a fixed Current Limit level when output is ON. Bit 6 - when set enables the current protection error logic. Bit 5 - when set enables Current Protect error bit in ESR/ESE register Bit 4 - when set causes the power up and reset state of the ESE register to be as follows: ESE - 72 if bit 5 set, 8 if not set. Bit 3 - when set allows a current protect error to create an error message of “303 - Current above Protection limit.” Bit 2 - when set causes a current protect error to set the output to off. Bit 1 - when sets allows a current measurement in current mode to generate a “Device Error” message. Bit 0 - when set enables a current measurement error in current mode to shut down unit. The query provides one byte which indicates the current limit setting.
DIAG:ERR:VOLT xx (xx =hex values from 00 to FF) Factory default value: 0	DIAG:ERR:VOLT?	Allows voltage protect and voltage limit functionality. Bit 7 - when set establishes a fixed Voltage Limit level when output is ON. Bit 6 - when set enables the voltage protection error logic. Bit 3 - when set allows a voltage protect error to create an error message of “304 - Voltage above Protection limit” Bit 2 - when set causes a voltage protect error to set the output to off. Bit 1 - when set allows a voltage measurement error in voltage mode to generate a “Device Error” message. Bit 0 - when set allows voltage “device error” message (see bit 1) to set the output to off. The query provides one byte which indicates the voltage limit setting.

TABLE 3-2. ENHANCED OPERATION - ERROR RESPONSE (CONTINUED)

COMMAND	QUERY	DESCRIPTION
DIAG:OFFLimit:CURR xx (xx = decimal value) Factory default value: 0	DIAG:OFFLimit:CURR?	The decimal value allows the user to establish the actual current limit DAC (digital to analog converter) setting when the output is off and the unit is in voltage mode. This variable is only used when bit 4 of the DIAG:OUTP register is set. The value is calculated using the equation: Current expected /(Current max * 1.1) * 256. The DVS standard setting is 128 derived from (1.1 / 2.2) * 256. The query provides one byte which indicates the actual DAC current limit setting.
DIAG:OFFLimit:VOLT xx (xx = decimal value) Factory default value: 0	DIAG:OFFLimit:VOLT?	The decimal value allows the user to establish the actual voltage limit DAC (digital to analog converter) setting when the output is off and the unit is in current mode. This variable is only used when bit 3 of the DIAG:OUTP register is set. The value is calculated using the equation: Voltage expected / Voltage max * 1.1) * 256 The DVS standard setting is 0. The query provides one byte which indicates the actual DAC current limit setting.
DIAG:ONLimit:CURR xx (xx = decimal value) Factory default value: 0	DIAG:ONLimit:CURR?	The decimal value allows the user to establish the actual current limit DAC (digital to analog converter) setting when the output is on and the unit is in voltage mode. This variable is only used when bit 7 of the DIAG:ERR:CURR register is set. The value is calculated using the equation: Current expected /(Current max * 1.1) * 256. The DVS standard setting is 128 derived from (1.1 / 2.2) * 256. The query provides one byte which indicates the actual DAC current limit setting.
DIAG:ONLimit:VOLT xx (xx = decimal value) Factory default value: 0	DIAG:ONLimit:VOLT?	The decimal value allows the user to establish the actual voltage limit DAC (digital to analog converter) setting when the output is on and the unit is in current mode. This variable is only used when bit 7 of the DIAG:ERR:VOLT register is set. The value is calculated using the equation: Voltage expected / Voltage max * 1.1) * 256 The DVS standard setting is 0. The query provides one byte which indicates the actual DAC current limit setting.
DIAG:OUTP n (n = hex character)	DIAG:OUTP? (Returns hex character)	DIAG:OUTP n controls the limit DAC values when the output is set to OFF. The values can either be set to maximum or to a user determined limit value (using the DIAG:OFFLimit commands. The SYST:SEC:IMM command sets DIAG:OUTP n = 0. DIAG:OUTP 0 = the default. This sets both voltage and current limit channels to 0 when the output is off. The four bits of the hex character are defined as follows: Bit 0 = 1 sets voltage limit to maximum when in Voltage mode and the output is off. Bit 1 = 1 sets current limit to maximum when in Current mode and the output is off. Bit 2 = 1 sets voltage limit to value determined by the DIAG:OFFLimit:VOLT command when the output is off. Bit 3 = 1 sets current limit to value determined by the DIAG:OFFLimit:CURR command when the output is off. For example, DIAG:OUTP 3 = sets both voltage and current limit to maximum. This can be useful for battery applications where the limit channel can allow the battery to discharge the battery voltage. Using both the maximum and DIAG:OFFLimit options (e.g., bits 0 and 2) at the same time is not recommended - the maximum value overrides the DIAG:OFFLimit setting.
DIAG:SAV	n/a	DIAG:SAV stores variables established by SYST:SET. (See PAR. 3.9.3 and B.82.)

TABLE 3-2. ENHANCED OPERATION - ERROR RESPONSE (CONTINUED)

COMMAND	QUERY	DESCRIPTION
DIAG:WIN xx (xx = hex value from 10 to 40)[1] Factory default value: 10	DIAG:WIN? (Returns window value xx (hex)	Allows user to specify value in LSB's for error window used for detection of output voltage or output current measurement errors (default = 32). Error window value can only be set between 16 (10 Hex) and 64 (40 Hex) using one two-digit hex character (xx)[1]. Query returns 2-digit hex character xx.
DIAG:LEADING xx (xx is the time in half milliseconds increments) (hex) Factory default value: 0	DIAG:LEADING? returns xx (hex), the time in half milliseconds increments)	xx establishes max dwell time for a two-step list. If the dwell time DD for the two-step list exceeds xx, a step is inserted with voltage = 0V for (DD - xx). MSTS default = 50. Consider the following: LIST:VOLT 0,5 Two step list, 0V then 5V. LIST:DWEL .05 Dwell time of 50 ms LIST:COUNT 0 Repeat continuously. OUTP ON Output enabled. VOLT:MODE LIST Run program The above list would normally produce a continuous square wave, 50ms at 0V, 50mS at 5V (Fig A). If DIAG:LEADING 50 (40 ms) is issued, the square wave changes to 60ms at 0V, 40mS at 5V as follows: the first 1/2 cycle = 50mS at 0V, the second 1/2 cycle = 40mS (the limit set by DIAG:LEADING) at 5V, then 10mS (50 - 40 = 10) at 0V (Fig B). If LIST:VOLT 0,5 is changed to LIST:VOLT -5,5 both 1/2 cycles will show 10mS steps at 0V (Fig C). 
DIAG:RESTore TO FACTORY (double space required before and after TO)	None	Restores diagnostic settings to the default values and stores them in non-volatile memory. This is a password protected command. This command requires string TO FACTORY to execute correctly. Possible Errors: -109,"Missing Parameter" - TO FACTORY was not found after command, double space is required on both sides of TO. -203,"Command Protected" - syst:PASS:STAT? response not 1 -311,"Memory Error" - unable to store data, try again.

1 The default window value of 32 indicates a 32 LSB margin for error during readback. The value for Readback is arrived at by taking 16 samples of voltage or current and averaging them. A readback error is produced when the calculated readback value (with calibration constants applied) exceeds the maximum readback (calculated readback + error window) or minimum readback (calculated readback – error window). Increasing the error window can avoid undesired errors. For example, in cases where the LIST command is used to produce a square wave output, overshoot inherent in the characteristics of the power supply can sometimes cause the readback average to increase enough to cause a readback error. Increasing the error window allows the square wave to be generated without producing an error.

3.9.4.1 CALIBRATION STORAGE

The GPIB interface maintains the calibration tables in Flash Memory until a PACK is executed. There are six calibration areas maintained in Flash Memory: Working, Prior, Oldest, Factory, Master, and First.

The calibration can be copied to another area using the CAL:COPY command. The syntax is as follows: CAL:COPY (source) (destination) where (source) and (destination) refer to the areas of Flash memory where calibration data is stored, designated as: WORKing, PRIor, OLDest, FACTory, MASTer, FIRst. Source refers to the calibration area that is to be copied, destination to the area that the calibration will be copied into.

The Master calibration should never be overwritten. Factory, Master, and First are set to the same values at the factory. The Working calibration is the active calibration. Each time a CAL:SAV is executed, the values are saved in the Working (active) area. At the same time, the values previously stored in Working are moved to Prior, and the values previously stored in Prior are moved to Oldest. The values previously stored in Oldest are no longer available. Table 3-3 illustrates calibration storage and the use of the CAL:COPY command. An example is shown in PAR. 3.5.2

TABLE 3-3. CALIBRATION STORAGE

COMMAND	WORKing	PRIor	OLDest	FACTory	MASTer	FIRst
1. CAL:SAVE	Cal 1 values			factory cal.	Master cal.	First cal.
2. CAL:SAVE	Cal 2 values	Cal 1 values		No Change	No Change	No Change
3. CAL:SAVE	Cal 3 values	Cal 2 values	Cal 1 values	No Change	No Change	No Change
4. CAL:SAVE	Cal 4 values	Cal 3 values	Cal 2 values	No Change	No Change	No Change
5. CAL:COPY FACTory WORKing	factory cal.	Cal 4 values	Cal 3 values	No Change	No Change	No Change
6. CAL:COPY PRIor WORKing	Cal 4 values	factory cal.	Cal 4 values	No Change	No Change	No Change

3.10 SERIES AND PARALLEL CONNECTION OF BOP CHANNELS

BOP channels can be connected in series to increase output voltage of the series connected pair. They can also be connected in parallel to increase output current of the parallel-connected pair.

Voltage sources, such as batteries, cells or stabilized d-c power supplies can be readily series-connected, but normally they can not be paralleled. Even small differences in their terminal potential would cause large, damaging, circulating currents. By complementary reasoning, current sources can easily be paralleled, but not series-connected.

There are two basic methods which are commonly used for the series or parallel connection of stabilized d-c power supplies. The first method is the “Automatic” series or parallel connection by which the power supply outputs are simply connected together, in series or parallel, as required. The “Automatic” method presents no problems as long as it is confined to series connection with voltage sources, or to parallel connection with current sources. The proper settings for automatic series and parallel connection are as follows:

- Automatic series connection: All BOP channels configured for voltage control. Set current limit of one channel lower than current limit for the other.
- Automatic parallel connection: All BOP channels configured for current control. Set voltage limit of one channel lower than voltage limit for the other.

The second method for series or parallel connection of BOP channels is the Master/Slave configuration, by which a designated Master channel controls the output of the Slave channel. This control method permits control of the common output by a single channel - the Master. Since this method presents the least problems for either series or parallel connections and since it is readily implemented with BOP power supplies, it is the recommended method for this kind of operation and is described below.

3.10.1 SERIES CONNECTION, MASTER/SLAVE OPERATION

Two BOP channels can be connected in series to increase the unit's output voltage. The Slave channel is set to always be in voltage mode, adding to the output voltage of the Master channel by repeating the Master's voltage, while the Master can operate in either voltage mode or current mode as needed.

A Master/Slave series configuration of multiple units **REQUIRES** that the slave be under analog control. However analog control is only possible if the BIT card for the channel is disabled. To disable the BIT card and allow analog control, refer to Application Note BOP 006, available on the Kepco web site at:

www.kepcopower.com/support/bop-app-note.htm

NOTE: If the bit cards for BOTH channels are disabled, the unit will behave exactly like a BOP 2X-D power supply. In this case refer to the BOP 2X-D Operator Manual instead of this manual for all installation, operation and calibration instructions. The BOP 2X-D Operator Manual is available on the Kepco web site at:

www.kepcopower.com/support/opmanls.htm#bop-2x

Proceed as follows:

1. Connect the BOP channels as shown in Figure 3-28.
   a. Load-wire size should be rated to at least the maximum load current and load-wire pairs should be twisted or tied together to keep the inductance and EMI as low as possible. For remote sensing, use #22 AWG twisted pair wires.

   b. Use shielded cable for connecting the coupling resistor (R_T). Connect shield to chassis-ground. Either the positive or the negative output side of the system can be grounded.

   c. The coupling resistor (RT) should be either a wirewound or a metal film component with a low temperature coefficient. It is calculated by:
      R_T (KOhms) = (E_OM (MAX) Volts)/1mA
      This value of R_T will be able to drive the Slave channel to its (±) maximum output voltage as the Master channel is controlled through its range.

   d. Make all changes to Master and Slave channels as described in the NOTES of Figure 3-28

   e. The BOP Circuit Breaker Control Circuit, as described in PAR. 3.10.2, should be used to interconnect both BOP channels in such a way that if one of them should fail, the other one is automatically shut down also.

2. Turn BOP channels on and begin operation. The system as shown in Figure 3-28 operates in Voltage Mode. When multiple channels are under digital control, the digital readback is for the master only. A good approximation for total output voltage of series configurations using two channels is to double the readback voltage (readback current is accurate).

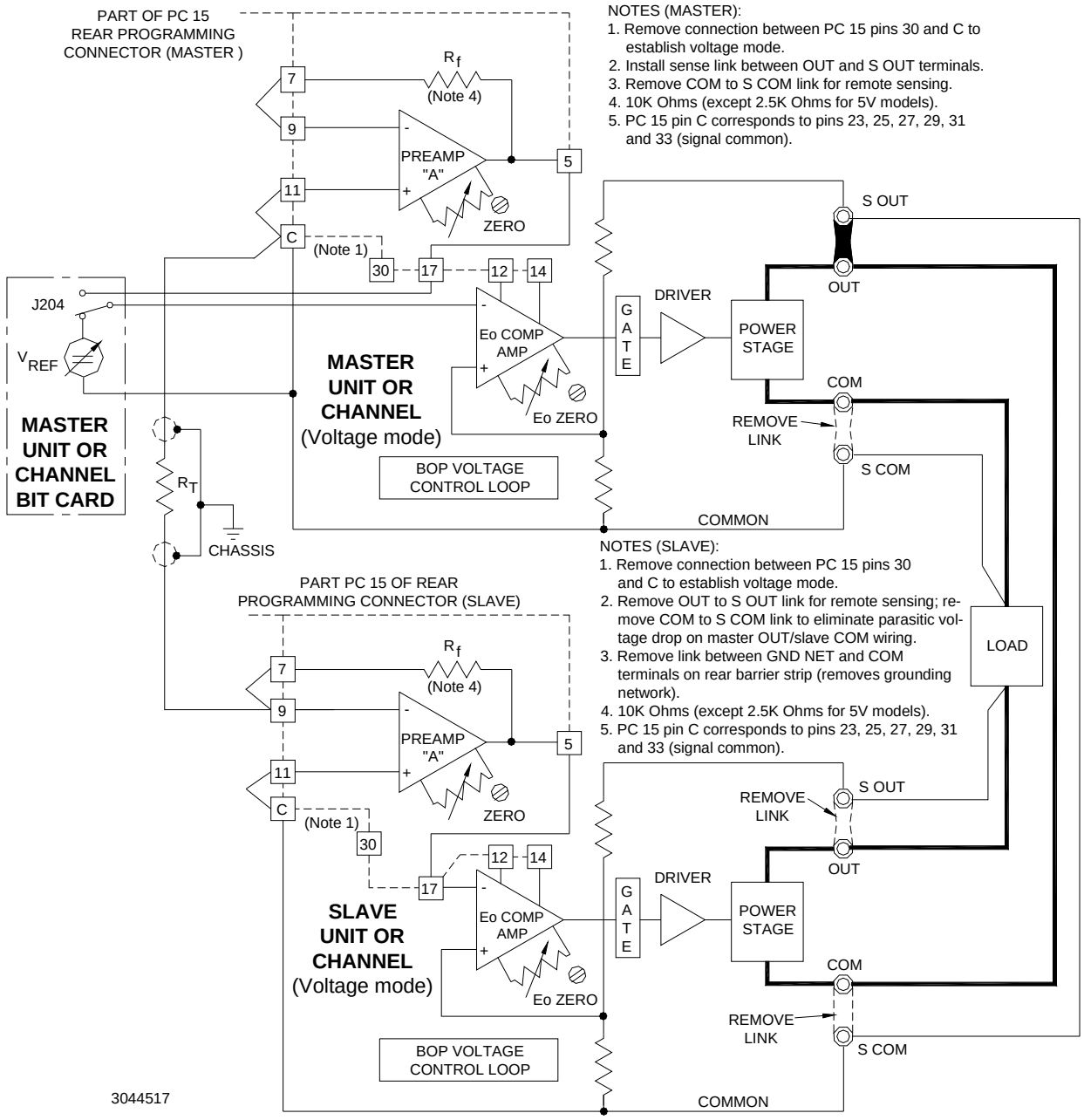


FIGURE 3-28. SERIES CONNECTION, VOLTAGE MODE (SHOWN), MASTER/SLAVE WITH DIGITAL CONTROL OF MASTER, REMOTE ERROR SENSING, SLAVE BIT CARD DISABLED

3.10.2 CIRCUIT BREAKER CONTROL CIRCUIT

Each BOP channel is protected by the dual-coil circuit breaker/ON-OFF switch (CB101) as shown in Figure 3-29. The current sensing coil of CB101 is connected in series with the primary winding of T201, thus detecting overcurrent reflected into the primary circuit and subsequently tripping the circuit breaker. The voltage sensing coil of CB101 is connected in series with an SCR (CR4) across the driver collector supply. If the SCR is triggered on, the voltage sensing coil is energized and trips the circuit breaker. The SCR (CR4) can be triggered in several ways:

- On a-c line power loss or interruption, Q1 will loose its turn-on bias first, due to the small time-constant provided in its base circuit (not shown in the simplified block diagram). As a result, Q2 is turned on, energizing the diode in the optical isolator (LC-1) and providing a turn-on signal for the SCR (CR4).

- In the event of an excessive heat rise on the power stage assembly, the thermal switch (S401) will close, energizing the diode in the optical isolator (LC-1), thus providing the turn-on signal for the SCR (CR4).

- An input signal applied across pins 49 (+) and 47 (–) of the PC 15 Rear Programming Connector energizes the diode in the second optical isolator (LC-2) which, in turn, energizes the diode in the first optical isolator (LC-1), thus providing for SCR turn-on.

NOTE: Upon turn-off, an isolated output signal is available at PC 15 Rear Programming Connector pins 26 (+) and 36 (–). This output signal can be used to turn off other equipment when the channel is turned off. As an example, applying it across pins 49 and 47 of the Rear Programming Connector of the other BOP channel allows the master of a parallel or series configuration to turn off the slave when the master is turned off.

- A manual, non-isolated turn-off is provided which, when implemented with an external switch contact as shown in Figure 3-29, will energize the diode in the first optocoupler (LC-1). thus providing the SCR turn-on.

- If it is <u>not</u> desired to trip the circuit breaker (CB101) upon a-c line power loss, it can be prevented by a connection or an external switch across pins 24 and 25 on the PC 15 Rear Programming Connector. In this manner, Q1 is inhibited from providing a turn-on signal for Q2, thus avoiding triggering the SCR and tripping the circuit breaker.

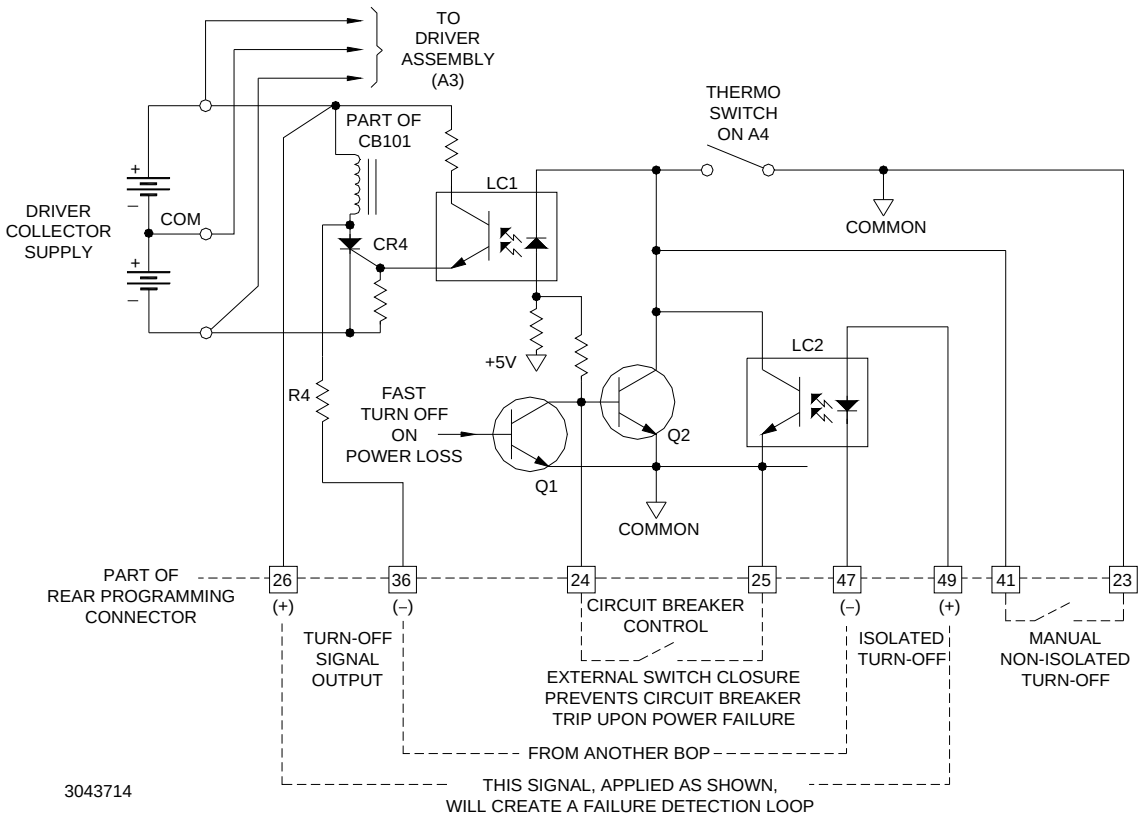


FIGURE 3-29. CIRCUIT BREAKER CONTROL CIRCUIT, SIMPLIFIED DIAGRAM.

3.10.3 PARALLEL CONNECTION, MASTER/SLAVE OPERATION

Both BOP channels of the BOP 2X-DG Dual Channel unit can be connected in parallel to increase output current. The slave channel is set to always be in current mode, adding to the output current of the master by repeating the master's current, while the master can operate in either voltage mode or current mode as needed.

A Master/Slave parallel configuration of multiple units **REQUIRES** that the slave be under analog control. However analog control is only possible if the BIT card for the channel is disabled. To disable the BIT card and allow analog control, refer to Application Note BOP 006, available on the Kepco web site at:

www.kepcopower.com/support/bop-app-note.htm

NOTE: If the bit cards for BOTH channels are disabled, the unit will behave exactly like a BOP 2X-D power supply. In this case refer to the BOP 2X-D Operator Manual instead of this manual for all installation, operation and calibration instructions. The BOP 2X-D Operator Manual is available on the Kepco web site at:

www.kepcopower.com/support/opmanls.htm#bop-2x

Figure 3-30 shows a parallel master/slave connection of two BOP power channels. Proceed as follows:

3. Connect the BOP channels as shown in Figure 3-30.
   f. Make the changes to Master and Slave channels as directed in the NOTES of set-up diagram, Figure 3-30.

   g. Load-wire size should be rated for at least the maximum load current and load-wire pairs should be twisted or tied together to keep the inductance and EMI as low as possible. Use shielded cable for the drive signal connection between the BOP channels. Connect the shield to common ground. For remote sensing, use #22 AWG twisted pair wires.

   h. The BOP Circuit-Breaker Control Circuit, as described in PAR. 3.10.2 should be used to interconnect both BOP channels in such a way, that if one of them should fail, the other one is automatically shut down also.

4. Following the set-up procedure as described above, turn both BOP channels on and commence operation. The system as shown in the set-up diagram, Figure 3-30, operates in Voltage Mode. When multiple channels are under digital control, the digital readback is for the master only. A good approximation for total output current of parallel configurations using two channels is to double the readback current (readback voltage is accurate).

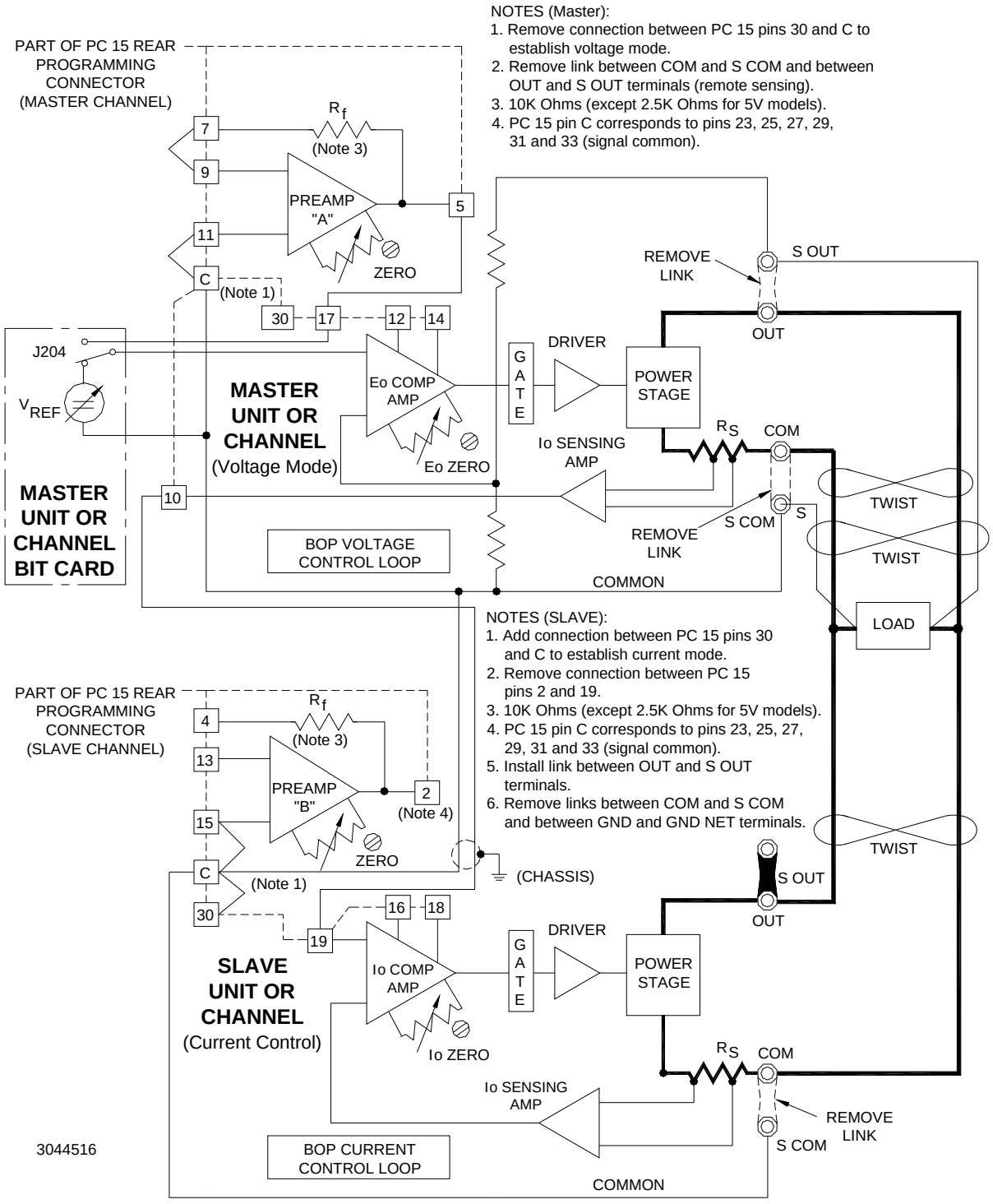


FIGURE 3-30. PARALLEL CONNECTION, VOLTAGE MODE (SHOWN) MASTER/SLAVE WITH DIGITAL CONTROL OF MASTER, REMOTE ERROR SENSING, SLAVE BIT CARD DISABLED

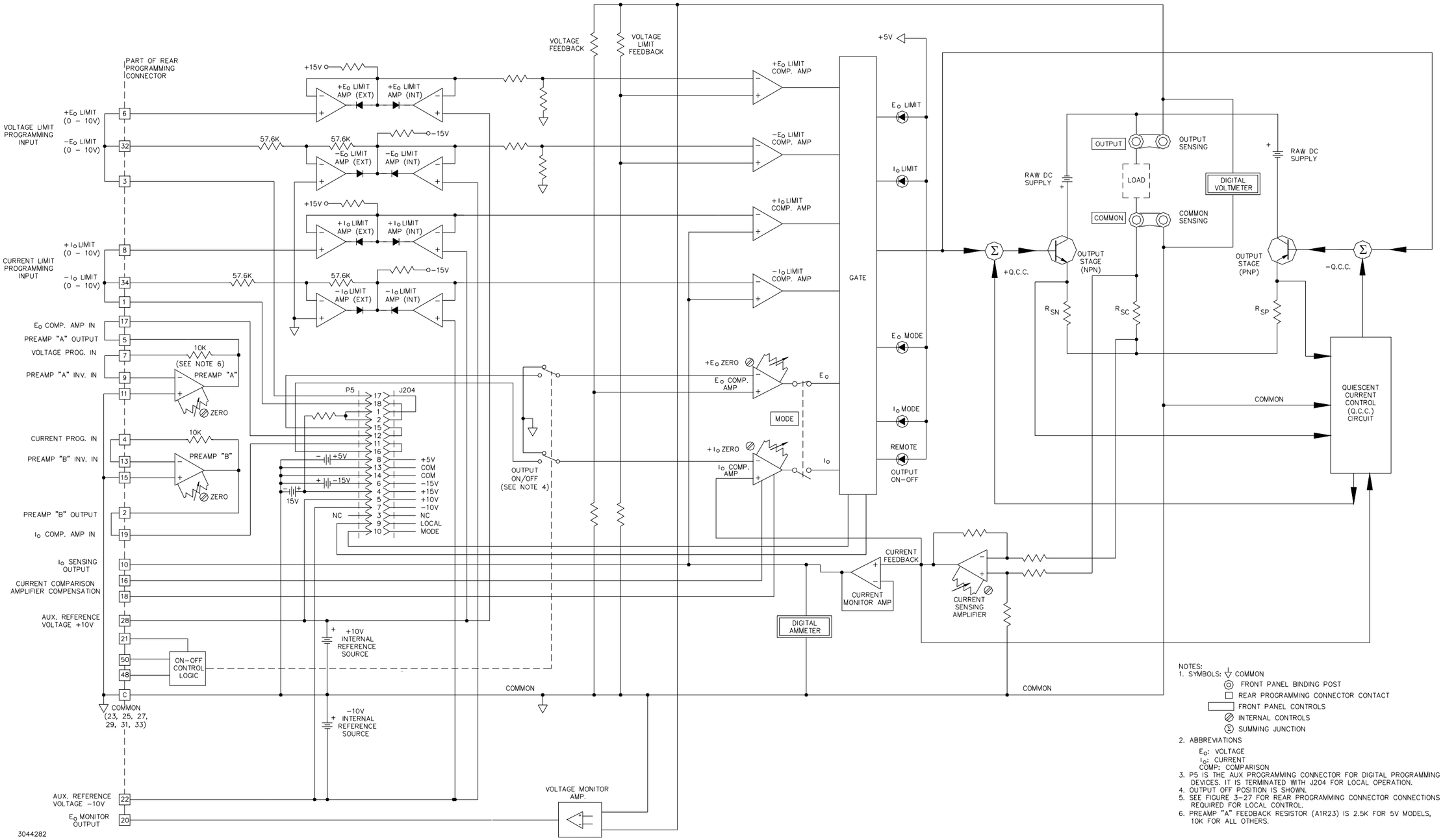
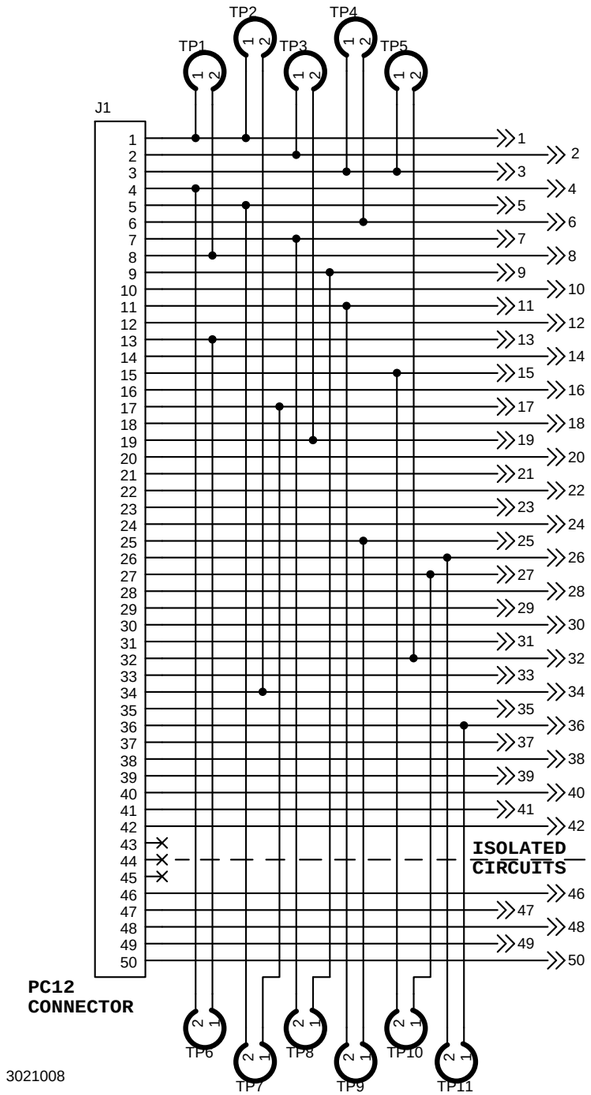


FIGURE 3-31. SIMPLIFIED SCHEMATIC DIAGRAM, CHANNEL 1 OR 2, BOP 2X SERIES



NOTES:

1. Connectors 195-0039 and 195-0121 both perform the same function. Connections shown are for PC 15, local control.

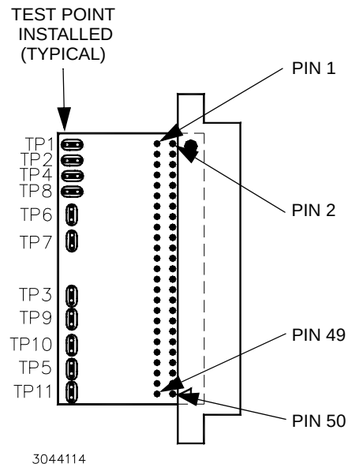
2. For P/N 195-0039, wires are added directly to the terminals for local control. User-added wires or components are connected directly to the terminals.

3. For P/N 195-0121 connections are made via the PWB traces and test points installed at the factory. The following table lists the pins connected when the test points are installed. For complete PWB connections, refer to the schematic diagram at the left.

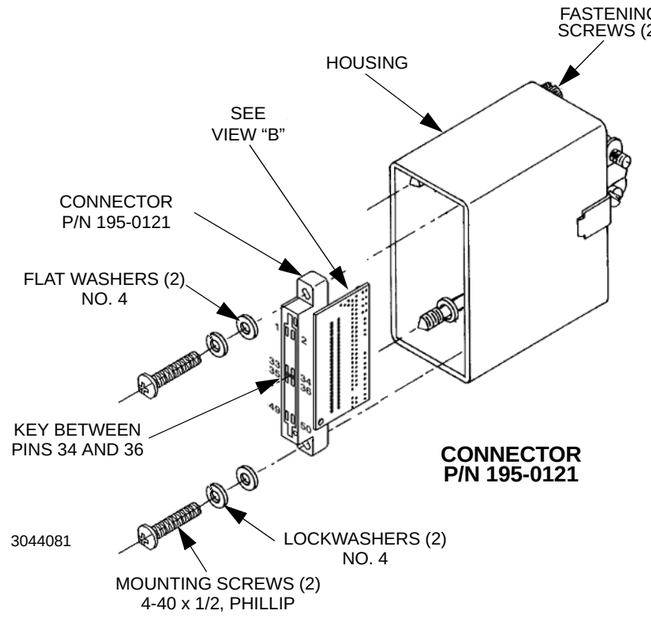
TEST POINT	PINS CONNECTED WHEN TEST POINT INSTALLED	
TP1	PIN 1	PIN 8
TP2	PIN 1	PIN 34
TP3	PIN 2	PIN 19
TP4	PIN 3	PIN 6
TP5	PIN 3	PIN 32
TP6	PIN 4	PIN 13
TP7	PIN 5	PIN 17
TP8	PIN 7	PIN 9
TP9	PIN 11	PIN 25
TP10	PIN 15	PIN 27
TP11	PIN 26	PIN 36

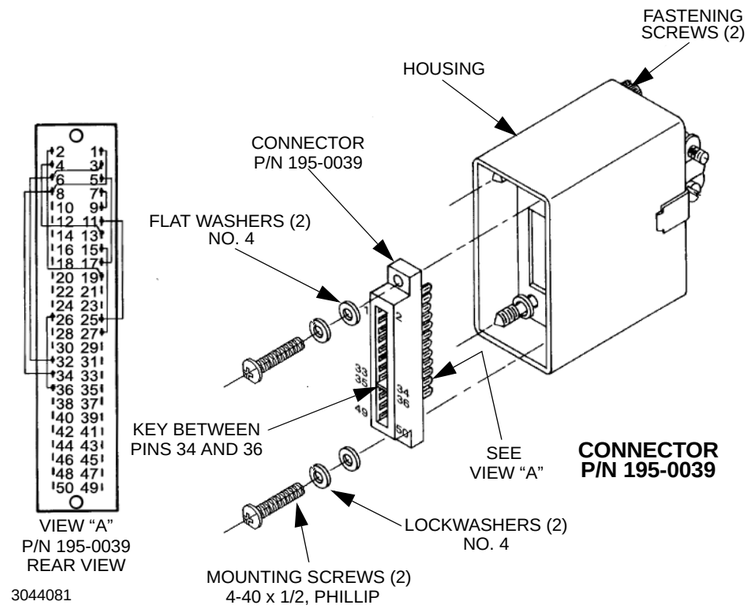
4. For P/N 195-0121 if necessary to isolate any pins connected by a test point, cut the test point. **CAUTION: Improper soldering technique may damage the PWB beyond repair.** Holes are provided at the test points for user-added wires or components. In addition, three holes per pin are provided on the PWB for user-added wires or components.

5. PC 12 (unwired) is supplied either without wires or without test points, depending on the connector used.



VIEW "B"
P/N 195-0121
PIN AND TEST POINT LOCATIONS





VIEW "A"
P/N 195-0039
REAR VIEW

3044081

FIGURE 3-32. PC 15/PC 12 REAR PROGRAMMING CONNECTOR INTERCONNECTIONS

090123

SECTION 4 - CALIBRATION

4.1 INTRODUCTION

WARNING

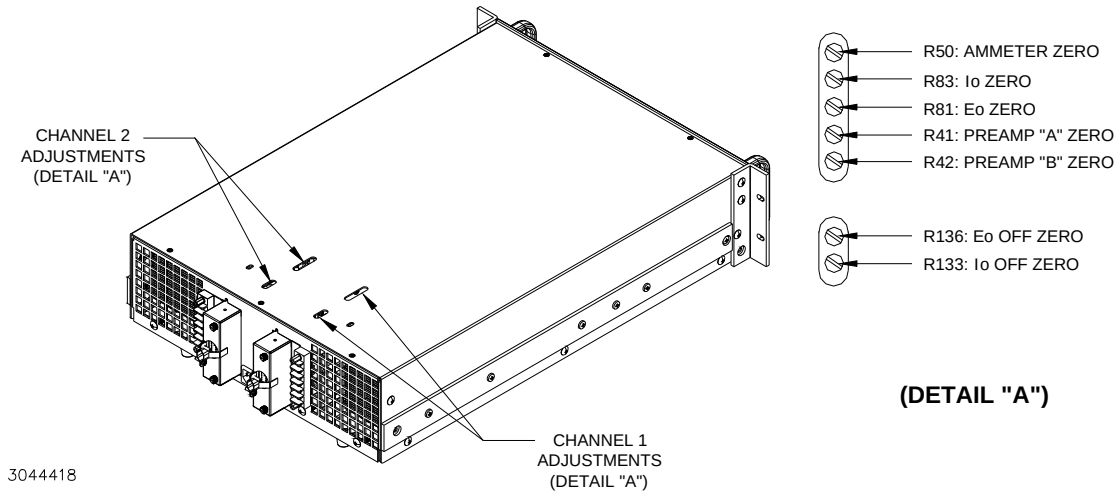
Hazardous voltage is present within the unit when powered on while the cover is removed. Removal of the cover is permitted only by authorized Service personnel. Procedures and instructions found in this section are to be performed by authorized Service personnel only.

Complete calibration and adjustment procedures for either BOP 2X-D channel are found n PAR. 4.3 and are listed in Table 4-1. The locations of all internal controls are illustrated In Figure 4-1. Unless otherwise specified, procedures apply to one channel of DG suffix models as well as DGL and DGC options.

In addition, calibration can also be accomplished remotely using the GPIB interface as described in PAR. 4.4. Separate procedures are provided to either calibrate the unit issuing SCPI commands manually (PAR. 4.4.1) or using an IVI driver 4.4.2).

TABLE 4-1. INTERNAL ADJUSTMENTS AND CALIBRATIONS

Reference Designation	Control Name	Purpose	Adjustment Procedure
R31, R32	±10V CAL	Reference Voltage Calibration	PAR. 4.3.1
R41	PREAMP “A” ZERO	Uncommitted Amplifier “A” Zero Adjustment	PAR. 4.3.2
R42	PREAMP “B” ZERO	Uncommitted Amplifier “B” Zero Adjustment	PAR. 4.3.3
R81	E_O COMP AMP ZERO	Voltage Channel Zero Adjustment	PAR. 4.3.4
R50	Ammeter Zero	Sensing Amplifier Offset Adjustment	PAR. 4.3.5
R83	I_O COMP AMP ZERO	Current Channel Zero Adjustment	PAR. 4.3.6
R133 (1)	I_0 OFF ZERO	Current Channel Zero Adjustment for OFF status	PAR. 4.3.6
R136 (1)	E_0 OFF ZERO	Voltage Channel Zero Adjustment for OFF status	PAR. 4.3.6
R316	I_O FS	Full Scale Output Current Calibration	PAR. 4.3.7
(1) Output OFF (disabled) status must be applied at ON-OFF CONTROL for this adjustment.			



NOTES
1. See PAR. 3.2.2 and Figure 3-32 for Rear Programming Connector details.
2. See PAR. 4.3.7 and Figure 4-3 for Adjustment of R316.

FIGURE 4-1. LOCATIONS OF ADJUSTMENT AND CALIBRATION CONTROLS

4.2 TEST EQUIPMENT REQUIREMENTS

Table 4-3 lists sense resistors recommended for measuring current and includes Kepco and Manufacturer's part numbers. The value of the sense resistor chosen should be known with 0.001% accuracy. If other than a recommended sense resistor is to be used, it must be rated for at least 100W power dissipation (actual power dissipation will be less than 10W). The thermal coefficient of the sense resistor chosen should be 10 ppm/°C or better. The sense resistor is to be mounted on a heat sink with a minimum surface area of 36 square inches, capable of dissipating 10 times the power rating of the sense resistor.

The digital voltmeter (DVM) used to measure output voltage and the voltage across the current sensing resistor should have at least six digits, multiple ranges, the smallest resolution must be at least one microvolt and the accuracy must be at least 0.0001%.

To calibrate the external programming capabilities of the BOP 2X-DG a DC reference voltage with the following minimum specifications is required.

- Value: 0.0V ±0.1mV and 10.0V ±0.1mV (either positive and negative or floating)
- Stability: better than 10^{-5}
- Internal Impedance: better than 0.1 Ohms
- Output Current: >0.5mA when sourcing, >0.15mA when sinking

An IEEE 488 System Controller, (with appropriate software) to be connected to the BOP 2X-DG Power Supply with an IEEE 488 cable.

TABLE 4-2. SUGGESTED SENSE RESISTORS

MODEL	VALUE	KEPCO PART NO.	MANUFACTURER	MANUFACTURER PART NO.
BOP 2X5-20D BOP 2X5-30D BOP 2X20-5D BOP 2X20-10D BOP 2X36-6D	0.001 OHM	115-3033	ISOTEK	RUG-Z-R001-0.1 TK10
BOP 2X50-2D BOP 2X50-4D BOP 2X72-3D BOP 2X100-1D BOP 2X100-2D	0.01 OHM	115-2997	ISOTEK	RUG-Z-R010-0.1 TK10
NOTE: Selected sense resistor must be mounted on a heatsink with a minimum surface area of 36 square inches to maintain thermal stability during calibration; forced cooling is recommended. Kepco Heatsink P/N 136-0451 will provide adequate cooling for the sense resistor.				

Refer to Figure 4-2 for equipment connections.

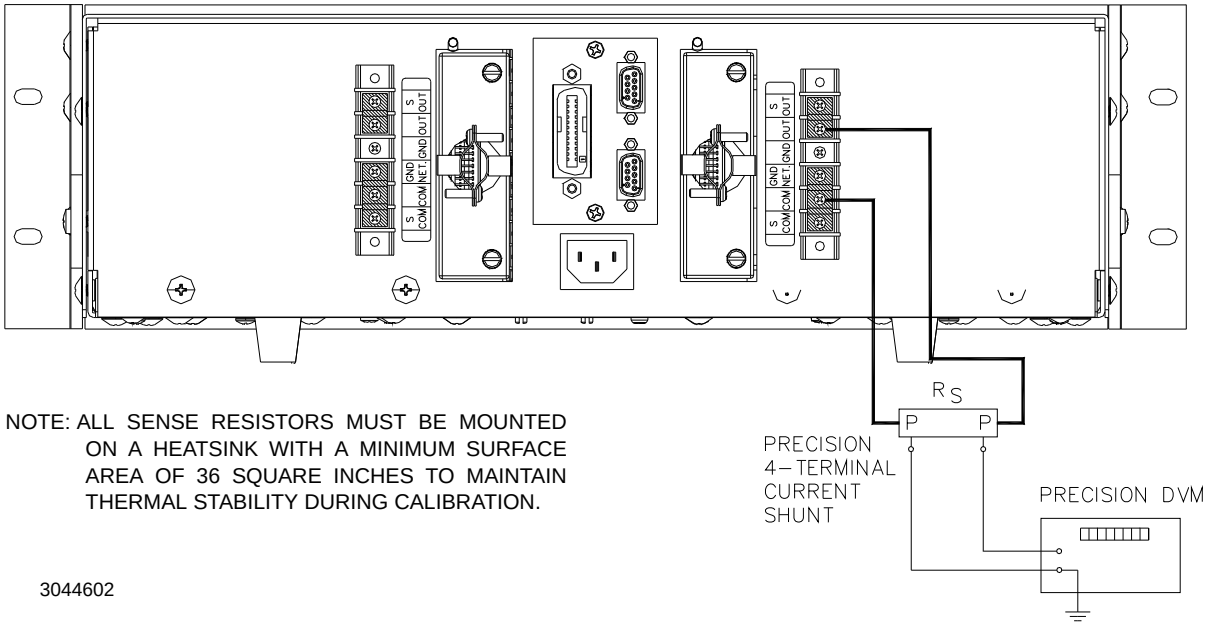


FIGURE 4-2. CURRENT SHUNT CONNECTIONS

4.3 INTERNAL ADJUSTMENTS AND CALIBRATIONS

See Figure 4-1 for location of adjustment controls. Refer to Figure 4-2 for sense resistor and DVM connections. For complete calibration of the unit it is recommended that the adjustments be done in the order presented.

NOTES: 1. These controls are factory calibrated. Recalibration is necessary only if components related to the circuitry involving these controls have been replaced or if there are indications that the unit is out of calibration.

2. Before using the BOP 2X-D channels in any application, please refer to Section 2 to get acquainted with the operating controls, a-c power requirements and information on load connections and grounding.

3. The Rear Programming Connector for each channel must be installed at all times.

4.3.1 ADJUSTMENT OF ±10 VOLT CALIBRATION CONTROLS (R31, R32)

1. Turn both BOP 2X-DG channels off and disconnect the unit from source power. Then remove the BOP 2X cover: 9 screws on each side, 1 screw at front, 1 screw on top

NOTE: The following procedure is applicable to either channel.

2. With the BOP 2X-DG powered off connect a DVM (see PAR. 4.2) to the Rear Programming Connector of the channel to be calibrated, from COMMON (pin 23, 25, 27, 29, 31 or 33) to pin 28 (+10V) and subsequently to pin 22 (–10V).

3. Locate the calibrating controls for the corresponding channel: R31 (+10V) and R32 (-10V). See Figure 4-1 for location. Turn on BOP channel to be calibrated.

4. First adjust R31 control to +10 Volts ±0.1mV on the digital voltmeter.

5. Move the DVM from pin 28 of the Rear Programming connector to pin 22 and adjust R32 to –10V ±0.1mV.

6. Turn off BOP channel.

7. Repeat steps 2 through 6 for the other channel.

4.3.2 PREAMPLIFIER “A” ZERO (R41)

NOTE: The following procedure is applicable to either channel.

1. Before adjusting Preamplifier “A” Zero control, turn both BOP channels off and configure the preamplifier of the channel to be calibrated: set amplification factor as desired, either inverting or non-inverting,

NOTE: Whenever Preamplifier “A” configuration is changed, this Zero control R41 should be readjusted.

2. Without a load connected to the output of the BOP 2X-D channel to be calibrated, place a short circuit across the input of Preamplifier “A.” For the default inverting configuration the short circuit is placed across the Rear Programming Connector of the channel to be calibrated from pin 9 to pin 11.

3. Set BOP channel to Voltage Mode by removing the connection between the corresponding Rear Programming Connector pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33).

4. Connect the DVM to the same Rear Programming Connector from COMMON (pin 23, 25, 27, 29, 31 or 33) to pin 5.

5. Locate the Preamplifier “A” Zero control R41 for the channel to be calibrated (see Figure 4-1).

6. Turn on BOP channel and adjust R41 for 0V ±0.1mV on the DVM.

7. Turn off BOP channel.

8. Repeat steps 1 through 7 to calibrate the other channel.

4.3.3 PREAMPLIFIER “B” ZERO (R42)

NOTE: The following procedure is applicable to either channel.

1. Before adjusting Preamplifier “B” Zero control, turn both BOP channels off and configure the preamplifier of the channel to be calibrated: set amplification factor as desired, either inverting or non-inverting,

NOTE: Whenever Preamplifier “B” configuration is changed, this Zero control R42 should be readjusted

2. Without a load connected to the output of the BOP channel to be calibrated, place a short circuit across the input of Preamplifier “B.” For the default inverting configuration the short circuit is placed across the Rear Programming Connector of the channel to be calibrated from pin 13 to pin 15.

3. Set BOP channel to Voltage Mode by removing the connection between the corresponding Rear Programming Connector pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33).

4. Connect the DVM to the corresponding Rear Programming Connector from COMMON (pin 23, 25, 27, 29, 31 or 33) to pin 2.

5. Locate the Preamplifier “B” Zero control R42 for the channel to be calibrated (see Figure 4-1).

6. Turn on BOP channel and adjust R42 for 0V ±0.1mV on the DVM.

7. Turn off BOP channel.

8. Repeat steps 1 through 7 for the other channel,

4.3.4 VOLTAGE CHANNEL ZERO ADJUSTMENT (R81)

NOTE: The following procedure is applicable to either channel.

1. Turn off both BOP channels. Without a load connected to the BOP channel output to be calibrated, connect a digital voltmeter to terminal 1 (OUTPUT SENSE) of TB 201 on the rear panel of the channel to be calibrated (see Figure 2-2), with DVM return to TB 201 terminal 6 (COMMON SENSING).

2. Set BOP channel to Voltage Mode by removing the connection between the corresponding Rear Programming Connector pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33).

3. Place a short circuit across the input of Preamplifier "A" of the channel to be calibrated. For the default inverting configuration the short circuit is placed across the corresponding Rear Programming Connector from pin 9 to pin 11.

4. Turn BOP channel on and locate the corresponding Voltage Channel Zero Adjustment (R81, see Figure 4-1).

5. Adjust R81 for output voltage of 0V ± (0.00001 x $E_O$nom) where $E_O$nom is the nominal (rated) output voltage of the BOP channel (in Volts). So, e.g., for BOP 2X50-4DG the tolerance is 0.00001 x 50 = ±0.0005V.

6. Turn off BOP channel.

7. Repeat steps 1 through 7 for the other channel,

4.3.5 AMMETER ZERO CONTROL ADJUSTMENT (R50)

NOTE: The following procedure is applicable to either channel.

1. Without a load connected to the BOP channel output to be calibrated, connect a digital voltmeter to the corresponding Rear Programming Connector from COMMON (pin 23, 25, 27, 29, 31 or 33) to pin 10.

2. Set BOP channel to Voltage Mode by removing the connection between the corresponding Rear Programming Connector, pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33).

3. Locate AMMETER ZERO control for the channel to be calibrated (R50, see Figure 4-1) and turn corresponding BOP 2X-D channel on.

4. Adjust corresponding R50 control to 0 Volts ±0.1mV on the digital voltmeter.

5. Repeat steps 1 through 4 for the other channel,

4.3.6 CURRENT CHANNEL ZERO ADJUSTMENT (R83)

NOTE: The following procedure is applicable to either channel.

1. Turn off BOP channel to be calibrated. Without a load connected to the BOP channel output, connect a precision 4-terminal current sensing shunt resistor (see Table 4-3) across the corresponding OUTPUT and COMMON terminals at the rear panel, Figure 2-2).

2. Connect the DVM to the shunt's sensing terminals.

3. Set BOP channel to Current Mode by installing a connection between the corresponding Rear Programming Connector, pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33).

4. Place a short circuit across the input of Preamplifier "B" of the channel to be calibrated, For the default inverting configuration the short circuit is placed across the corresponding Rear Programming Connector from pin 13 to pin 15.

5. Locate Current Channel Zero Adjustment for the channel to be calibrated (R83, see Figure 4-1) and turn BOP channel on.

6. Adjust corresponding R83 for output voltage of 0V ± (0.0001 x $I_O$nom x R_S)V where $I_O$nom is the nominal (rated) output current of the BOP channel (in Amps) and R_S is the resistance of the current sensing shunt resistor (in Ohms). So, e.g., for BOP 2X50-4DG with a 0.01-Ohm shunt resistor the tolerance is 0.0001 x 4 x 0.01 = ±0.00004V.

7. Turn off BOP channel.

8. Repeat steps through 7 for the other channel,

4.3.7 FULL SCALE OUTPUT CURRENT CALIBRATION (R316)

NOTE: The following procedure is applicable to either channel.

1. Turn both BOP 2X-D channels off and disconnect the unit from source power. Then remove the BOP cover: 4 screws at each side (1 towards the rear, 3 at the bottom), 5 screws on top (2 in front, 3 at rear) plus 3 screws at each ear bracket.

2. Without a load connected to the BOP 2X-D channel output to be calibrated, connect a precision 4-terminal current sensing shunt resistor (see Table 4-3) across the corresponding OUTPUT and COMMON terminals at the rear panel, Figure 2-2).

3. Connect the DVM to the shunt's sensing terminals.

4. Connect –10V ±0.1mV reference voltage across the corresponding Rear Programming Connector via an external 10K Ohm resistor (R_{EXT}): –10V through R_{EXT} to pin 13, return to pin 15 as shown in Figure 3-3. R_{EXT} must be 10K Ohms ± 0.001% tolerance. If a 10K resistor of this tolerance is not available, measure the 10K Ohm resistor to be used with 0.1 Ohms accuracy.

5. Reconnect unit to source power and turn on BOP 2X-D channel. Locate R316 for the channel to be calibrated (see Figure 4-3).

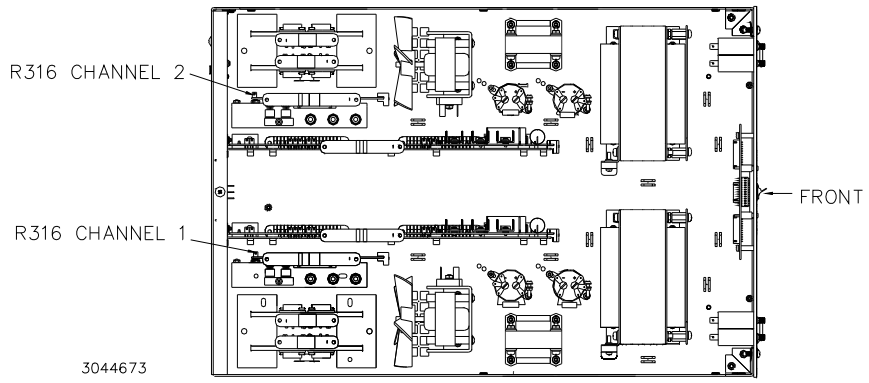


FIGURE 4-3. BOP 2X-DG POWER SUPPLY, TOP VIEW, COVER REMOVED, R316 LOCATION

CAUTION: This adjustment must be done with the cover removed to avoid damage to the power supply.

6. Adjust corresponding R316 so that DVM connected to current sensing shunt indicates ($I_O$nom x R_S)(R_{EXT}/10)V where
   $I_O$nom = nominal (rated) output current of the BOP 2X-D channel (in Amps)
   R_S = resistance of the current sensing shunt resistor (±0.001% tolerance, Ohms).
   R_{EXT} = measured value of 10k Ohm resistor (step 4 above) in kOhms.
   The total output current tolerance will ±0.003%. The 0.003% tolerance/accuracy comes from R_S tolerance = 0.001% + R_{EXT} tolerance = 0.001% + I_O adjustment/calibration tolerance = 0.001%.
   So, e.g., for BOP 2X50-4D with a known or measured 0.01-Ohm R_S and for R_{EXT} of 10k Ohms ±0.1 Ohms,, adjust R316 for (4 x 0.01)(10/10) = 0.04V ± 0.4μV.
   The resulting output current tolerance/accuracy will be ± (0.003%)(4A)= ± 0.12mA.

7. Repeat steps 1 through 6 for the other channel,

4.3.8 ADJUSTMENT OF I_O-OFF ZERO (A1R133) AND E_O-OFF ZERO (A1R136)

1. At the front panel of BOP channel to be calibrated, set the POWER switch to OFF.

2. At the rear panel of the BOP channel to be calibrated, connect a wire jumper between terminals 21 and 23 of the Rear Programming connector to set the output to OFF (disabled).

3. Connect precision shunt resistor across the BOP OUTPUT and COMMON terminals of the BOP channel to be calibrated. Connect DVM to the shunt's sensing terminals for output current measurements.

4. Set BOP channel to Current Mode by making a connection between the corresponding Rear Programming Connector pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33), then set the channel POWER switch at front panel to ON.

5. Adjust I_O-OFF ZERO (A1R133) potentiometer of the BOP 2X-D channel to be calibrated (see Figure 4-1) for 0A ±(0.01% of $I_O$nom) as read on the DVM connected to the shunt. E.g., for BOP 2X5-20D $I_O$nom is 20A and adjustment tolerance is ±2mA.

6. Turn off BOP channel and remove the shunt resistor from the output. Connect the DVM to the output sensing terminals of the BOP channel to be calibrated (OUT S and COM S).

7. Set BOP channel to Voltage Mode by removing the connection between the corresponding Rear Programming Connector pin 30 and COMMON (pin 23, 25, 27, 29, 31 or 33), then set the channel POWER switch at front panel to ON.

8. Adjust E_O-OFF ZERO (A1R136) of the BOP channel to be calibrated for 0V ±(0.01% of $E_O$nom) as read on DVM. E.g., for BOP 2X5-20DG $E_O$nom is 5V and adjustment tolerance is ±0.5mV.

9. Turn off BOP channel and repeat steps 1 through 8 above for the other channel.

4.4 CALIBRATION OF BOP 2X-DG CHANNEL USING GPIB INTERFACE

Calibration of the BOP 2X-DG Channel using the GPIB interface is performed using SCPI commands. The BOP-BIT Instrument driver, titled “Drivers for BOP with BIT 4886 Card” available at **www.kepcopower.com/drivers/drivers-dl3.htm#bit4886** provides a graphical interface with informational displays and prompts which lead you through the calibration of the channel. This VISA compliant driver works with many GPIB cards from suppliers like National Instruments and Hewlett-Packard.

Calibration using the IVI-COM driver is described in PAR. . The BOP 2X-DG channel can also be calibrated using the manual procedure described in PAR. 4.4.1.

NOTE: During calibration various circuits of the GPIB interface are verified. If an error occurs during the verification process, the interface card will beep and an error 240, Hardware, will be placed in the error queue.

The calibration values for different BOP 2X-DG Models are defined in Tables 4-1 and 4-2 for voltage and current, respectively. For each step requiring a measurement during calibration the tolerance of the measurements are included in Tables 4-1 and 4-2 and are not repeated in the individual steps.

Table 4-3 lists suggested sense resistors and includes Kepco and Manufacturer’s part numbers. The accuracy of these resistors is 0.1% and will result in a system accuracy of 0.11% if the listed values in Tables 4-1 and 4-2 are used. **For a system accuracy of 0.01%, measure the actual value of the sense resistor accurate to 7 places. Then calculate the expected values per Table 4-2 notes A-E; use with the listed tolerances of Table 4-2 to perform the calibration**.

TABLE 4-1. CALIBRATION MEASUREMENTS AND TOLERANCES - VOLTAGE

MODEL	LOW RANGE ZERO	VOLT ZERO	MAX OUTPUT	HIGH RANGE ACCURACY	LOW RANGE ACCURACY	LOW RANGE NOMINAL	VOLTAGE PROTECT ADJUST
	PAR. 4.4.1 STEP 2	PAR. 4.4.1 STEP 3	PAR. 4.4.1 STEPS 4, 5	PAR. 4.4.1 STEPS 6, 8	PAR. 4.4.1 STEPS 11, 13	PAR. 4.4.1 STEP 10	PAR. 4.4.1 STEP 15
	PAR. 4.4.2.4 STEP 5	PAR. 4.4.2.4 STEP 6	N/A	PAR. 4.4.2.4 STEPS 7, 8	PAR. 4.4.2.4 STEPS 9, 10	N/A	PAR. 4.4.2.4 STEP 11
BOP 2X5-20DD BOP 2X5-30DG	0V ±0.0003V	0V ±0.0003V	5V	5.0003V (MAX) -5.0003V (MIN) ±0.0003V	5.0003V (MAX) -5.0003V (MIN) ±0.0003V	5V	+5.003V ±0.003V
BOP 2X20-5DG BOP 2X20-10DG	0V ±0.0003V	0V ±0.0003V	20V	20.0015V (MAX) -20.0015V (MIN) ±0.0015V	5.0003V (MAX) -5.0003V (MIN) ±0.0003V	5V	+20.0035V ±0.0035V
BOP 2X36-6DG	0V ±0.0004V	0V ±0.0004V	36V	36.0027V (MAX) -36.0027V (MIN) ±0.0027	9.0006V (MAX) -9.0006V (MIN) ±0.0006	9V	+36.005V ±0.005V
BOP 2X50-2DG BOP 2X50-4DG	0V ±0.0005V	0V ±0.0005V	50V	50.0035V (MAX) -50.0035V (MIN) ±0.0035	12.5009V (MAX) -12.5009V (MIN) ±0.0009	12.5V	+50.007V ±0.007V
BOP 2X72-3DG BOP 2X72-6DG	0V ±0.001V	0V ±0.001V	72V	72.005V (MAX) -72.005V (MIN) ±0.005	18.0014V (MAX) -18.0014V (MIN) ±0.0014	18V	+72.0095V ±0.0095V
BOP 2X100-1DG BOP 2X100-2DG	0V ±0.001V	0V ±0.001V	100V	100.0075V (MAX) -100.0075V (MIN) ±0.0075	25.002V (MAX) -25.002V (MIN) ±0.002	25V	+100.0125V ±0.0125V

TABLE 4-2. CALIBRATION MEASUREMENTS AND TOLERANCES - CURRENT

MODEL	SENSE RESISTOR VALUE (SEE NOTE AND TABLE 4-3)	LOW CURRENT ZERO	MAIN CURRENT ZERO	MAIN CURRENT NOMINAL	HIGH CURRENT ACCURACY	LOW CURRENT NOMINAL	LOW CURRENT ACCURACY	CURRENT PROTECT ADJUST
		PAR. 4.4.1 STEP 17	PAR. 4.4.1 STEP 18	PAR. 4.4.1 STEPS 19, 20	PAR. 4.4.1 STEPS 21, 23	PAR. 4.4.1 STEP 24, 25	PAR. 4.4.1 STEPS 26, 28	PAR. 4.4.1 STEP 30
		PAR. 4.4.2.4 STEP 16	PAR. 4.4.2.4 STEP 17	N/A	PAR. 4.4.2.4 STEPS 18, 19	N/A	PAR. 4.4.2.4 STEPS 20, 21	PAR. 4.4.2.4 STEP 22
BOP 2X100-1DG[1]	1 Ohm	0V ±0.000025V	0V ±0.00005V	1V	1.000075V (MAX) -1.000075V (MIN) ±0.000075V	0.25V	0.25005V (MAX) -0.25005V (MIN) ±0.00005V	1.0013V ±0.0013V
BOP 2X50-2DG[2] **BOP 2X100-2DG**[2]	1 Ohm	0V ±0.00005V	0V ±0.0001V	2V	2.00015V (MAX) -2.00015V (MIN) ±0.00015V	0.5V	0.5001V (MAX) -0.5001V (MIN) ±0.0001V	2.00026V ±.00026V
BOP 2X72-3DG[2]	1 Ohm	0V ±0.000075V	0V ±0.00015V	3V	3.0003V (MAX) -3.0003V (MIN) ±0.0003V	0.75V	0.75015V (MAX) -0.75015V (MIN) ±0.00015V	3.0039V ±0.0039V

TABLE 4-2. CALIBRATION MEASUREMENTS AND TOLERANCES - CURRENT (CONTINUED)

MODEL	SENSE RESISTOR VALUE (SEE NOTE AND TABLE 4-3)	LOW CURRENT ZERO	MAIN CURRENT ZERO	MAIN CURRENT NOMINAL	HIGH CURRENT ACCURACY	LOW CURRENT NOMINAL	LOW CURRENT ACCURACY	CURRENT PROTECT ADJUST
		PAR. 4.4.1 STEP 17	PAR. 4.4.1 STEP 18	PAR. 4.4.1 STEPS 19, 20	PAR. 4.4.1 STEPS 21, 23	PAR. 4.4.1 STEP 24, 25	PAR. 4.4.1 STEPS 26, 28	PAR. 4.4.1 STEP 30
		PAR. 4.4.2.4 STEP 16	PAR. 4.4.2.4 STEP 17	N/A	PAR. 4.4.2.4 STEPS 18, 19	N/A	PAR. 4.4.2.4 STEPS 20, 21	PAR. 4.4.2.4 STEP 22
BOP 2X50-4DG[2]	1 Ohm	0V ±0.0001V	0V ±0.0002V	4V	4.0004V (MAX) -4.0004V (MIN) ±0.0004V	1V	1.0002V (MAX) -1.0002V (MIN) ±0.0002V	4.0007V ±0.0007V
BOP 2X20-5DG[2]	0.1 Ohm	0V ±0.0000125V	0V ±0.000025V	0.5V	0.5000375V (MAX) -0.5000375V (MIN) ±0.0000375V	0.125V	0.125025V (MAX) -0.125025V (MIN) ±0.000025V	0.50065V ±0.00065V
BOP 2X36-6DG[2]	0.1 Ohm	0V ±0.000015V	0V ±0.00003V	0.6V	0.600045V (MAX) -0.600045V (MIN) ±0.000225V	0.15V	0.15003V (MAX) -0.15003V (MIN) ±0.00003V	0.60075V ±0.00075V
BOP 2X20-10DG[2]	0.1 Ohm	0V ±0.000025V	0V ±0.00005V	1.0V	1.000075V (MAX) -1.000075V (MIN) ±0.000075V	0.25V	0.25005V (MAX) -0.25005V (MIN) ±0.00005V	1.00013V ±0.00013V
BOP 2X5-20DG[2]	0.001 Ohm	0V ±0.00000005V	0V ±0.0000001V	0.02V	0.02000015V (MAX) -0.02000015V (MIN) ±0.00000015V	0.005V	0.0050001V (MAX) -0.0050001V (MIN) ±0.0000001V	0.02000025V ±0.0000025V
BOP 2X5-20DG[2]	0.001 Ohm	0V ±0.00000005V	0V ±0.0000001V	0.03V	0.03000015V (MAX) -0.03000015V (MIN) ±0.00000015V	0.0075V	0.0075001V (MAX) -0.0075001V (MIN) ±0.0000001V	0.0300025V ±0.0000025V
REFER TO NOTES IF EXACT SENSE RESISTOR VALUE IS KNOWN.	**R_{SENSE}** SEE NOTE.	SEE NOTE A.	SEE NOTE B.	SEE NOTE C.	SEE NOTES C AND D	SEE NOTE E.	SEE NOTES E AND B.	SEE NOTE F.
USE SPACE PROVIDED AT RIGHT FOR CAL-CULATED VALUES.								

TABLE 4-2. CALIBRATION MEASUREMENTS AND TOLERANCES - CURRENT (CONTINUED)

MODEL	SENSE RESISTOR VALUE (SEE NOTE AND TABLE 4-3)	LOW CURRENT ZERO	MAIN CURRENT ZERO	MAIN CURRENT NOMINAL	HIGH CURRENT ACCURACY	LOW CURRENT NOMINAL	LOW CURRENT ACCURACY	CURRENT PROTECT ADJUST
		PAR. 4.4.1 STEP 17	PAR. 4.4.1 STEP 18	PAR. 4.4.1 STEPS 19, 20	PAR. 4.4.1 STEPS 21, 23	PAR. 4.4.1 STEP 24, 25	PAR. 4.4.1 STEPS 26, 28	PAR. 4.4.1 STEP 30
		PAR. 4.4.2.4 STEP 16	PAR. 4.4.2.4 STEP 17	N/A	PAR. 4.4.2.4 STEPS 18, 19	N/A	PAR. 4.4.2.4 STEPS 20, 21	PAR. 4.4.2.4 STEP 22

NOTES: 1. The following formulas are used to calculate proper calibration values if the exact Sense Resistor value is known.

R_{SENSE} = the measured value of the sense resistor to 6 places (minimum).

I = Rated current output of BOP (e.g., for BOP 2X50-4DE, I = 4)

A. (R_{SENSE}) (I/4) (0.00005)

B. (R_{SENSE}) (I/4) (0.0001) [FOR MODELS WHERE I = 3 OR 4 USE (R_{SENSE})(I/4) (0.00015)]

C. (R_{SENSE})(I)

D. (R_{SENSE})(I) (0.00075) [FOR MODELS WHERE I = 3 OR 4 USE (R_{SENSE})(I) (0.00012)]

E. (R_{SENSE}) (I/4)

F. [(R_{SENSE}) (Iomax)]/3800

2. D suffix = Digital meter, G suffix = GPIB Interface.

TABLE 4-3. SUGGESTED SENSE RESISTORS

FOR BOP WITH RATED CURRENT	USE SENSE RESISTOR VALUE	KEPCO PART NO.	MANUFACTURER	MANUFACTURER PART NO.
20A	0.001 OHM, 30W	KT 3131	ISOTEK	RTO-B-R001-1
12A	0.01 OHM, 30W	KT 3130	ISOTEK	RTO-B-R01-1
5A, 6A, 8A, 10A	0.1 OHM, 30W	KT 3126	ISOTEK	RTO-B-R1-1
1A, 2A, 3A, 4A	1 OHM, 30W	KT 3146	ISOTEK	RTO-B-1R00-1.0

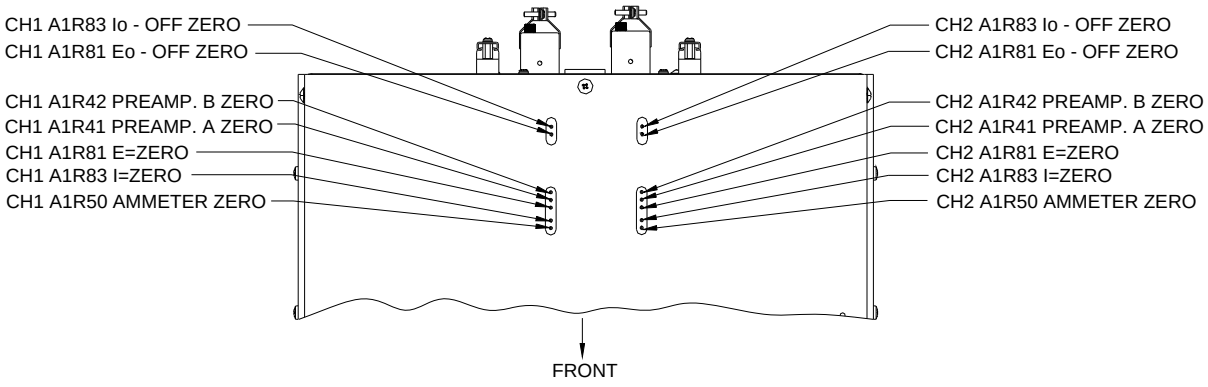
NOTE: ALL SENSE RESISTORS MUST BE MOUNTED ON A HEATSINK WITH A MINIMUM SURFACE AREA OF 36 SQUARE INCHES TO MAINTAIN THERMAL STABILITY DURING CALIBRATION (KEPCO P/N 136-0451 RECOMMENDED FOR KT SENSE RESISTORS LISTED).

4.4.1 MANUAL CALIBRATION

1. Initiate calibration by sending the SCPI command **SYSTem:PASSword:CENable DEFAULT** and then send **CAL:STATe 1**. The password (DEFAULT) has been set at the factory. If the password has been changed from DEFAULT, substitute the correct password for the unit in the **SYST:PASS:CEN** command. If the password has been forgotten, consult factory.

   If voltage calibration is not needed, proceed to step 16 for current calibration.

2. Set the BOP to zero volts output under the 1/4 range by sending **CAL:LVOLT ZERO**. Connect a Digital Voltmeter (DVM) to the BOP output to measure the power supply output. Adjust A1R81 on BOP until the DVM reads the voltage specified in Table 4-1 for LOW RANGE ZERO.



3044528

FIGURE 4-4. MANUAL CALIBRATION CONTROLS (TOP VIEW)

3. Set the BOP output voltage to zero by sending **CAL:VOLT ZERO**. Send the command **CAL:DPOT 1** to increase the voltage or **CAL:DPOT -1** to decrease the voltage until the DVM reads the voltage specified in Table 4-1 for VOLT ZERO.

4. Set the BOP to maximum positive output voltage by sending **CAL:VOLT MAX**. Measure the voltage output using a DVM of at least 7 digits in accuracy.

5. Decrease the voltage output by sending **CAL:DPOT -1** commands until the voltmeter reads less than the rated output voltage specified in Table 4-1 for MAX OUTPUT. Then send **CAL:DPOT 1** and verify the voltage is slightly above the rated output voltage specified in Table 4-1 for MAX OUTPUT

6. Referring to Table 4-1 (HIGH RANGE ACCURACY) for value and ± tolerance, send the command **CAL:DATA -10** to decrease the voltage until the voltage is less than the rated output. Send the command **CAL:DATA 1** to increase the voltage or the command **CAL:DATA -1** to decrease the voltage until the DVM reads the value specified in Table 4-1 for HIGH RANGE ACCURACY MAX.

7. Set the BOP to maximum negative output by sending **CAL:VOLT MIN**. Continue to measure the output of the supply using the DVM.

8. If the output is less negative than the value specified in Table 4-1 for HIGH RANGE ACCURACY MIN, send **CAL:DATA -100** and verify the output is now more negative. If the value is still not more negative, send **CAL:DPOT -1** to change the course adjustment and provide a more negative output.

   Send the command **CAL:DATA 10** to increase the voltage. Continue to send **CAL:DATA 10** until the voltage is less negative than the value specified in Table 4-1 for HIGH RANGE ACCURACY MIN. Send **CAL:DATA -1** until the reading is within the limits specified in Table 4-1 for HIGH RANGE ACCURACY MIN.

   If the **CAL:DPOT** command was used to adjust the negative output, send **CAL:VOLT MAX** to the unit and repeat step 6 before proceeding to step 9.

9. Set the BOP to 1/4 scale (low) range positive output voltage by sending **CAL:LVOLT MAX**. Output voltage should be as specified in Table 4-1 for LOW RANGE NOMINAL. Measure the voltage output using a Digital Voltmeter of at least 7 digits in accuracy.

10. Decrease the voltage output by sending **CAL:DPOT -1** commands until the voltmeter reads less than the rated output. Then send **CAL:DPOT 1** and verify the voltage is slightly above value specified in Table 4-1 for LOW RANGE NOMINAL.

11. Send the command **CAL:DATA -10** to decrease the voltage until the voltage is less than value specified in Table 4-1 for LOW RANGE ACCURACY MAX. Send the command **CAL:DATA 1** to increase the voltage or the command **CAL:DATA -1** to decrease the voltage to obtain the value specified in Table 4-1 for LOW RANGE ACCURACY MAX.

12. Set the BOP to maximum negative 1/4 scale range output by sending **CAL:LVOLT MIN**. Continue to measure the output of the supply.

13. If the output is less negative than the value specified in Table 4-1 for LOW RANGE ACCURACY MIN, send **CAL:DATA -100** and verify the output is now more negative. If the value is not more negative, Send **CAL:DPOT -1** to change the course adjustment and provide a more negative output.

    Send the command **CAL:DATA 10** to increase the voltage. Continue to send **CAL:DATA 10** until the voltage is less negative than the value specified in Table 4-1 for LOW RANGE ACCURACY MIN. Send **CAL:DATA -1** until the reading is within the limits specified in Table 4-1 for LOW RANGE ACCURACY MIN.

    If the **CAL:DPOT** command was used to adjust the negative output, send **CAL:LVOLT MAX** to the unit and repeat step 11 before proceeding to step 14.

14. Set the BOP to voltage limit by sending **CAL:VPR MAX**. Continue to measure the output of the BOP.

15. Send the command **CAL:DATA -10** to decrease the voltage until the output voltage measured is within, or close to, the tolerance specified in Table 4-1 for VOLTAGE LIMIT ADJUST. Send the command **CAL:DATA 1** to increase the voltage or **CAL:DATA -1** to decrease the voltage as necessary until the measured value is within the tolerance specified in Table 4-1 for VOLTAGE LIMIT ADJUST.

NOTE: If Current calibration is not required, proceed to step 31.

16. Send **CAL:ZERO** to prepare for current calibration. After sending the command, the BOP output will be set to zero volts. Connect the Kelvin type sense resistor to the BOP output using a heat sink capable of dissipating 10 times rated power of sense resistor. Connect the DVM to the sensing terminals of the Kelvin type sense resistor as shown in Figure 4-2. Table 4-2 provides suggested sense resistor values for various BOP current outputs, as well as the formula for calculating expected measured values and tolerances for any sense resistor where the precise resistance is known. Table 4-3 lists possible sources for obtaining the suggested sense resistors

17. Set the BOP to zero volts across the sense resistor (corresponding to zero current) under the low (1/4 Scale) current range by sending **CAL:LCURR ZERO**. Adjust A1R83 on BOP until the DVM reads the voltage specified in Table 4-2 for LOW CURRENT ZERO.

18. Set the BOP current to 0 Amps in current mode by sending **CAL:CURR ZERO**. Use the command **CAL:DPOT 1** to increase the current or **CAL:DPOT -1** to decrease the current until the DVM reads the voltage specified in Table 4-2 for MAIN CURRENT ZERO.

19. Set the BOP to maximum positive output current by sending **CAL:CURR MAX**. Measure the voltage across the sense resistor using a Digital Voltmeter of at least 7 digits in accuracy. Verify the DVM reads the voltage specified in Table 4-2 for MAIN CURRENT NOMINAL. The voltage shown on the DVM should be positive; if not, reverse the leads to the DVM.

20. Decrease the voltage across the sense resistor by sending **CAL:DPOT -1** commands until the voltmeter reads less than the value specified in Table 4-2 for MAIN CURRENT NOMINAL. Then send **CAL:DPOT 1** and verify the voltage is slightly above the specified voltage.

21. Send the command **CAL:DATA -10** to decrease the voltage across the sense resistor until the voltage is less than the value specified in Table 4-2 for HIGH CURRENT ACCURACY MAX. Send the command **CAL:DATA 1** to increase the voltage or **CAL:DATA -1** to decrease the voltage until the DVM reads the voltage specified in Table 4-2 for HIGH CURRENT ACCURACY MAX.

22. Set the BOP to maximum negative output by sending **CAL:CURR MIN**. Continue to measure the voltage across the sense resistor.

23. If the voltage across the sense resistor is less negative than the value specified in Table 4-2 for HIGH CURRENT ACCURACY MIN, send **CAL:DATA -100** and verify the output is now more negative. If the value is not more negative than the value found in Table 4-2 for HIGH CURRENT ACCURACY MIN, send **CAL:DPOT -1** to change the course adjustment and provide a more negative output.

    Send the command **CAL:DATA 10** to increase the voltage across the sense resistor. Continue to send **CAL:DATA 10** until the voltage is less than the value specified in Table 4-2 for HIGH CURRENT ACCURACY MIN. Send **CAL:DATA -1** until the DVM reads the voltage specified in Table 4-2 for HIGH CURRENT ACCURACY MIN.

    If the **CAL:DPOT** command was used to adjust the negative output, send **CAL:CURR MAX** to the unit and repeat step 21 before proceeding to step 24.

24. Set BOP to the maximum current at low range (1/4 Scale) by sending **CAL:LCURR MAX**. Verify the DVM reads the voltage specified in Table 4-2 for LOW CURRENT NOMINAL.

25. Decrease the voltage across the sense resistor by sending **CAL:DPOT -1** commands until the voltmeter reads less than the value specified in Table 4-2 for LOW CURRENT NOMINAL. Then send **CAL:DPOT 1** and verify the voltage is slightly above this value.

26. Send the command **CAL:DATA -10** to decrease the voltage across the sense resistor until the voltage is less than the value specified in Table 4-2 for LOW RANGE ACCURACY MAX. If necessary, send the command **CAL:DATA 1** to increase the voltage across the sense resistor or **CAL:DATA -1** to decrease the voltage across the sense resistor until the DVM reads the voltage specified in Table 4-2 for LOW RANGE ACCURACY MAX.

27. Set the BOP to maximum negative 1/4 scale range output current by sending **CAL:LCURR MIN**. Continue to measure the voltage across the sense resistor.

28. If the voltage across the sense resistor is less negative than the value specified in Table 4-2 for LOW RANGE ACCURACY MIN, send **CAL:DATA -100** and verify the output is now

more negative. If the value is not more negative, Send **CAL:DPOT -1** to change the course adjustment and provide a more negative output.

Send the command **CAL:DATA 10** to increase the voltage across the sense resistor. Continue to send **CAL:DATA 10** until the voltage is less than the value specified in Table 4-2 for LOW RANGE ACCURACY MIN. Send **CAL:DATA -1** until the reading is within the limits specified in Table 4-2 for LOW RANGE ACCURACY MIN.

If the **CAL:DPOT** command was used to adjust the negative output, after adjusting the negative supply output, send **CAL:LCURR MAX** to the unit and repeat step 26 before proceeding to step 29.

29. Set the BOP to current limit by sending **CAL:CPR MAX**. Continue to measure the output of the BOP.

30. Send the command **CAL:DATA -10** to decrease the voltage across the sense resistor until the output voltage measured is within, or close to, the tolerance specified in Table 4-2 for CURRENT LIMIT ADJUST. Send the command **CAL:DATA 1** to increase the voltage or **CAL:DATA -1** to decrease the voltage as necessary until the measured value is within the tolerance specified in Table 4-1 for CURRENT LIMIT ADJUST.

31. Save the calibration levels by sending **CAL:SAVE 2/21/2004**. The 9 characters at the end are optional and are the calibration date of the power supply. The date entered in this manner is reported in the ***idn?** query. This can be prove useful for checking calibration dates when periodic calibration is imposed by system or quality requirements.

4.4.2 CALIBRATION USING IVI DRIVER

Calibration of the BOP channel is performed using SCPI commands implemented through the Instrument driver. The driver provides a graphical interface with informational displays and prompts which lead you through the calibration of the Power Supply. This IVI compliant driver works with many GPIB cards from suppliers like National Instruments and Hewlett-Packard.

This driver requires a helper application (visa32.dll) to be installed on the computer being used. VISA uses resource strings (see Table 4-4) to address the unit.

TABLE 4-4. VISA RESOURCE STRING CORRESPONDING TO INTERFACE

INTERFACE	VISA RESOURCE STRING	COMMENT
GPIB	GPIB::xx::INSTR	The GPIB address replaces xx.
SERIAL	ASRLy::INSTR	The com port number replaces y.

4.4.2.1 SETUP

1. The following calibration procedure uses a calibration panel which is part of the IVI driver for the BIT 4886 which can be downloaded from the Kepco website at:
   www.kepcopower.com/drivers.htm
   Unzip the files and double-click on setup.exe to install the driver.

2. After the program is installed, double click on
   \Program Files\KepcodcPower\\KepcoControlPanel\KepcoControlPanel.exe
   to run the program. Two windows appear: the VISA Resource Descriptor Input panel (Figure 4-5) and the Main Control panel (with display blank) as shown in Figure 4-6.

3. Enter the VISA resource descriptor of the BOP-BIT into the VISA Resource Descriptor Input panel (Figure 4-5), or leave blank to use the default setting (GPIB::6), then click OK.

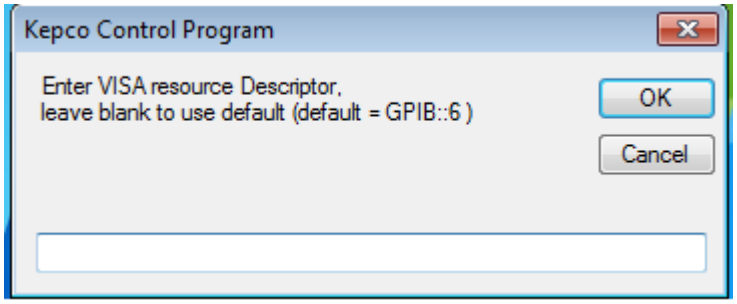


FIGURE 4-5. INPUT VISA RESOURCE DESCRIPTOR

4. The program will attempt to connect to the BOP channel. If connection is successful, the Main Control Panel display now shows the Instrument Model and Driver Revision as shown in Figure 4-6. If an **Initialization Error** message appears, either the IVI Shared components or VISA-COM was not installed correctly. Return to step 1 and reinstall the driver.

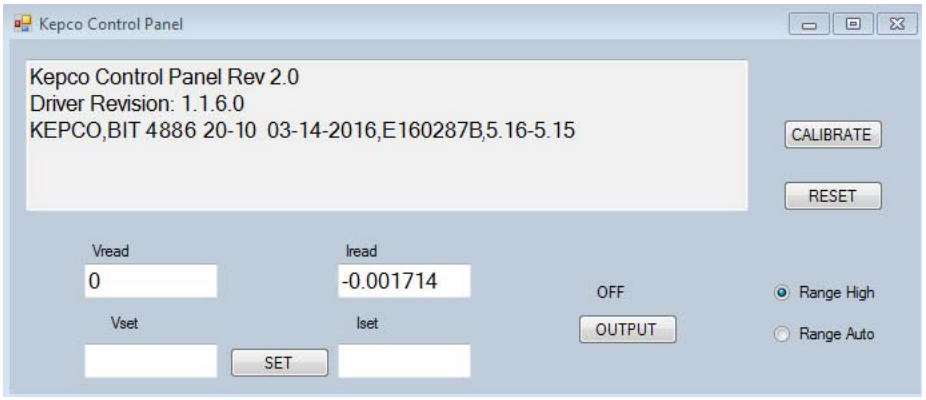


FIGURE 4-6. MAIN CONTROL PANEL, TYPICAL FOR BOP 2X20-10DG, CHANNEL 2

4.4.2.2 MAIN CONTROL PANEL

The main control panel allows setting the voltage and current, enabling and disabling the output, selecting the current range, resetting the unit, and calibrating the unit.

1. To set the unit's voltage and current, enter the values into the **Vset** and **Iset** boxes, respectively, and click the **SET** button.

2. To enable or disable the output, click the **OUTPUT** button.

3. To reset the unit to power-up state (output OFF, voltage and current set to 0), click the **RESET** button.

4. Range High and Range Auto radio buttons allow range selection.

4.4.2.3 CALIBRATION CONTROLS

CAUTION: Before performing calibration, set BOP OUTPUT to OFF (send OUTP OFF command (see PAR. B.20) and connect short across BOP output, then set OUTPUT to ON and Voltage to 10V. **It is necessary to wait 10 minutes before calibrating the unit to allow for thermal stabilization.** After 10 minutes, set OUTPUT to OFF and remove short from BOP output, then proceed with calibration.

All adjustments are done using the four arrow buttons that appear during Calibration (see Figure 4-7 and Table 4-5). The double arrow buttons **<<** and **>>** either increase (**>>**) or decrease (**<<**) the output by a maximum of 18 steps (each step is one LSB (Least Significant Bit), equivalent to 0.024% of nominal value) at a time; the **>** and **<** buttons either increase (**>**) or decrease (**<**) the output one step at a time.

FIGURE 4-7. CALIBRATION CONTROLS

TABLE 4-5. CALIBRATION PANEL FUNCTIONS

BUTTON OR WINDOW	Function
<< button	Coarse Adjustment - Decreases the output value by a maximum of LSBs.
>> button	Coarse Adjustment - Increases the output value by a maximum of 18 LSBs.
< button	Fine Adjustment - Decreases the output value by one LSB.
> button	Fine Adjustment - Increases the output value by one LSB.
OK button	Enters the value displayed in the text field and advances calibration to next step.

4.4.2.4 CALIBRATION PROCEDURE

1. Click on the **Calibrate** button on the main control panel (see Figure 4-6). This opens a calibration panel (see Figure 4-7 and Table 4-5) that allows calibration in either voltage or current mode.

2. After initializing, the password window appears (see Figure 4-8). The password (DEFAULT) has been set at the factory. If the password has been changed from DEFAULT, type the proper password in the text box and click **OK**. If the password has been forgotten, consult factory. If the password is correct, the calibration panel (Figure 4-7) is active and calibration can be accomplished by following the screen prompts.

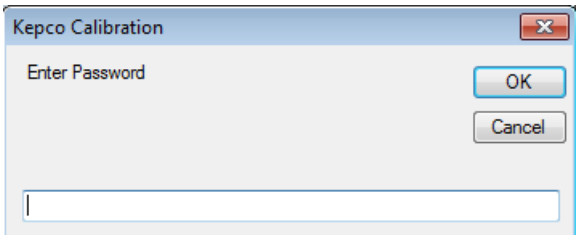


FIGURE 4-8. PASSWORD ENTRY

3. For Voltage Calibration, click **YES** button (see Figure 4-9).

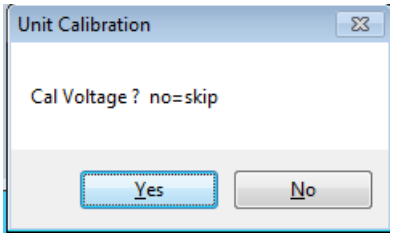


FIGURE 4-9. VOLTAGE CALIBRATION

4. The text window reads **`Connect DVM to Output Sense, Remove Shunt`**. Remove any connections between +OUT and –OUT. Connect DVM (high) to +OUT and DVM (low) to –OUT, then click **OK** button.

NOTE: Use external DVM for all measurements.

5. The text window reads **`LVOLT ZERO Adjust R81.`** Adjust A1R81 on BOP until the DVM reads the voltage specified in Table 4-1 for LOW RANGE ZERO, then click **OK** button.

6. The text window reads **`VOLT ZERO`**. Using the **<** and **>** buttons (see PAR. 4.4.2.3), adjust the output voltage until the DVM reads the voltage specified in Table 4-1 for VOLT ZERO, then click **OK** button.

7. The text window reads **`VOLT MAX.`** Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust the output voltage until DVM reading is as specified in Table 4-1 for HIGH RANGE ACCURACY then click **OK** button.

8. The text window reads **`VOLT MIN.`** Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust the output voltage until DVM reading is as specified in Table 4-1 for HIGH RANGE ACCURACY, then click **OK** button.

9. The text window reads **`LVOLT MAX.`** Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust the output voltage until DVM reads voltage specified in Table 4-1 for LOW RANGE ACCURACY, then click **OK** button.

10. The text window reads **`LVOLT MIN.`** Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust the output voltage until DVM reads voltage specified in Table 4-1 for LOW RANGE ACCURACY, then click **OK** button.

11. The text window reads **VPR MAX.** Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust the output voltage until DVM reads voltage specified in Table 4-1 for VOLTAGE PROTECT ADJUST, then click **OK** button.

12. The text window reads **VPR MIN - No adjustment, check only.** Click **OK** button to proceed to next step.

13. The text window reads **VOLT CAL DONE.** A separate window opens and reads **Cal Current? no=skip.** Click **YES** button to perform current calibration.

14. The text window reads **Connect shunt across output**. Connect precision shunt resistor (see PAR.) across +OUT and –OUT terminals. After connecting the shunt click **OK** button.

15. The text window reads **With shunt connected across output, connect DVM across shunt**. Connect DVM (high) to the shunt sensing terminal that correlates to +OUT and DVM (low) to the shunt sensing terminal that correlates to –OUT. Click **OK** button to proceed to next step.

16. The text window reads **LCURR ZERO Adjust R83**. Adjust output current until DVM reads voltage specified in Table 4-2 for LOW CURRENT ZERO, then click **OK** button.

17. The text window reads **CURR ZERO**. Using the fine adjust **<** and **>** buttons, adjust output until DVM reads voltage specified in Table 4-2 for MAIN CURRENT ZERO, then click **OK** button.

18. The text window reads **CURR MAX**. Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust output current until DVM reads voltage specified in Table 4-2 for HIGH CURRENT ACCURACY, then click **OK** button.

19. The text window reads **CURR MIN**. Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust output current until DVM reads voltage specified in Table 4-2 for HIGH CURRENT ACCURACY, then click **OK** button.

20. The text window reads **LCURR MAX**. Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust output current until DVM reads voltage specified in Table 4-2 for LOW CURRENT ACCURACY, then click **OK** button.

21. The text window reads **LCURR MIN**. Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust output current until DVM reads voltage specified in Table 4-2 for LOW CURRENT ACCURACY, then click **OK** button.

22. The text window reads **CPR MAX**. Using the coarse **<<** and **>>** and fine **<** and **>** buttons, adjust output current until DVM reads voltage specified in Table 4-2 for CURRENT PROTECT ADJUST, then click **OK** button.

23. The text window reads **CPR MIN - No Adjustment, check only**. Click **OK** button to proceed to next step.

24. The text window reads **CAL Done, Disconnect Shunt**. Click **OK**. Current calibration is complete and automatically saved. Disconnect the external shunt and DVM.

APPENDIX A - SCPI COMMON COMMAND/QUERY DEFINITIONS

A.1 INTRODUCTION

This appendix defines the SCPI common commands and queries used with the BIT 4886 Interface Card installed in each BOP 2X-DG channel. Common commands and queries are preceded by an asterisk (*) and are defined and explained in paragraphs A.2 through A.17, arranged in alphabetical order. Table A-1 provides a quick reference of all SCPI common commands and queries used in the Interface Card.

TABLE A-1. SCPI COMMAND/QUERY INDEX

COMMAND	PAR.	COMMAND	PAR.
*CLS	A.2	*RST	A.10
*ESE, ?	A.3, A.4	*SAV	A.11
*ESR?	A.5	*SRE, ?	A.12, A.13
*IDN?	A.6	*STB?	A.14
*OPC	A.7	*TRG	A.15
*OPT?	A.8	*TST?	A.16
*RCL	A.9	*WAI	A.17

A.2 *CLS — CLEAR STATUS COMMAND *CLS

Syntax: *CLS

Description: **Clears status data.** Clears the error queue of the instrument. Forces power supply to “operation complete idle” and “operation complete query” state. Clears all Event Registers summarized in Status Byte Register without affecting the corresponding Enable Registers: Standard Event Status Register (ESR), Operation Status Event Register, Questionable Status Event Register, and Status Byte Register (STB). Related commands: *OPC. (See example, Figure A-1.)

A.3 *ESE — STANDARD EVENT STATUS ENABLE COMMAND *ESE

Syntax**:** *ESE <integer> where <integer> = positive whole number: 0 to 255 per Table A-2.
Default Value: 0

Description: **This command programs the standard Event Status Enable register bits.** The contents function as a mask to determine which events of the Event Status Register (ESR) are allowed to set the ESB (Event Summary Bit) of the Status Byte Register. Enables the Standard events to be summarized in the Status Byte register (1 = set = enable function, 0 = reset = disable function). All of the enabled events of the standard Event Status Enable register are logically ORed to cause ESB (bit 5) of the Status Byte Register to be set (1 = set = enable, 0 = reset = disable). (See example, Figure A-1.)

TABLE A-2. STANDARD EVENT STATUS ENABLE REGISTER AND STANDARD EVENT STATUS REGISTER BITS

CONDITION	NU	NU	CME	EXE	DDE	QUE	NU	OPC
BIT	7	6	5	4	3	2	1	0
VALUE	128	64	32	16	8	4	2	1

NU (Not Used)
CME Command Error
EXE Execution Error
DDE Device Dependent Error
QUE Query Error
OPC Operation Complete

A.4 *ESE? — STANDARD EVENT STATUS ENABLE QUERY *ESE?

Syntax**:** *ESE? Return value: Integer> value per Table A-2.

Description: **Returns the mask stored in the Standard Event Status Enable Register.** Contents of Standard Event Status Enable register (*ESE) determine which bits of Standard Event Status register (*ESR) are enabled, allowing them to be summarized in the Status Byte register (*STB). All of the enabled events of the Standard Event Status Enable Register are logically ORed to cause ESB (bit 5) of the Status Byte Register to be set (1 = set = enable function, 0 = reset = disable function). (See example, Figure A-1.)

A.5 *ESR? — EVENT STATUS REGISTER QUERY *ESR?

Syntax**:** *ESR?
Return value: <integer> (Value = contents of Event Status register as defined in Table A-2.)

Description: **Causes the power supply to return the contents of the Standard Event Status register. After it has been read, the register is cleared.** The Standard Event Status register bit configuration is defined in Table A-2 (1 = set, 0 = reset). The error bits listed in Table A-2 are also related to error codes produced during parsing of messages and to errors in the power supply (see PAR. B.80)

- Any 1xx type error sets the Command error bit (5) see.
- Any 2xx type error sets the Execution error bit (4).
- Any 3xx type error sets the Device error bit (3). The Device error bit will be set when Current Error or Voltage Error is detected and the corresponding Status Questionable bit is set (see PAR. B.28).
- Any 4xx type error sets the Query error bit (2).

Related Commands: *CLS, *ESE, *OPC. (See example, Figure A-1.)

A.6 *IDN? — IDENTIFICATION QUERY *IDN?

Syntax**:** *IDN?
Return value: Character string

Description: **Identifies the instrument.** This query requests identification. The power supply returns a string which contains the manufacturer name, the model, the serial number and the firmware level. For example, for BOP 2X20-10DG the query returns KEPCO,BIT 4886 20-10,E1233456**A** 11/10/98,1.81-1.81 for channel 1 and KEPCO,BIT 4886 20-10,E1233456**B** 11/10/98,1.81-1.81 for channel 2. The character string contains the following fields: <Manufacturer>, <Model>, <Voltage>, <Current>, <Serial Number>, <Main revision>-<Flash revision> where: <Manufacturer> = KEPCO, <Model> = BIT 4886 (interface card installed in BOP 2X-DG) VV-AA (where VV = nominal output voltage of each channel, AA = nominal output current of each channe), <Serial Number> = SSSSSSX (where X = A for channel 1, B for channel 2) <date> = (see CAL:SAVE DATE, PAR. B.11) <Main revision=n.m, e.g, 1.0) ><Firmware revision>=n.m (n.m revision, e.g, 1.0) (Also see example, Figure A-1.)

A.7 *OPC — OPERATION COMPLETE COMMAND

Syntax**:** *OPC

Description: **Causes power supply to set status bit 0 (Operation Complete) when pending operations are complete** This command sets Standard Event Status Register bit 0 (see Table A-2) to “1” when all previous commands have been executed and changes in output level have been completed. This command does not prevent processing of subsequent commands, but bit 0 will not be set until all pending operations are completed. (1 = set = enable function, 0 = reset = disable function). (See example, Figure A-1.) As an example, the controller sends command(s), then sends *OPC. If controller then sends *ESR?, the power supply responds with either a “0” (if the power supply is busy executing the programmed commands), or a “1” (if the previously programmed commands are complete). (See example, Figure A-1.)

*CLS	Power supply clears status data.
*ESE 60	Power supply enables bits 5, 4, 3 and 2, allowing command error, execution error, device dependent error and query error to set the Event Status Summary bit when an STB command is executed.
*ESE?	Returns 60, (value of the mask) verifying that bits 5, 4, 3 and 2 are enabled.
*ES	Unknown command will set command error (Bit 5).
*ESR?	Returns 32 (bit 5 set), indicating Command Error has occurred since the last time the register was read.
*IDN?	Power supply returns: KEPCO,BIT 4886 20-10,E1233456A 11/10/98, 1.81-1.81 for channel 1 of BOP 2X20-10DG
*OPC	Allows status bit 0 to be set when pending operations complete
VOLT 21;CURR 3	Sets output voltage to 21V, output current to 3A
*ESR?	Returns 129 (128 + 1, power on, bit 7 = 1, operation complete, bit 1 = 1)
*ESR?	Returns 0 (event status register cleared by prior *ESR?)
VOLT 15;CURR 5;*ESR?	Sets output voltage to 15V, output current to 5A, puts “1” on output bus when command operations are complete.
*RST	Power supply reset to power on default state.
*SRE 40	When ESB or QUES bits are set (Table A-3), the Request for Service bit will be set.
*SRE?	Returns the value of the mask (40).
*STB?	For example, the Power supply responds with 96 (64 + 32) if MSS and the Event Status Byte (Table A-3) summary bit have been set. The power supply returns 00 if no bits have been set.
VOLT 25	Power supply voltage commanded to 25V.
VOLT:TRIG 12	Programs power supply voltage to 12V when *TRG received.
INIT	Trigger event is initialized.
*TRG	Power supply reverts to commanded output voltage of 12V. ** LOAD DISCONNECTED
*TST?	Power supply executes self test and responds with 0 if test completed successfully, with 1 if test failed.

FIGURE A-1. IEEE 488 COMMANDS

A.8 *OPT? — OPTIONS QUERY *OPT?

Syntax**:** *OPT?
Returns string determined by power supply model.

Description: **Causes the power supply to return an ASCII string which defines the functionality of the power supply.** The functionality is defined as follows:

STRING DATA	MEANING
CAL	Support for CALibrate is present.
RL1	Commands sent over GPIB cause unit to enter remote mode (except for MEASure).
MEM	Indicates the number of memory steps supported.
LST	Indicates the number of LIST steps supported.

A.9 *RCL — RECALL COMMAND *RCL

Syntax**:** *RCL <integer> (1 to 99)

Description: **Restores power supply to previously defined levels of output voltage, output current and triggers.** This command selects one of the 99 power supply memory locations, each of which stores values for output current, and output voltage and trigger levels. Executing a *RCL recalls the previously defined trigger settings from memory and places them in the trigger control section of the BIT 4886. Executing a second *RCL will cause the values in the trigger control section to be moved to the bit 4886 output channel, allowing the power supply to operate with the recalled trigger information. The following parameters are affected by *RCL: VOLT:TRIG, CURR:TRIG, and FUNC:MODE:TRIG.

A.10 *RST — RESET COMMAND *RST

Syntax**:** *RST

Description: **Resets power supply to the power on default state.** The power supply output set to off (see DIAG:OUTP, Table 3-2, to establish the “off” state parameters) and the power supply is programmed to the power on values of the following parameters: CURR[:LEV][:IMM] = 0, VOLT[:LEV][:IMM] = 0, MODE = VOLT. If the power supply is in either an overvoltage or overcurrent state, this condition is reset by *RST. After sending *RST it is necessary to send OUTPUT ON for programmed values to appear at the output. The *RST command always returns the mode to Voltage and the range to automatic. (See example, Figure A-1.)

A.11 *SAV — SAVE COMMAND *SAV

Syntax**:** *SAV <integer> (1 to 99)

Description: **Saves the present state of output voltage, output current and trigger values, to the specified memory location.** This command stores the present state of the power supply to one of 99 memory locations in Flash Memory (see PAR. 3.9.4). The following parameters are stored by *SAV: VOLT:TRIG, CURR:TRIG, and FUNC:MODE:TRIG. The stored values can be restored by the *RCL command.

A.12 *SRE — SERVICE REQUEST ENABLE COMMAND *SRE

Syntax**:** *SRE<integer> where <integer> = value from 0 - 255 per Table A-3, except bit 6 cannot be programmed.

Description: **Sets the condition of the Service Request Enable register.** The Service Request Enable register determines which events of the Status Byte Register are summed into the MSS (Master Status Summary) and RQS (Request for Service) bits. RQS is the service request bit that is cleared by a serial poll, while MSS is not cleared when read. A “1” (1 = set = enable, 0 = reset = disable) in any Service Request Enable register bit position enables the corresponding Status Byte bit to set the RQS and MSS bits. All the enabled Service Request Enable register bits then are logically ORed to cause Bit 6 of the Status Byte Register (MSS/RQS) to be set. Related Commands: *SRE?, *STB?. (See example, Figure A-1.)

TABLE A-3. SERVICE REQUEST ENABLE AND STATUS BYTE REGISTER BITS

CONDITION	OPER	MSS RQS	ESB	MAV	QUES	ERR QUE	NU	NU
BIT	7	6	5	4	3	2	1	0
VALUE	128	64	32	16	8	4	2	1

OPER	Operation Status Summary
MSS	Master Status Summary
RQS	Request for Service
ESB	Event Status Byte summary
MAV	Message available
QUES	QUEStionable Status Summary
ERR QUE	1 or more errors occurred (see PAR. B.80)
NU	(Not Used)

A.13 *SRE? — SERVICE REQUEST ENABLE QUERY *SRE?

Syntax**:** *SRE? Response: <integer> = value from 0 - 255 per Table A-3.

Description: **Reads the Service Enable Register.** Used to determine which events of the Status Byte Register are programmed to cause the power supply to generate a service request (1 = set = function enabled, 0 = reset = function disabled). Related Commands: *SRE, *STB? (See example, Figure A-1.)

A.14 *STB? — STATUS BYTE REGISTER QUERY *STB?

Syntax**:** *STB? Response: <integer> value from 0 to 255 per Table A-3.

Description: **Reads Status Byte Register without clearing it.** This Query reads the Status Byte Register (bit 6 = MSS) without clearing it (1 = set = function enabled, 0 = reset = function disabled). The register is cleared only when subsequent action clears all set bits. MSS is set when the power supply has one ore more reasons for requesting service. (A serial poll also reads the Status Byte Register, except that bit 6 = RQS, not MSS; and RQS will be reset.) Related Commands: *SRE, *SRE?. (See example, Figure A-1.)

A.15 *TRG — TRIGGER COMMAND *TRG

Syntax**:** *TRG

Description: **Triggers the power supply to be commanded to preprogrammed values of output current and voltage.** When the trigger is armed (checked by examining WTG bit in Status Operational Condition register), *TRG generates a trigger signal. The trigger will change the output of the power supply to the output voltage and current levels specified by VOLT:TRIG and CURR:TRIG commands and clear the WTG bit in the Status Operation Condition register. If INIT:CONT has been issued, the trigger subsystem is immediately rearmed for subsequent triggers, and the WTG bit is again set to 1. *TRG or GET are both addressed commands (only devices selected as listeners will execute the command). Related Commands: ABOR, INIT, TRIG, CURR:TRIG, VOLT:TRIG. (See example, Figure A-1.)

A.16 *TST? — SELF TEST QUERY *TST?

Syntax**:** *TST? Returned value: 0 or non-zero (0 = pass test, non-zero = fail test)

Description: **Power Supply test.** This query causes the power supply to do a self test and provide the controller with pass/fail results. A 0 is returned if the unit passes the test. If the unit fails, a number is returned to indicate the cause of the error. The test executes each of the subtests even when any one fails. If any test fails an error code bit is set which is returned to the user. The error codes returned are listed in Table 3-1.

A.17 *WAI — WAIT-TO-CONTINUE COMMAND *WAI

Syntax**:** *WAI Response:

Description: **Causes the power supply to wait until all previously issued commands and queries are complete before executing subsequent commands or queries.** This command can be used to guarantee sequential execution of commands and queries. When all pending operations are complete (all previous commands have been executed, changes in output level have been completed), the WAI command is completed and execution of subsequent commands can continue.

APPENDIX B - SCPI COMMAND/QUERY DEFINITIONS

B.1 INTRODUCTION

This appendix defines the SCPI subsystem commands and queries used with the BIT 4886 Interface Card found in each channel of the BOP 2X-DG. Commands and queries are defined in PAR. B.4 through B.92, arranged in groups as they appear in the tree diagram, Figure 3-25. Table B-1 provides a quick reference of all SCPI subsystem commands and queries used.

TABLE B-1. SCPI SUBSYSTEM COMMAND/QUERY INDEX

COMMAND	PAR.	COMMAND	PAR.
CAL:CPR	B.3	[SOUR:]LIST:VOLT, ?	B.45, B.46
CAL:STAT, ?	B.4, B.5	[SOUR:]LIST:VOLT:POIN?	B.47
CAL:CURR	B.6	[SOUR]:CURR, ?	B.48, B.49
CAL:DATA	B.7	[SOUR]:CURR:MODE, ?	B.50, B.51
CAL:DPOT	B.8	[SOUR]:CURR:RANG, ?	B.52, B.53
CAL:LCURR	B.9	[SOUR]:CURR:RANG:AUTO	B.54
CAL:LVOLT	B.10	[SOUR]:CURR:TRIG?	B.55, B.56
CAL:SAVE	B.11	[SOUR]:VOLT	B.57, B.58
CAL:VOLT	B.12	[SOUR]:VOLT:MODE, ?	B.59, B.60
CAL:VPR	B.13	[SOUR]:VOLT:RANG, ?	B.61, B.62
CAL:ZERO	B.14	[SOUR]:VOLT:RANG:AUTO	B.63
INIT[:IMM]	B.15	[SOUR]:VOLT:TRIG	B.64, B.65
INIT:CONT, ?	B.16, B.17	STAT:OPER:COND?	B.66
MEAS:CURR?	B.18	STAT:OPER:ENAB, ?	B.67, B.68
MEAS:VOLT?	B.19	STAT:OPER[:EVEN]?	B.69
OUTP[:STAT], ?	B.20, B.21	STAT:PRES	B.70
[SOUR:]FUNC:MODE, ?	B.22, B.23	STAT:QUES[:EVEN]?	B.71
[SOUR:]FUNC:MODE:TRIG, ?	B.24, B.25	STAT:QUES:COND?	B.72
[SOUR:]LIST:CLE	B.26	STAT:QUES:ENAB, ?	B.73, B.74
[SOUR:]LIST:COUN, ?	B.27, B.28	SYST:BEEP	B.75
[SOUR:]LIST:COUN:SKIP, ?	B.29, B.30	SYST:ERR?	B.80
[SOUR:]LIST:CURR, ?	B.31, B.32	SYST:ERR:CODE?	B.81, B.82
[SOUR:]LIST:POIN?	B.33	SYST:PASS CEN, CDIS	B.83, B.84
[SOUR:]LIST:DIR, ?	B.34, B.35	SYST:PASS:STAT	B.86
[SOUR:]LIST:DWEL, ?	B.36, B.37	SYST:REM, ?	B.87, B.88
[SOUR:]LIST:DWEL:POIN?	B.38	SYST:SEC	B.89
[SOUR:]LIST:GEN, ?	B.39, B.40	SYST:SET, ?	B.90, B.91
[SOUR:]LIST:QUER, ?	B.41, B.42	SYST:VERS?	B.92
[SOUR:]LIST:SEQ, ?	B.43, B.44		
NOTE: Commands listed above that are followed by ", ?" have a related query.			

B.2 NUMERICAL VALUES

The SCPI data parser on the BIT 4886 supports a maximum of 8 digits after the decimal point and a maximum integer of 4 x 10^8. Any values greater than these are not processed by the device and no error is generated. The largest string that can be received or transmitted by the BIT 4886 is 253 characters.

All numerical data is returned in scientific notation, digits with decimal point and Exponent, e.g., 2.71E1 for 27.1 after calibration constants have been applied. Thus. for example, VOLT 14;VOLT? may return 1.39997E1 which indicates that the unit has been calibrated to provide 13.9997V for a programmed value of 14V, within the calculation accuracy of the BIT 4886. Error "-120" results from syntactical errors, e.g., the exponent exceeds 8, a letter is identified, etc. Error "-222" is produced if the value exceeds the range of acceptable values for the parameter.

B.3 CALibrate:CPRotect COMMAND — CAL:CPR

Syntax: Short Form: CAL:CPR {MIN | MAX}
Long Form: CALibrate:CPRotect {MIN | MAX}

Description: **Selects Current Protection limit calibration, only effective with power supply in Calibrate status.** CAL:CPR MAX selects maximum positive current protection limit calibration. CAL:CPR MIN selects maximum negative current protection limit calibration. **Related Commands:** CAL:STAT, CAL:ZERO, CAL:DATA, CAL:SAVE.

B.4 CALibrate:STATus COMMAND — CAL:STAT

Syntax: Short Form: CAL:STAT <boolean>
Long Form: CALibrate:STATus <boolean>
where boolean = 0 or OFF, 1 or ON

Description: **Sets the power supply to Calibrate status.** <boolean> 1 or ON causes power supply to enter Calibrate status. <boolean> 0 or OFF causes power supply to exit Calibrate status. If the wrong password was not enabled, error message -224 is posted to the queue. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

B.5 CALibrate[:STATus]? QUERY — CAL[:STAT]?

Syntax: Short Form: CAL:STAT? Long Form: CALibrate:STATus?
Return Value: <DATA>
where DATA = ZERO-ZERO, VOLT-ZERO, VOLT-MAX, VOLT-MIN, CURR-ZERO, CURR-MAX, CURR-MIN, or OFF.

Description: **Identifies whether the power supply is in Calibrate status and indicates which calibration step is active.** OFF indicates power supply is not in Calibrate status. ZERO-ZERO indicates power supply is at 0V, 0A, waiting for connection of load resistor. VOLT-ZERO indicates zero offset voltage adjustment is active. VOLT-MAX indicates maximum voltage adjustment is active. VOLT-MIN indicates minimum (negative) voltage adjustment is active. CURR-ZERO indicates zero offset current adjustment is active. CURR-MAX indicates maximum current adjustment is active. CURR-MIN indicates minimum (negative) current adjustment is active. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

SYST:PASS:CEN 1234	If password set to 1234, enables commands requiring password.
CAL:STAT 1	Power supply enters Calibrate status.
CAL:STAT?	Returns ZERO-ZERO indicating power supply in Calibrate status.
CAL:VOLT ZERO	Voltage Zero Calibration selected.
CAL:VOLT MAX	Voltage positive Calibration selected.
CAL:DPOT 1	Output voltage increased by 10 LSB.
CAL:VOLT MIN	Voltage negative Calibration selected.
CAL:DATA -2	Output voltage decreased by 2 LSBs.
CAL:ZERO	Output voltage and current set to zero.
---	* User connects shunt and connects DVM to output.
CAL:CURR ZERO	Current Zero Calibration selected.
CAL:DPOT -1	Output current decreased by 10 LSB.
CAL:CURR MIN	Current negative Calibration selected.
CAL:DPOT 1	Output current increased by 10 LSB.
CAL:CURR MAX	Current positive Calibration selected.
CAL:DATA -1	Output current decreased by 1 LSB.
CAL:SAVE	Calibration values saved.
CAL:STAT 0	Power supply exits Calibrate mode.
CAL:STAT?	Returns 0 indicating power supply not in Calibrate status
---	* User disconnects shunt and DVM.

FIGURE B-1. USING CALIBRATION COMMANDS AND QUERIES

B.6 CALibrate:CURRent COMMAND — CAL:CURR

Syntax: Short Form: CAL:CURR {MIN | MAX | ZERO}
Long Form: CALibrate:CURR {MIN | MAX | ZERO}

Description: **Selects Current calibration, only effective with power supply in Calibrate status.** CAL:CURR ZERO selects Current Zero Calibration. CAL:CURR MIN selects Current Full Scale Negative Calibration. CAL:CURR MAX selects Current Full Scale Positive Calibration. Normally Current Zero is done first, then Current Full Scale Positive Calibration. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

B.7 CALibrate:DATA COMMAND — CAL:DATA

Syntax: Short Form: CAL:DATA {N} Long Form: CALibrate:DATA {N}

Description: **Provides fine adjustment of output during calibration only; Increases or decreases output by N LSB's.** CAL:CURR -N decreases output by N LSBs. CAL:CURR N increases output by N LSBs. This command is used during calibration to adjust the output for zero calibration as well as full scale calibration. This command is only effective if Calibration status is active. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:LVOLT, CAL:LCURR, CAL:CURR, CAL:DPOT, CAL:ZERO, CAL:SAVE. (See example, Figure B-1.)

B.8 CALibrate:DPOT COMMAND — CAL:DPOT

Syntax: Short Form: CAL:DPOT {N} Long Form: CALibrate:DPOT {N}

Description: **Provides coarse adjustment of the output during calibration to within 5% of maximum and 2% of nominal; increases or decreases output by 50 LSB increments.** Acceptable values of N are within ±32, corresponding to ±1600 (50 x 32) LSB's. CAL:DPOT -N decreases output voltage by N LSBs. CAL:VOLT N increases output voltage by N LSBs. This command is used during calibration to adjust the output for Zero calibration as well as Full Scale Positive calibration. Output voltage is monitored on a calibrated digital multimeter and increased or decreased as required using this command. Output current is monitored on a calibrated digital multimeter (reading voltage at the sense terminals of the shunt resistor) and increased or decreased as required using this command. This command is only effective if Calibration status and either Voltage Zero, Voltage Maximum. Current Zero or Current Maximum calibration are active. CAL:STAT, CAL:VOLT, CAL:LVOL, CAL:VOLT, CAL:CURR, CAL:DATA, CAL:ZERO, CAL:SAVE. (See example, Figure B-1.)

B.9 CALibrate:LCURR COMMAND — CAL:LCURR

Syntax: Short Form: CAL:LCURR {MIN | MAX | ZERO}
Long Form: CALibrate:LCURR {MIN | MAX | ZERO}

Description: **Selects low current range (1/4 scale) Current calibration, only effective with power supply in Calibrate status.** CAL:LCURR ZERO selects Low Current Range Zero Calibration. CAL:LCURR MIN selects Low Current Range Full Scale Negative Calibration. CAL:LCURR MAX selects Low Current Range Full Scale Positive Calibration. Normally Low Current Range Zero is done first, then Low Current Range Full Scale Positive Calibration. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

B.10 CALibrate:LVOLT COMMAND — CAL:LVOLT

Syntax: Short Form: CAL:LVOLT {MIN | MAX | ZERO}
Long Form: CALibrate:LVOLT {MIN | MAX | ZERO}

Description: **Selects low voltage range (1/4 scale) Voltage calibration, only effective with power supply in Calibrate status.** CAL:LVOLT ZERO selects Low Voltage Range Zero Calibration. CAL:LVOLT MIN selects Low Voltage Range Full Scale Negative Calibration. CAL:LVOLT MAX selects Low Voltage Range Full Scale Positive Calibration. Normally Low Voltage Range Zero is done first, then Low Voltage Range Full Scale Positive Calibration. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

B.11 CALibrate:SAVE COMMAND — CAL:SAVE

Syntax: Short Form: CAL:SAVE Long Form: CALibrate:SAVE
Short Form: CAL:SAVE DATE <string> Long Form: CALibrate:SAVE DATE <string>
where DATE is optional allowing <string> of up to 12 contiguous characters identifying calibration date to be saved. Active DATE <string > is returned by *IDN? query.

Description: **Saves computed calibration values in non-volatile memory.** This command saves only the calibration values computed after entering Calibration status. If, for example, only Voltage calibration was performed, these are the only values saved. These values are stored in Flash Memory (see PAR. 3.9.4). Previous values are not lost, and can be restored (see PAR. 3.9.4.1). This command should be the last command before exiting Calibrate status. CAL:STAT, CAL:DATA, CAL:DPOT, CAL:VOLT, CAL:CURR, CAL:ZERO. (See example, Figure B-1.)

B.12 CALibrate:VOLTage COMMAND — CAL:VOLT

Syntax: Short Form: CAL:VOLT {MIN | MAX | ZERO}
Long Form: CALibrate:VOLT {MIN | MAX | ZERO}

Description: **Selects Voltage calibration, only effective with power supply in Calibrate status.**
CAL:VOLT ZERO selects Voltage Zero Calibration. CAL:VOLT MIN selects Voltage Full Scale Negative Calibration. CAL:VOLT MAX selects Voltage Full Scale Positive Calibration. Normally Voltage Zero is done first, then Voltage Full Scale Positive Calibration. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:ZERO, CAL:DATA, CAL:DPOT, CAL:SAVE. (See example, Figure B-1.)

B.13 CALibrate:VPRotect COMMAND — CAL:VPR

Syntax: Short Form: CAL:VPR {MIN | MAX}
Long Form: CALibrate:VPRotect {MIN | MAX}

Description: **Selects Voltage Protection limit calibration, only effective with power supply in Calibrate status.** CAL:VPR MAX selects maximum positive voltage protection limit calibration. CAL:VPR MIN selects maximum negative voltage protection limit calibration. **Related Commands:** CAL:STAT, CAL:ZERO, CAL:DATA, CAL:SAVE.

B.14 CALibrate:ZERO COMMAND — CAL:ZERO

Syntax: Short Form: CAL:ZERO Long Form: CALibrate:ZERO

Description: **Sets output to zero while calibration equipment connections are changed.** This command is used when changing from Voltage to Current calibration or vice-versa. CAL:ZERO sets output voltage and current to zero. The user then connects or disconnects the shunt resistor or the digital voltmeter from the output terminals of the power supply as required for the subsequent Voltage or Current calibration. The next command should select the new parameter to be calibrated. **Related Commands:** CAL:STAT, CAL:VOLT, CAL:CURR, CAL:SAVE. (See example, Figure B-1.)

B.15 INITiate[:IMMediate] COMMAND — INIT[:IMM]

Syntax: Short Form: INIT:[IMM] Long Form: INITiate[:IMMediate]

Description: **Enables a single trigger.** This command enables a single trigger. A *TRG command completes the sequence. Upon receipt of the *TRG command, the power supply will return to the commanded values of voltage and current established by the VOLT:TRIG and CURR:TRIG commands. After a *TRG command has been received, subsequent *TRG commands have no effect unless preceded by INIT or INIT:CONT ON. Related Commands: *TRG, TRIG. (See example, Figure B-4.)

B.16 INITiate:CONTinuous COMMAND — INIT:CONT

Syntax: Short Form: INIT:CONT {ON | OFF} or {1 | 0} (1 = on, 0 = off)
Long Form: INITiate:CONTinuous {ON | OFF} or {1 | 0} (1 = on, 0 = off)

Description: **INIT:CONT ON enables continuous triggers.; INIT:CONT OFF disables continuous triggers.** If INIT:CONT is OFF, then INIT[:IMM] arms the trigger system for a single trigger. If INIT:CONT is ON, then the trigger system is continuously armed and INIT[:IMM] is redundant. Executing *RST command sets INIT:CONT to ON. (See example, Figure B-4.)

B.17 INITiate:CONTinuous QUERY — INIT:CONT?

Syntax: Short Form: INIT:CONT? Long Form: INITiate:CONTinuous?
Return Value: 1 or 0

Description: **Determines whether continuous triggers are enabled or disabled.** Power supply returns value of INIT:CONT flag: “1” = continuous triggers are enabled (INIT:CONT ON); “0” = continuous triggers disabled (INIT:CONT OFF). (See example, Figure B-4.)

B.18 MEASure[:SCALar]:CURRent[:DC]? QUERY — MEAS:CURR?

Syntax: Short Form: MEAS[:SCAL]:CURR[:DC]?
Long Form: MEASure[:SCALar]:CURRent[:DC]?
Return Value: <num_value> (digits with decimal point and Exponent)

Description: **Measures actual current.** This query returns the actual value of output current (measured at the output terminals) as determined by the programmed value of voltage and current and load conditions. (See example, Figure B-4.)

B.19 MEASure[:VOLTage][:SCALar][:DC]? QUERY — MEAS:VOLT?

Syntax: Short Form: MEAS[:SCAL]:VOLT[:DC]?
Long Form: MEASure[][:SCALar]:VOLTage[:DC]?
Return Value: <num_value> (digits with decimal point and Exponent)

Description: **Measures actual voltage.** This query returns the actual value of output voltage (measured at the output terminals) as determined by the programmed value of voltage and current and load conditions. (See example, Figure B-4.)

B.20 OUTPut[:STATe] COMMAND — OUTP

Syntax: Short Form: OUTP[:STAT] <boolean> Long Form: OUTPut[:STATe] <boolean>
<boolean>=(0 or OFF, 1 or ON)

Description: **Enables or disables the power supply output.** Upon power up the power supply is in Local mode: the output is always on and the front panel controls affect the unit's operation. Upon entering Remote mode, the default state is OUTP OFF; OUTP ON must be executed to enable the output. When OUTP OFF is executed, the programmed values of voltage and current are saved, then voltage and current are programmed to 0. When OUTP ON is executed, the power supply output is restored to the previously saved programmed values. The saved values of voltage and current can be viewed by VOLT? and CURR? queries. Related Commands: OUTP?.

B.21 OUTPut[:STATe] QUERY — OUTP?

Syntax: Short Form: OUTP[:STAT]? Long Form: OUTPut[:STATe]?
Return Value: <int_value> (0 or 1)

Description: **Indicates whether power supply output is enabled or disabled.** Returns 0 if output disabled, returns 1 if output enabled. Related Commands: OUTP.

B.22 [SOURce:]FUNCtion:MODE COMMAND — FUNC:MODE

Syntax: Short Form: FUNC:MODE {VOLT | CURR} Long Form: [SOURce:]FUNCtion:MODE {VOLT | CURR}

Description: **Establishes the operating mode of the power supply.** VOLT = Voltage mode. CURR = Current mode. FUNC:MODE VOLT commands power supply to Voltage mode, FUNC:MODE CURR commands power supply to Current mode. Commanded mode establishes parameters (voltage or current) monitored for error conditions. Actual mode depends upon load conditions. When commanded to Voltage mode, if load conditions cause the power supply to try to exceed the current limit, the unit will flag an error condition. When commanded to Current mode, if load conditions cause the power supply to try to exceed the voltage limit, the unit will flag an error condition. If VOLT:MODE or CURR:MODE is set to TRANS, FUNC:MODE command automatically changes the mode to FIXed. (See example, Figure B-4.)

B.23 [SOURce:]FUNCtion:MODE? QUERY — FUNC:MODE?

Syntax: Short Form: FUNC:MODE? Long Form: [SOURce:]FUNCtion:MODE?
Return value: 0 (voltage mode) or 1 (current mode)

Description: **Returns the operating mode of the power supply.** 0 = Voltage mode. 1 = Current mode.

B.24 [SOURce:]FUNCtion:MODE:TRIGger — FUNC:MODE:TRIG

Syntax: Short Form: FUNC:MODE:TRIG {VOLT | CURR}
Long Form: [SOURce:]FUNCtion:MODE:TRIGger {VOLT | CURR}

Description: **Establishes the operating mode of the power supply when a TRIGger command is sent.** VOLT = Voltage mode. CURR = Current mode. This command establishes the operating mode for triggers which is to be saved or recalled using *SAV or *RCL command. FUNC:MODE:TRIG VOLT commands power supply to Voltage mode when a TRIGger command is sent, FUNC:MODE:TRIG CURR commands power supply to Current mode when a TRIGger command is sent. If VOLT:MODE or CURR:MODE is set to TRANS, FUNC:MODE:TRIG command automatically changes the mode to FIXed. (See example, Figure B-5.)

B.25 [SOURce:]FUNCtion:MODE:TRIGger? QUERY — FUNC:MODE:TRIG?

Syntax: Short Form: FUNC:MODE:TRIG? Long Form: [SOURce:]FUNCtion:MODE:TRIGger?
Return value: 0 = voltage mode or 1 = current mode

Description: **Returns the operating mode which the power supply will enter when a trigger command is sent.** VOLT = Voltage mode. CURR = Current mode. (See example, Figure B-5.)

B.26 [SOURce:]LIST:CLEar COMMAND — LIST:CLE

Syntax: Short Form: LIST:CLE Long Form: LIST:CLEar>

Description: **Clears all list entries by setting all pointers to 0.** Also sets LIST:DIR to UP, LIST:GEN to DSEQ and LIST:COUNT to 1. Related Commands: All LIST commands (See example, Figures B-2 and B-3.)

B.27 [SOURce:]LIST:COUNt COMMAND — LIST:COUN

Syntax: Short Form: LIST:COUN<int_value 0 to 255> Long Form: LIST:COUNt> <int_value 0 to 255>

Description: **Establishes how many times the list is executed.** Allows user to establish how many times the list (established by LIST:GEN DSEQ or LIST:GEN SEQ and LIST:SEQ) is executed. The order (from beginning to end or from end to beginning) is determined by LIST:DIR. For LIST:COUN 0, the unit will execute the sequence indefinitely until either a VOLT:MODE FIXED, or PROG:STOP command is received. Commands: LIST:GEN, LIST:SEQ, LIST:DIR. (See example, Figures B-2 and B-3.)

B.28 [SOURce:]LIST:COUNt? QUERY — LIST:COUN?

Syntax: Short Form: LIST:COUNT? Long Form: LIST:COUNt?
Return Value: <int_value>

Description: **Identifies how many times the list will be executed.** Returns value set by LIST:COUN command. (See example, Figure B-3.)

B.29 [SOURce:]LIST:COUNt:SKIP COMMAND LIST:COUN:SKIP

Syntax: Short Form: LIST:COUN:SKIP nn Long Form: LIST:COUNt:SKIP nn
nn = <int_value 0 to 255>

Description: **Allows beginning steps of list-generated waveform to be run once, then ignored.**

When a list is to be repeated using LIST:COUNT, this command allows the user to skip the first nn steps once the full set has been executed. After the first iteration (which executes all steps), the first nn steps are skipped. The LIST:COUN:SKIP command allows the user to precondition a list-generated waveform by setting unique conditions at the beginning that are not repeated for the rest of the repetitions. LIST:CLEar sets nn to 0. Only works in LIST:DIR UP mode; if LIST:DIR DOWN is issued, this command has no effect. Related Commands: LIST:COUN, LIST:COUN:SKIP?, LIST:SEQ, LIST:DIR, LIST:CLE. (See example, Figures B-2 and B-3.)

B.30 [SOURce:]LIST:COUNt:SKIP? QUERY LIST:COUN:SKIP?

Syntax: Short Form: LIST:COUN:SKIP? Long Form: LIST:COUNt:SKIP?
Return Value: <int_value>

Description: **Identifies how many steps will be skipped the first time the list is executed.** Returns value set by LIST:COUN:SKIP command. (See examples, Figure B-3.)

B.31 [SOURce:]LIST:CURRent COMMAND LIST:CURR

Syntax: Short Form: LIST:CURR <exp_value>, <exp_value>, . . . (to max of 1002 data points)
Long Form: LIST:CURRent <exp_value>, <exp_value>, . . . (to max of 1002 data points)
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Adds the current value (in Amps) to list.** This command sequentially adds LIST:CURRent values to the main channel List Data Table locations illustrated in Table B-2. Starting location is indicated by LIST:CURR:POIN? These locations correspond to the default sequence (LIST:GEN DESQ). The maximum number of entries is 1002. Since the input buffer of the BIT 4886 has a limit of 253 characters, multiple commands are necessary to complete the full 1002 entries of the list. If LIST:VOLT has any entries, an error message: -221,"Settings conflict" is posted in the error queue. Related Commands: LIST:CURR:POIN?. (See example, Figure B-3.)

TABLE B-2. LIST DATA TABLE

Location (DSEQ)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	. . . . .	1001
Main Channel (commanded mode) LIST:CURRent or LIST:VOLTage																			. . . . .	
LIST:DWELl																			. . . . .	
NOTE: FUNC:MODE determines whether main channel is for voltage or current. FUNC:MODE CURR must precede LIST:CURR commands, FUNC:MODE VOLT must precede LIST:VOLT commands																				

B.32 [SOURce:]LIST:CURRent? QUERY LIST:CURR?

Syntax: Short Form: LIST:CURR? Long Form: LIST:CURRent?
Return Value: <value1>, <value2>, . . . to <value16>

Description: **Identifies the parameters (main channel) entered for the list.** Starting at location established by LIST:QUERy, returns comma-separated list of up to 16 values indicating the main channel parameters entered, i.e., the contents of main channel locations of Table B-2. Related Commands: LIST: CURR, LIST:QUERy. If LIST:VOLT has any entries, an error message: -221,"Settings conflict" is posted in the error queue. (See example, Figure B-3.)

B.33 [SOURce:]LIST:CURRent:POINts? QUERY LIST:CURR:POIN?

Syntax: Short Form: LIST:CURR:POIN? Long Form: LIST:CURRent:POINts?
Return Value: <value> (0 to 1001)

Description: **Identifies the total number of points in a list and the next location to be filled by LIST:CURR command.** The LIST:CURR pointer is initially at 0 via LIST:CLE. For each data point entered by a LIST:CURR command the list pointer is incremented. If LIST:CURR:POIN? returns 5, the LIST:CURR pointer is at 5, indicating there are 6 data points comprising the list. If LIST:VOLT has any entries, an error message: -221,"Settings conflict" is posted in the error queue. Related Commands: LIST:CURR. (See example, Figure B-3.)

B.34 [SOURce:]LIST:DIRection COMMAND LIST:DIR

Syntax: Short Form: LIST:DIR (UP|DOWN) Long Form: LIST:DIRection (UP|DOWN)

Description: **Allows the list to be executed from beginning to end (UP) or from end to the beginning (DOWN).** *RST or LIST:CLEar sets the list to the UP direction (beginning to end)**.** Works with both LIST:GEN DSEQ and LIST:GEN SEQ commands. Related Commands: LIST:GEN. LIST:SEQ, LIST:DWEL?. (See example, Figures B-2 and B-3.)

B.35 [SOURce:]LIST:DIRection? QUERY LIST:DIR?

Syntax: Short Form: LIST:DIR? Long Form: LIST:DIRection?
Return Value: <value> (UP or DOWN)

Description: **Identifies the direction (up or down) for executing the list established by LIST:DIR.** Related Commands: LIST: DIR. (See example, Figure B-3.)

B.36 [SOURce:]LIST:DWELl COMMAND LIST:DWEL

Syntax: Short Form: LIST:DWEL <value> (0.0005 to 10),<value>,<value>, . . . to maximum of 1002 values
Long Form: LIST:DWELl <value> (0.0005 to 10),<value>,<value>, . . . to maximum of 1002 values

Description: **Determines how long the main channel parameters will be active.** Sets time value (from 0.0005 to 10) in seconds for List:Dwell locations illustrated in Table B-2. These locations are associated with the corresponding main channel locations illustrated in Table B-2. The main channel is determined by FUNC:MODE, either current (FUNC:MODE CURR) or voltage (FUNC:MODE VOLT) If LIST:DWEL is entered for only location 0, that time duration will apply to all steps when either VOLT:MODE LIST or CURR:MODE LIST is executed. Related Commands: VOLT:MODE, FUNC:MODE, LIST:CURR, LIST:VOLT, LIST:DWEL?, LIST:SEQ. (See example, Figures B-2 and B-3.)

B.37 [SOURce:]LIST:DWELl? QUERY LIST:DWEL?

Syntax: Short Form: LIST:DWEL? Long Form: LIST:DWELl?
Return Value: <value>

Description: **Identifies the dwell times entered for the list.** Starting at location established by LIST:QUERy, returns comma-separated list of up to 16 values indicating the dwell time parameters entered. i.e., the contents of LIST:DWEL locations of Table B-2. Related Commands: LIST: DWEL, LIST:QUERy. (See example, Figure B-3.)

B.38 [SOURce:]LIST:DWELl:POINts? QUERY LIST:DWEL:POIN?

Syntax: Short Form: LIST:DWEL:POIN? Long Form: LIST:DWELl:POINts?
Return Value: <value> (0 to 1001)

Description:**Identifies the number of locations for which time values have been entered and the next location to be filled by a LIST:DWEL command.** If LIST:DWEL:POIN? returns 6, dwell times have been entered for locations 0 through 5 and location 6 is the next to be filled by a LIST:DWEL command. LIST:DWEL, LIST:DWEL:POIN. (See example, Figure B-3.)

SIMPLE STAIRCASE SAWTOOTH WAVEFORM EXAMPLE

*RST	Resets power supply to the default state.
CURR 1	Sets programmed current to 1A.
LIST:CLE	Clears all list entries by setting all pointers to 0.
LIST:VOLT -5,-4,-3,-2, -1,0,1,2,3,4,5	Create 10 steps that increases from -5V to +5V.
LIST:DWEL 2	Set the dwell time for each step to 2 seconds.
LIST:COUN 10	Determines that the list will be repeated 10 times when executed.
OUTP ON	Turns the BOP output on.
VOLT:MODE LIST	Initiates execution of the list.

Upon sending VOLT:MODE LIST, the unit outputs a sawtooth waveform that increases from -5V to +5V and repeats 10 times. Each step is 1V with a dwell time of 2 seconds. There is an abrupt change from +5V to -5V each time the list transitions from the last step to the first step of the waveform. After the waveform repeats 10 times, the list stops running and the unit output will be +5V (the last step in the list). The total list running time is equal to:
10 (from LIST COUN) x 11 (total number of list steps) x 2 seconds (dwell time of each step) = 220 seconds

To invert the waveform, when the list stops running, add the following command between OUTP ON and VOLT:MODE LIST above.

LIST:DIR DOWN	List direction is from the end to the beginning of the list.

Upon sending VOLT:MODE LIST, the unit outputs an inverted sawtooth waveform that is similar to the non inverted waveform, except it decreases from +5V to -5V and abruptly returns to +5V. After the waveform repeats 10 times, the list stops running and the unit output will be -5V (the first step in the list, and the last step executed). The total list running time is the same as before the direction was reversed.

SIMPLE STAIRCASE TRIANGLE WAVEFORM EXAMPLE

*RST	Resets power supply to the default state
CURR 1	Sets programmed current to 1A
LIST:CLE	Clears all list entries by setting all pointers to 0.
LIST:VOLT 0,1,2,3,4,5,6,7,8,9,10	Create 11 steps that increase from 0V to +10V.
LIST:VOLT 9,8,7,6,5,4,3,2,1,0	Create 10 steps that decrease from +9V to 0V. The 21-step list forms a triangle stair waveform that first increases from 0V to +10V and then decreases to 0V.
LIST:DWEL 2	Sets the dwell time for each step to 2 seconds.
LIST:COUN 10	Determines that the list will be repeated 10 times when executed.
OUTP ON	Turns the BOP output on
VOLT:MODE LIST	Initiates execution of the list.

The unit outputs a triangle stair waveform, that increases from 0V to +10V and then decrease to 0V. The waveform repeats 10 times. Each step is 1V with a dwell time of 2 seconds. The waveform stays at 0V for 4 seconds because for each repetition, the last step of the decreasing list and the first step of the increasing list are both 2 seconds at 0V. When the list stops running, the output will be 0V (the last step of the list). The total list running time is equal to:
10 (from LIST COUN) x 21 (total number of list steps) x 2 seconds (dwell time of each step) = 420 seconds

To eliminate the first step and make the waveform times even, when the list stops running, add the following command between OUTP ON and VOLT:MODE LIST above.

LIST:COUN:SKIP 1	Skip first step (0V level) after the first iteration

Upon sending VOLT:MODE LIST, the unit outputs a triangle stair waveform, that is identical to the original, except that the first step (0V) is only executed for the first iteration. After 10 repetitions the list stops running and the output is 0V. The total list running time is equal to:
10 (from LIST COUN) x 20 (total number of repeated list steps) x 2 seconds (dwell time of each step) = 400 seconds + 2 seconds (dwell time of first step, run only once) = 402 seconds.

FIGURE B-2. USING LIST COMMANDS FOR SAWTOOTH AND TRIANGLE WAVEFORMS

NOTES: Examples below are intended only to illustrate command functions. Refer to PAR. 3.5.2 for programming techniques to optimize performance.

FUNC:MODE VOLT	Initializes the power supply mode to be voltage.
LIST:CLEAR	Initializes the list processor to add entries, clears main channel (LIST:CURR or LIST:VOLT) and LIST:DWEL data tables (Table B-2) and List Sequence table (Table B-3).
LIST:DWELL .010	Sets the time duration for location 0 to be 0.010 second (Since dwell times for the rest of the locations in this sample list are not entered before running the list, the dwell time will be 0.010 second for all locations.
LIST:VOLT -20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0	Starting at location 0 (-20), up to location 10 (0), fills the list with 11 data points.
LIST:VOLT:POIN?	Returns 11. Indicates that 11 data points have been entered, and location 11 is the next location to be filled (for the 12th data point.

NOTE: See PAR. B.2 for format and accuracy of all numerical data returned.

LIST:QUERY?	Returns 0 (pointer cleared by LIST:CLE).
LIST:VOLT?	Returns -20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0 (the contents of locations 0 through 10). (See PAR. B.2 for format and accuracy of numerical data.)
LIST:VOLT 2,4,6,8,10,12,14,16,18,20	Adds 10 points to the list (location 11 through 20). List now has 21 points.
LIST:VOLT:POIN?	Returns 21 (the next location to be filled by LIST:VOLT).
LIST:VOLT?	Returns -20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0,2,4,6,8 (the contents of locations 0 through 15).
LIST:GEN SEQUENCE	Enables the execution of a user-determined sequence list
LIST:SEQ 0, 0, 0, 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	Establishes the user-determined sequence.
LIST:SEQ 19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0	Continues the user-determined list sequence. The SEQ values entered mean that if LIST:GEN SEQ is issued, the list will execute location 0 four times (-20V for 40 milliseconds) location 1 through 20 and 19 through 0 (10 milliseconds each) for a total of 44 data points when VOLT:MODE LIST is issued.
LIST:SEQ?	Returns 0,0,0,0,1,2,3,4,5,6,7,8,9,10,11,12 (LIST:SEQ locations 0 through 15).
LIST:QUERY 16	Start LIST queries from location 16.
LIST:SEQ?	Returns 13,14,15,16,17,18,19,20,19,18,17,16,15,14,13,12 (LIST:SEQ locations 16 through 31).
LIST:QUERY 32	Start LIST queries from location 32.
LIST:SEQ?	Returns 11,10,9,8,7,6,5,4,3,2,1,0 (LIST:SEQ locations 32 through 43).
LIST:COUNT 100	Specifies that when VOLT:MODE LIST is issued, the user-determined sequence will be executed 100 times. (44 steps in the list will be executed 100 times)
LIST:COUN:SKIP 4	Specifies that the first 4 steps of the sequence will only be executed the first time through the list. For the subsequent 99 iterations established by LIST:COUNT 100, location 0 (-20V) will last 10 milliseconds.
CURR 1;VOLT -20	initializes the BOP output.
OUTPUT ON	Turns the output on (BOP now provides -20 volts)
VOLT:MODE LIST	Executes the list. For 40 milliseconds the BOP outputs -20V, then output a staircase triangle wave from -20V to +20V and back down to -20V. This staircase will have a uniform spacing between voltage changes of 10 milliseconds and will repeat 100 times.
VOLT?	Returns -20 (the last step in the list set the unit to -20V.
LIST:GEN?	Returns SEQ to indicate the list will be executed in the user-determined order entered by LIST:SEQ (Table B-3).

FIGURE B-3. USING LIST COMMANDS AND QUERIES (SHEET 1 OF 2)

LIST:GEN DSEQ	Changes the list to execute sequentially from locations 0 through 20).
LIST:COUN:SKIP?	Returns 4. Skip first 4 steps after the first time through count.
LIST:COUN:SKIP 0	Specifies no steps to be skipped after the first time through the count.
LIST:COUNT?	Returns 100.
LIST:COUNT 10	Determines that the list will be repeated 10 times when executed.
LIST:COUNT?	Returns 10.
VOLT:MODE LIST	Initiates execution of the list. The unit outputs a sawtooth waveform that increases from -20V to +20V and repeats 10 times. There is an abrupt change from +20V to -20V each time the list transitions from location 20 to location 0.
VOLT?	Returns 20, (the last step in the list set the unit to 20V.
LIST:VOLT:POIN?	Returns 21 (the next location to be filled by LIST:VOLT).
LIST:VOLT 10,0,-10	Add 3 points to the list (location 21 through 23).
LIST:COUN?	Returns 10 indicating that the list will be repeated 10 times when executed.
LIST:DIR?	Returns UP (the default.)
LIST:DIR DOWN	Reverses the sequence order.
VOLT:MODE LIST	The list (24 steps) is executed. The output starts at -10V, goes to 0V, to 10V to 20V then decreases to -20V in 2V steps and repeats 10 times. The sawtooth is similar to the previous sawtooth, except the waveform is inverted and the abrupt change from -20 to +20 is executed in three steps of 10 volts.
LIST:GEN?	Returns DSEQ (default sequence))
LIST:GENERATION:SEQ	Changes sequence to user determined sequence.
VOLT:MODE:LIST	The original list of 4400 points will be executed (without the three new steps.)
LIST:QUERy 0	Sets query pointer to zero.
LIST:DWEL:POIN?	Returns 1 indicating the next dwell time will be entered in location 1.
LIST:DWELL .01,.01,.01,.01,.01,.01,.01,.01,.01	Enters dwell time of 0.01 seconds in locations 1 through 9.
LIST:DWELL 1	Enters dwell time of 1 second in location 10.
VOLT:MODE LIST	the list is not executed, the unit returns error -221,Settings Error indicating the lists are not balanced.
LIST:DWELL:POINTS?	Returns 11 to indicate the dwell list has 11 entries
LIST:VOLT:POINTS?	Returns 24 to indicate the voltage list has 24 entries.
LIST:DWELL .01,.01,.01,.01,.01,.01,.01,.01,.01,.01,.1,.1,.1	Dwell times are entered in locations 11 through 23.
DWELL:POINTS?	Returns 24 indicating there are now 24 step dwell times on the list.
LIST:QUER?	Returns 0 indicating list queries will start from location 0
LIST:QUER 18	Causes list queries to start at location 18
LIST:DWEL?	Returns .01,.01,.01,.1,.1,.1 (the dwell times for locations 18 through 23.
LIST:VOLT?	Returns 16,18,20,10,0,-10
LIST COUNT 0	List will repeat indefinitely.
VOLT:MODE LIST	The list is executed. Staircase wave now consists of 24 steps of .01 seconds except for the 0V step (location 10) which outputs 0V for 1 second. The end of the staircase has three steps of 0.1 second.
LIST:GEN?	Returns error -221,Settings Error because a LIST command cannot be executed while the LIST is still being executed.
VOLT:MODE FIX	Stops execution of the list
VOLT?	VOLT?Returns -20 (the last step in the list set the unit to -20V.
VOLT: -4	Programs output to -4V
VOLT:MODE TRAN .05	Prepares for a voltage transient lasting 0.05 seconds.
VOLT:RANG 4	Sets unit to 1/4 scale (e.g., for BOB 20-20M, maximum voltage now 5V.
VOLT: 3	Output goes to +3V for 0.05 second, then returns to -4V.

FIGURE B-3. USING LIST COMMANDS AND QUERIES (SHEET 2 OF 2)

B.39 [SOURce:]LIST:GENeration COMMAND

LIST:GEN

Syntax: Short Form: LIST:GEN (SEQ | DSEQ)
Long Form: LIST:GENeration (SEQuence | DSEQuence)

Description: **Establishes the order for executing the list.** Selects either default sequence (DSEQ) or a user-determined sequence (SEQ).

DSEQ is the default sequence shown in Table B-2 and Table B-3, 0 through 1001 When LIST:GEN DSEQ is issued, the data points are executed in order either from beginning (location 0) to end (the last location with data, up to location 1001), or from end to beginning (location 0). The order is established by LIST:DIR command.

SEQ allows the list to be executed by an arbitrary sequence (up to 512 steps) determined by LIST:SEQ and LIST:DIR. Related Commands: LIST:SEQ, LIST:DIR. (See example, Figure B-3.)

TABLE B-3. LIST SEQUENCE TABLE

LOCATION	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	. . . . .	511	512		1001
DSEQ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	. . . . .				1001
SEQ																			. . . . .		NOT AVAILABLE		

B.40 [SOURce:]LIST:GENeration? QUERY

LIST:GEN?

Syntax: Short Form: LIST:GEN? Long Form: LIST:GENeration?
Return Value: <value> (SEQ or DSEQ)

Description: **Identifies the sequence for executing the list.** Returns DSEQ for the default sequence, SEQ for user determined sequence. Related Commands: LIST:GEN. (See example, Figure B-3.)

B.41 [SOURce:]LIST:QUERy COMMAND

LIST:QUER

Syntax: Short Form: LIST:QUER <int_value>
Long Form: LIST:QUERy <int_value>
int_value = 0 to 1001

Description: **Determines first location to be queried by LIST:SEQ? query.** Related Commands: LIST:SEQ?. LIST:QUER?. (See example, Figure B-3.)

B.42 [SOURce:]LIST:QUERy? QUERY

LIST:QUER?

Syntax: Short Form: LIST:SEQ? Long Form: LIST:SEQuence?
Return Value: <int_value>

Description: **Identifies first location to be queried by LIST:SEQ?, LIST:VOLT?, LIST:CURR?, LIST:DWEL? queries.** Related Commands: LIST:QUER, LIST:SEQ. (See example, Figure B-3.)

B.43 [SOURce:]LIST:SEQuence COMMAND

LIST:SEQ

Syntax: Short Form: LIST:SEQ <int_value>, <int_value>, etc. to max. of 512 locations
Long Form: LIST:SEQuence <int_value>, <int_value>, etc. to max. of 512 locations
int_value = 0 to 511

Description: **Determines the order to execute data points of the list when LIST:GEN SEQ is issued The LIST:SEQ command is provided for backward compatibility. It is not recommended to be used in any new designs as it may be eliminated at some point in the future.** Integer values fill the SEQ cells of the List Sequence Table (see Table B-3) in order, starting at location 0. For example, for a list with five points, if LIST:SEQ 4,2,1,3,0 is followed by LIST:GEN SEQ, the list will start from data point 4 (see Table B-2), then execute points 2, 1, 3, 0. (On the other hand, if LIST:GEN DSEQ is issued, data points are executed sequentially starting at 0, then 1, 2, 3, 4.) SEQ steps can be arranged to repeat steps in any order; for example a 17-step sequence for a list having five data points may be as follows: LIST SEQ 0,1,2,3,4,5,4,3,2,1,0,5,5,5,1,1,1 Related Commands: LIST:GEN. LIST:DIR, LIST:COUN, LIST:COUN:SKIP. (See example, Figure B-3.)

B.44 [SOURce:]LIST:SEQuence? QUERY LIST:SEQ?

Syntax: Short Form: LIST:SEQ? Long Form: LIST:SEQuence?
Return Value: <value1>, <value2>, . . . to <value16>

Description: **Identifies the user-determined sequence for executing the list.** Starting at location established by LIST:QUERy, returns comma-separated list of up to 16 values indicating the user-determined sequence for executing the list, i.e., the contents of the SEQ locations of Table B-3. Related Commands: LIST: SEQ, LIST:QUERy. (See example, Figure B-3.)

B.45 [SOURce:]LIST:VOLTage COMMAND LIST:VOLT

Short Form: LIST:VOLT[:LEV] <exp_value>, <exp_value>, . . . (to max of 1002 data points)
Long Form: LIST:VOLTage[:LEVel] <exp_value>, <exp_value>, . . . (to max of 1002 data points)
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Adds the voltage value (in Volts) to list.** This command sequentially adds LIST:VOLTage values to the main channel List Data Table locations illustrated in Table B-2. LIST:CLE sets starting location to 0. Next location indicated by LIST:VOLT:POIN? These locations correspond to the default sequence (LIST:GEN DESQ) The maximum number of entries is 1002. Since the input buffer of the BIT 4886 has a limit of 253 characters, multiple commands are necessary to complete the full 1002 entries of the list. If LIST:CURR has any entries, an error message: -221,"Settings conflict" is posted in the error queue. Related Commands: LIST:VOLT:POIN?, FUNC:MODE, LIST:CLE, *RST. (See example, Figures B-2 and B-3.)

B.46 [SOURce:]LIST:VOLTage? QUERY LIST:VOLT?

Syntax: Short Form: LIST:VOLT? Long Form: LIST:VOLTage?
Return Value: <value1>, <value2>, . . . to <value16>

Description: **Identifies the parameters (main channel) entered for the list.** Starting at location established by LIST:QUERy, returns comma-separated list of up to 16 values indicating the main channel parameters entered. i.e., the contents of the main channel locations of Table B-2. Related Commands: LIST: VOLT, LIST:QUERy. If LIST:CURR has any entries, an error message: -221,"Settings conflict" is posted in the error queue. (See example, Figure B-3.)

B.47 [SOURce:]LIST:VOLTage:POINts? QUERY LIST:VOLT:POIN?

Syntax: Short Form: LIST:VOLT:POIN? Long Form: LIST:VOLTage:POINts?
Return Value: <value> (0 to 1001)

Description: **Identifies the total number of points in a list and the next location to be filled by LIST:VOLT command.** The LIST:VOLT pointer is initially at 0 via *RST or LIST:CLE. For each data point entered by a LIST:VOLT command the list pointer is incremented If LIST:VOLT:POIN? returns 5, the LIST:VOLT pointer is at 5 indicating there are 5 data points comprising the list (locations 0 though 4) and location 5 is the next to be filled. If LIST:CURR has any entries, an error message: -221,"Settings conflict" is posted in the error queue. Related Commands: LIST:VOLT. (See example, Figure B-3.)

B.48 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude] COMMAND CURR

Syntax: Short Form: [SOUR:]CURR[:LEV][:IMM][:AMP] <exp_value>
Long Form: [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude] <exp_value>
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Sets programmed current level at power supply output.** This command programs output current to a specific value; actual output current will depend on load conditions. If CURR:RANG is set to 4, any data that exceeds 1/4 of I_{MAX} causes error message -222 "data out range" to be posted to the error queue (See example, Figure B-4.)

B.49 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude] QUERY CURR?

Syntax: Short Form: [SOUR:]CURR[:LEV][:IMM][:AMP]? MIN, MAX
Long Form: [SOURce:]CURRent[:LEVel][:IMMediate][:AMPlitude]? MIN, MAX
Return Value:<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Returns either the programmed value, maximum value, or minimum value of current.** The CURR? query returns the programmed value of current. Actual output current will depend on load conditions. The CURR?MAX query returns the maximum current allowed for a particular model. CURR? MIN returns minimum current allowed for power supply (always negative). **Related Commands:** CURR. (See example, Figure B-4.)

NOTES: 1. The power supply is assumed to be operating in Voltage mode.
2 Examples below are intended only to illustrate command functions. Refer to PAR. 3.5.2 for programming techniques to optimize performance.

OUTP ON	Turns the output on.
VOLT 21; CURR 1.5	Since power supply is in voltage mode, output is programmed to go to 21V, current limit 1.5A.
INIT:CONT ON	Continuous triggers enabled.
INIT:CONT?	Power supply returns “1.”
VOLT:TRIG 15;CURR:TRIG 3	Power supply output programmed to return to 15V, current limit 3A upon receipt of trigger.
*TRG	Power supply output returns to 15V,current limit 3A.
VOLT 21; CURR 5E-2	Power supply output programmed to go to 21V, current limit 0.05A.
MEAS:VOLT?	If actual value of output voltage is 20.9V, power supply returns 2.09E1.
MEAS:CURR?	If actual value of output current is 0.0483A, power supply returns 4.83E-2.
FUNC:MODE CURR	Establishes Current mode as power supply operating mode.
VOLT 21; CURR 1.1	Power supply programmed to voltage limit 21V, 1.1A.
CURR?	Returns 1.1.
FUNC:MODE VOLT	Establishes Voltage mode as power supply operating mode.
CURR:TRIG?	Returns 3 (current value established by CURR:TRIG).
VOLT:TRIG?	Returns 15 (voltage value established by VOLT:TRIG).
TRIG	Power supply output returns to 15V, current limit 3A.
INIT:CONT 0	Triggers disabled.
INIT:CONT?	Power supply returns “0.”
VOLT 0	Power supply output programmed to go to 0V.
MEAS:VOLT?	Returns 0. (measured output voltage).
VOLT?	Returns 0.(programmed output voltage).
CURR?	Returns 1.5 (programmed current).
MEAS:CURR?	Returns 0. (measured output current).
CURR:RANG 4	Set the current range to 1/4 of full scale. The maximum current for a 100-1 BOP is now 0.25ampere.
CURR .3	Selecting a range greater than 0.25 results in no change of output current.
CURR .2	Sets power supply to deliver 0.2 ampere at 0.024% accuracy.
CURR AUTO	The power supply remains in 1/4 scale.
CURR .5	Sets the output to be 1/2 ampere in the high range.
CURR:RANG?	Unit returns a 1 indicating unit range is full scale.

FIGURE B-4. PROGRAMMING THE OUTPUT

B.50 [SOURce:]CURRent:MODE COMMAND CURR:MODE

Syntax: Short Form: [SOUR:]CURR:MODE (FIX | LIST | TRAN) nn
Long Form: [SOURce:]CURRent:MODE (FIXed | LIST | TRANsient) nn
nn = <value> = time in seconds for transient

Description: **Allows the user to execute or stop a list, or to execute a transient.** The default mode is FIXed: the power supply executes commands normally, and LIST commands can be issued to establish the parameters and sequence of a list.

When CURR:MODE LIST is issued, a list is executed (See LIST commands and Figure B-3) using the voltage limit setting that is in effect when CUIRR:MODE LIST is issued. If the list runs to completion, the settings of the next to last step of the list will be in effect and the voltage limit setting will be unchanged. While the list is being executed, LIST:VOLT, LIST:CURR and FUNC:MODE commands are not accepted and will produce a command error or settings conflict error.

Issuing CURR:MODE FIX while the list is running will stop the execution of the list and set the power supply to 0V and 0A. The voltage limit setting will be set to 0V.

CURR:MODE TRANs nn causes the next CURR: command to produce a transient pulse of nn seconds duration, after which the current reverts back to the previous setting (Similar to VOLT:MODE TRANs (see PAR. B.59 for examples). If the time nn exceeds 10 seconds, error message -222 “data out range” is posted to the error queue. **Related Commands:** LIST commands. (See example, Figures B-2 and B-3.)

B.51 [SOURce:]CURRent:MODE? QUERY

CURR:MODE?

Syntax: Short Form: [SOUR:]CURR[:LEV]:MODE?
Long Form: [SOURce:]CURRent[:LEVel]:MODE?
Return value: FIXED or LIST or TRANS

Description: **Identifies active current mode.** Returns LIST while list is being executed. Returns TRANSIENT after CURR:MODE:TRAN command has been issued, but before CURR: or *TRG command executes the transient. Returns FIXED while in fixed (default) mode of operation. **Related Commands:** LIST commands. (Similar to VOLT:MODE?, see Figure B-3.)

B.52 [SOURce:]CURRent[:LEVel]RANGe COMMAND

CURR:RANG

Syntax: Short Form: [SOUR:]CURR[:LEV]:RANG <value>
Long Form: [SOURce:]CURRent[:LEVel]:RANGe <value>
<value> = 1 (full scale) or 4 (1/4 scale)

Description: **Allows the user to specify the operating range for control of output current.** Range = 1 allows control of output current from 0 to full scale. Range = 4 allows the full range of the 16 bit D to A converter to control 1/4 of the full scale output current. This command will generate a command warning error if the unit is not in current mode. It will be remembered until a func:mode command is processed. (See example, Figure B-4.)

B.53 [SOURce:]CURRent[:LEVel]RANGe? QUERY

CURR:RANG?

Syntax: Short Form: [SOUR:]CURR[:LEV]:RANG?
Long Form: [SOURce:]CURRent[:LEVel]:RANGe?

Description**: Identifies programmed current range.** Returns 1 (full scale) or 4 (1/4 scale) (See example, Figure B-4.)

CURR:RANG:AUTO

B.54 [SOURce:]CURRent:[:LEVel]RANGe:AUTO COMMAND

Syntax: Short Form: [SOUR:]CURR[:LEV]:RANG:AUTO <boolean>
Long Form: [SOURce:]CURRent[:LEVel]:RANGe:AUTO <boolean>
<boolean> = 1 (on) or 0 (off)

Description: **Allows the user to specify automatic range for control of output voltage or current.** Upon power up or upon receiving *RST, automatic ranging is selected. Automatic ranging can be turned off by VOLT:RANG:AUTO 0, CURR:RANG:AUTO 0, VOLT:RANG 1 or 4, or CURR:RANG 1 or 4. When automatic ranging is set, the correct range is automatically selected based on the programmed parameter. If the programmed parameter is more than 1/4 of nominal full scale, full scale is selected; if equal to or less than 1/4 of full scale, the 1/4 scale range is selected. For example, if auto ranging is set for a unit with 100V nominal output operating in voltage mode, a programmed voltage of up to 25.0V automatically selects range to 1/4 scale, while programming anything above 25.0V selects full scale. Function is identical to VOLT:RANG:AUTO.

B.55 [SOURce:]CURRent[:LEVel]TRIGgered[:AMPlitude] COMMAND

CURR:TRIG

Syntax: Short Form: [SOUR:]CURR[:LEV]:TRIG[:AMP] <exp_value>
Long Form: [SOURce:]CURRent[:LEVel]:TRIGgered[:AMPlitude] <exp_value>
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Programs current value to be transferred to output by *TRG or TRIG commands.** This command can be used to reset many power supplies to preselected parameters by issuing a single *TRG or TRIG command. Actual output current will depend on load conditions. If the value exceeds the maximum for the model being programmed, error message -222,"Data out of range" is posted in output queue. **Related Commands:** CURR, *TRG, TRIG (See example, Figure B-5.)

B.56 [SOURce:]CURRent[:LEVel]TRIGgered[:AMPlitude]? QUERY

CURR:TRIG?

Syntax: Short Form: [SOUR:]CURR[:LEV]:TRIG[:AMP]?
Long Form: [SOURce:]CURRent[:LEVel]:TRIGgered[:AMPlitude]?
Return Value: <exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Returns the current value established by CURR:TRIG command.** (See example, Figure B-5.)

NOTES: 1. The power supply is assumed to be operating in Voltage mode.
2 Examples below are intended only to illustrate command functions. Refer to PAR. 3.5.2 for programming techniques to optimize performance.

*RST	BOP goes to 0 volts, 0 amperes, Voltage mode.
VOLT 10;CURR 1	The voltage is placed in memory of the BOP. No Output Changes.
OUTP ON	Turns on Output, 10 Volts up to 1 ampere will be delivered.
VOLT:TRIG 1;CURR:TRIG 2	Trigger is placed in RAM.
*SAV 6	Volt 1, Current 2 and Volt mode stored in Memory location 6.
*TRG	No action - Trigger control not sent.
VOLT:TRIG 3;FUNC:MODE:TRIG CURR	No action - Trigger placed in RAM.
*SAV 7	Volt 3, curr 2, current mode saved in location 7
*RCL 6;VOLT:TRIG?	BOP returns 1 (= 1V) (saved value of memory 6)
FUNC:MODE:TRIG?	BOP returns 0 (= trigger will command voltage mode)
*RCL 6;VOLT?	BOP returns 1 (output voltage now set to 1V.).
*RCL 7;:INIT;:VOLT:TRIG?	Enables the trigger capability, unit returns 3 (= 3V)
*TRG	The voltage is set to 3, and the unit mode changes to Current
FUNC:MODE:TRIG?;:FUNC MODE VOLT	Unit switches to voltage mode. Returns 1 (= trigger will command current mode.
FUNC:MODE:TRIG?	Unit returns 0 indicating Voltage mode trigger in effect.

FIGURE B-5. USING RECALL AND TRIGGER FUNCTIONS

B.57 [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude] COMMAND — VOLT

Syntax: Short Form: [SOUR:]VOLT[:LEV][:IMM][:AMP] <exp_value>
Long Form: [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude] <exp_value>
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Sets programmed voltage level at power supply output.** This command programs output voltage to a specific value; actual output voltage will depend on load conditions. If the value exceeds the maximum for the model being programmed, error message -222,"Data out of range" is posted in output queue. (See example, Figure B-4.

B.58 [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude]? QUERY — VOLT?

Syntax: Short Form: [SOUR:]VOLT[:LEV][:IMM][:AMP]? {MIN | MAX}
Long Form: [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude]? {MIN | MAX}

Description: **Identifies programmed voltage, maximum allowable voltage, or minimum voltage (always 0).** The VOLT? query returns the programmed value of voltage. Actual output voltage will depend on load conditions. The VOLT?MAX query returns the maximum voltage allowed for a particular model. VOLT? MIN returns minimum voltage allowed for power supply (always negative). **Related Commands:** VOLT. (See example, Figure B-4.)

B.59 [SOURce:]VOLTage:MODE COMMAND — VOLT:MODE

Syntax: Short Form: [SOUR:]VOLT:MODE (FIX | LIST | TRAN) nn
Long Form: [SOURce:]VOLTage:MODE (FIXed | LIST | TRANsient) nn
nn = <value> = time in seconds for transient

Description: **Allows the user to execute or stop a list, or to execute a transient.** The default mode is FIXed: the power supply executes commands normally, and LIST commands can be issued to establish the parameters and sequence of a list.

When VOLT:MODE LIST is issued, a list is executed (See LIST commands and Figure B-3) using the current limit setting that is in effect when VOLT:MODE LIST is issued. If the list runs to completion, the settings of the next to last step of the list will be in effect and the current limit setting will be unchanged. While the list is being executed, LIST:VOLT, LIST:CURR and FUNC:MODE commands are not accepted and will produce a command error or settings conflict error.

Issuing VOLT:MODE FIX while the list is running will stop the execution of the list and set the power supply to 0V and 0A. The current limit setting will be set to 0A.

VOLT:MODE TRANs nn causes the next VOLT: command to produce a transient pulse of nn sec-

onds duration, after which the voltage reverts back to the previous setting. E.g., VOLT:25 sets the output to 25 volts, and VOLT:MODE TRAN .02 primes the unit for a transient of 0.02 seconds. When VOLT:10 is issued, the power supply output goes to 10V for 0.02 seconds, then reverts to 25V. Similarly, sending VOLT:TRIG 14, VOLT:MODE TRAN .05 and *TRG causes the output to go to 14V for 0.05 seconds, then revert to 25V. **Related Commands:** LIST commands. (See example, Figures B-2 and B-3.)

B.60 [SOURce:]VOLTage:MODE? QUERY

VOLT:MODE?

Syntax: Short Form: [SOUR:]VOLT[:LEV]:MODE?
Long Form: [SOURce:]VOLTage[:LEVel]:MODE?
Return value: FIXED or LIST or TRANS

Description: **Identifies active voltage mode.** Returns LIST while list is being executed. Returns TRANSIENT after VOLT:MODE TRAN nn command has been issued, but before VOLT: or *TRG command executes the transient. Returns FIXED while in fixed (default) mode of operation. **Related Commands:** LIST commands. (See example, Figure B-3.)

B.61 [SOURce:]VOLTage[:LEVel]:RANGe COMMAND

VOLT:RANG

Syntax: Short Form: [SOUR:]VOLT[:LEV]:RANG <value>
Long Form: [SOURce:]VOLTage[:LEVel]:RANGe <value>
<value> = 1 (full scale) or 4 (1/4 scale)

Description: **Allows the user to specify the operating range for the active mode (either voltage or current).** When in voltage mode this command establishes the voltage range, in current mode it establishes the current range. Range = 1 allows control of output voltage or current from 0 to full scale. Range = 4 allows the full range of the 16 bit D to A converter to control 1/4 of the full scale output voltage or current to provide greater accuracy. This command overrides VOLT:RANG:AUTO and turns auto ranging off. (Similar to CURR:RANG shown in Figure B-4.)

B.62 [SOURce:]VOLTage[:LEVel]:RANGe? QUERY

VOLT:RANG?

Syntax: Short Form: [SOUR:]VOLT[:LEV]:RANG?
Long Form: [SOURce:]VOLTage[:LEVel]:RANGe?
Return value: 1 (full scale) or 4 (1/4 scale)

Description: **Identifies active range.** Returns 1 (full scale) or 4 (1/4 scale) (Similar to CURR:RANG? shown in Figure B-4.)

VOLT:RANG:AUTO

B.63 [SOURce:]VOLTage[:LEVel]:RANGe:AUTO COMMAND

Syntax: Short Form: [SOUR:]VOLT[:LEV]:RANG:AUTO <boolean>
Long Form: [SOURce:]VOLTage[:LEVel]:RANGe:AUTO <boolean>
<boolean> = 1 (on) or 0 (off)

Description: **Allows the user to specify automatic range for control of output voltage or current.** Upon power up or upon receiving *RST, automatic ranging is selected. Automatic ranging can be turned off by VOLT:RANG:AUTO 0, CURR:RANG:AUTO 0, VOLT:RANG 1 or 4, or CURR:RANG 1 or 4. When automatic ranging is set, the correct range is automatically selected based on the programmed parameter. If the programmed parameter is more than 1/4 of nominal full scale, full scale is selected; if equal to or less than 1/4 of full scale, the 1/4 scale range is selected. For example, if auto ranging is set for a unit with 100V nominal output operating in voltage mode, a programmed voltage of up to 25.0V automatically selects range to 1/4 scale, while programming anything above 25.0V selects full scale. Function is identical to CURR:RANG:AUTO.

B.64 [SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude] COMMAND ?

VOLT:TRIG

Syntax: Short Form: [SOUR:]VOLT[:LEV]:TRIG[:AMP] <exp_value>
Long Form: [SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude] <exp_value>
<exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Programs voltage value to be transferred to output by *TRG or TRIG commands.** Actual output voltage will depend on load conditions. If the value exceeds the maximum for the model being programmed, error message -222,"Data out of range" is posted in output queue. If value exceeds VOLT:LIM:HIGH value, a value corresponding to the voltage limit will be programmed. (See example, Figure B-5.)

B.65 [SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude]? QUERY VOLT:TRIG?

Syntax: Short Form: [SOUR:]VOLT[:LEV]:TRIG[:AMP]?
Long Form: [SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPlitude]?
Return Value: <exp_value> = digits with decimal point and Exponent, e.g., 2.71E1 for 27.1

Description: **Returns value established by VOLT:TRIG command representing voltage value to be programmed by *TRG or TRIG command.** (See example, Figure B-5.)

B.66 STATus:OPERation:CONDition QUERY STAT:OPER:COND?

Syntax: Short Form: STAT:OPER:COND? Long Form: STATus:OPERation:CONDition?
Return Value: <int_value> 0 to 1313 (1 + 32 + 256 + 1024).

Description: **Returns the value of the Operation Condition Register (see Table B-4).** The Operation Condition Register contains unlatched real-time information about the operating conditions of the power supply. Bit set to 1 = function enabled (active, true); bit reset to 0 = function disabled (inactive, false). (See example, Figure B-6.)

TABLE B-4. OPERATION CONDITION REGISTER, OPERATION ENABLE REGISTER, AND OPERATION EVENT REGISTER BITS

CONDITION	NU	CC	NU	CV	NU	NU	NU	NU
BIT	15-11	10	9	8	7 - 6	5	4 - 1	0
VALUE	32,768 - 2048	1024	512	256	128 - 64	32	16 -2	1

CC - POWER SUPPLY IN CURRENT MODE
CV - POWER SUPPLY IN VOLTAGE MODE
NU NOT USED

B.67 STATus:OPERation:ENABle COMMAND STAT:OPER:ENAB

Syntax: Short Form: STAT:OPER:ENAB <int_value> 0 to 1313 (1 + 32 + 256 + 1024)
Long Form: STATus:OPERation:ENABle <int_value> 0 to 1313 (1 + 32 + 256 + 1024)

Description: **Sets Operation Enable Register.** The Operation Enable Register is a mask for enabling specific bits in the Operation Event Register which will cause the operation summary bit (bit 7) of the Status Byte register to be set. Bit set to 1 = function enabled (active, true); bit reset to 0 = function disabled (inactive, false). The operation summary bit is the logical OR of all the enabled bits in the Operation Event register. (See example, Figure B-6.)

B.68 STATus:OPERation:ENABle? QUERY STAT:OPER:ENAB?

Syntax: Short Form: STAT:OPER:ENAB? Long Form: STATus:OPERation:ENABle?
Return Value: <int_value> 0 to 1313 (1 + 32 + 256 + 1024).

Description: **Reads Operation Enable Register (see Table B-4).** Returns value of Operation Enable Register bits. Bit set to 1 = function enabled (active, true); bit reset to 0 = function disabled (inactive, false). (See example, Figure B-6.)

B.69 STATus:OPERation[:EVENt]? QUERY STAT:OPER?

Syntax: Short Form: STAT:OPER[:EVEN]? Long Form: STATus:OPERation[:EVENt]?
Return Value: <int_value> 0 to 1313 (1 + 32 + 256 + 1024).

Description: **Indicates changes in conditions monitored by Operational Event Register (see Table B-4).** Returns the value of the Operation Event register. The Operation Event register is a read-only register which holds (latches) all events that occur. Reading the Operation Event register clears it. (See example, Figure B-6.)

B.70 STATus:PRESet COMMAND STAT:PRES

Syntax: Short Form: STAT:PRES Long Form: STATus:PRESet

Description: **Disables reporting of all status events.** This command sets all bits of the Operation Condition (Table B-4) and Questionable Condition Registers to 0, preventing all status events from being reported. (See example, Figure B-6.)

NOTES: 1. The power supply is assumed to be operating in Voltage mode.
2 Examples below are intended only to illustrate command functions. Refer to PAR. 3.5.2 for programming techniques to optimize performance.

STAT:PRES	Operation Condition and Questionable Condition registers are reset.
STAT:QUES:ENAB 12228	Allows latching of CE and VE bits.
STAT:OPER:ENAB 1280	Mask enabled for CC and CV bits.
STAT:OPER:ENAB?	Returns 1280 (256 + 1024) (CC and CV bits set).
INIT:CONT ON	Continuous triggers enabled.
STAT:OPER:COND?	Power supply returns 256 to indicate that power supply is in Voltage mode.
STAT:OPER?	Returns 256, indicating that since the last reading of the Operation Event Register the power supply has entered Voltage mode.
STAT:OPER?	Returns 0 indicating no changes since previous reading of the Operation Event register.
STAT:QUES?	Returns 0 (no questionable conditions occurred since previous reading
--- OVERCURRENT CONDITION OCCURS	
SYST:ERR?	Power supply returns 0,“No error” message.
*RST;:VOLT 5;CURR 1;OUTP ON	The BOP is in Voltage mode and delivering 5 Volts.
*ESR?	Bop returns a 0 - all functions are normal.
FUNC:MODE CURR	BOP is in Current mode but Limit Light is on.
*ESR?;STAT:QUES:COND?	BOP returns 8;4097 - Current error.
*ESR?;STAT:QUES?	BOP returns 0;4096 - no new errors, Current Error.
*ESR?;STAT:QUES?	BOP returns 0;0 - no new errors.
MEAS:CURR?;VOLT?	BOP returns 1.0E-4;5.00003E0 - Voltage is 5 volts.
STAT:QUES:COND?	BOP returns 4097 - error still in effect.
Short is applied to the BOP.	
*ESR?:STAT:QUES:COND?	BOP returns 0;1 - Current mode is selected.
MEAS:VOLT?;CURR?	BOP returns.1E-4;1.00003E0 - Current is 1 amperes.
FUNC:MODE VOLT;*ESR?	BOP switches mode, returns 3 - settling.
*ESR?:;STAT:QUES?	BOP returns 8;8194.
Short is removed from the BOP output.	
STAT:QUES:COND?	BOP returns 2, voltage mode operation OK.

FIGURE B-6. USING STATUS COMMANDS AND QUERIES

B.71 STATus:QUEStionable[:EVENt]? QUERY

STAT:QUES?

Syntax: Short Form: STAT:QUES[EVEN]? Long Form: STATus:QUEStionable[EVENT]?
Return Value: <int_value> actual register value

Description: **Indicates the latched condition of the Questionable Event register.** Returns the value of the Questionable Event register (see Table B-5). The Questionable Event register is a read-only register which holds (latches) all events. Only bits 13 and 12 are latched in the Status Questionable Event register. Bits 0 and 1 of the Status Questionable Condition Register are not latched in the power supply. Reading the Questionable Event register clears it. (See example, Figure B-6.)

TABLE B-5. QUESTIONABLE EVENT REGISTER, QUESTIONABLE CONDITION REGISTER AND QUESTIONABLE CONDITION ENABLE REGISTER BITS

CONDITION	NU	CE	VE	NU	CM	VM
BIT	15-14	13	12	11 - 2	1	0
VALUE	32,768 -16,359	8192	4096	2048 - 4	2	1

CE CURRENT ERROR
VE VOLTAGE ERROR
CM CURRENT MODE
VM VOLTAGE MODE
NU NOT USED

B.72 STATus:QUEStionable:CONDition? QUERY STAT:QUES:COND?

Syntax: Short Form: STAT:QUES:COND? Long Form: STATus:QUEStionable:CONDition?
Return Value: <int_value> actual register value

Description: **Returns the value of the Questionable Condition Register (see Table B-5).** The Questionable Condition Register contains unlatched real-time information about questionable conditions of the power supply. Bit set to 1 = condition enabled (active, true); bit reset to 0 = condition disabled (inactive, false). Bits 1 or 0 may be both be set, indicating the power supply is settling after a voltage change. (See example, Figure B-6.)

B.73 STATus:QUEStionable:ENABle COMMAND STAT:QUES:ENAB

Syntax: Short Form: STAT:QUES:ENAB <int_value> Long Form: STATus:QUESionable:ENABle <int_value>

Description: **Programs Questionable Condition Enable Register (see Table B-5).** The Questionable Condition Enable Register determines which conditions are allowed to set the Questionable Condition Register; it is a mask for enabling specific bits in the Questionable Event register that can cause the questionable summary bit (bit 3) of the Status Byte register to be set. The questionable summary bit is the logical OR of all the enabled bits in the Questionable Event register. Bit set to 1 = function enabled (active, true); bit reset to 0 = function disabled (inactive, false). (See example, Figure B-6.)

B.74 STATus:QUEStionable:ENABle? QUERY STAT:QUES:ENAB?

Syntax: Short Form: STAT:QUES:ENAB? Long Form: STATus:QUESionable:ENABle?
Return Value: <int_value> actual register value

Description: **Reads Questionable Condition Enable Register (see Table B-5).** Power supply returns value of Questionable Condition Enable Register, indicating which conditions are being monitored. Bit set to 1 = function enabled (active, true); bit reset to 0 = function disabled (inactive, false). **Related Commands:** STAT:QUES?. (See example, Figure B-6.)

B.75 SYSTem:BEEP COMMAND SYST:BEEP

Syntax: Short Form: SYST:BEEP Long Form: SYSTem:BEEP

Description: **Causes the unit to emit a brief audible tone.**

B.76 SYSTem:COMMunication:SERial:ECHO COMMAND SYST:COMM:SER:ECHO

Syntax**:** Short Form: SYST:COMM:SER:ECHO {ON | OFF}
Long Form: SYSTem:COMMunication:SERial:ECHO {ON | OFF}

Description: **Enables (ON) or disables (OFF) echo mode** (see PAR. 3.7.1.1) Sending ON causes all subsequent characters to be echoed back. Sending OFF turns off the character echo after the next line terminator character. The *RST command has no effect on echo status. See PAR. 3.5.1 and Figure 3-23 for special programming considerations.

B.77 SYSTem:COMMunication:SERial:ECHO? QUERY SYST:COMM:SER:ECHO?

Syntax**:** Short Form: SYST:COMM:SER:ECHO?
Long Form: SYSTem:COMMunication:SERial:ECHO?
Return Value: {ON | OFF}

Description: **Identifies whether echo mode is active (ON) or disabled (OFF)** (see PAR. 3.7.1.1).

B.78 SYSTem:COMMunication:SERial:PACE COMMAND SYST:COMM:SER:PACE

Syntax**:** Short Form: SYST:COMM:SER:PACE {NONE | XON}
Long Form: SYSTem:COMMunication:SERial:PACE {NONE | XON}

Description: **Enables (XON) or disables (NONE) data flow control via the serial interface** (see PAR. 3.7.1.2). See PAR. 3.5.1 and Figure 3-23 for special programming considerations.

B.79 SYSTem:COMMunication:SERial:PACE? QUERY SYST:COMM:SER:PACE?

Syntax**:** Short Form: SYST:COMM:SER:PACE {NONE | XON}
Long Form: SYSTem:COMMunication:SERial:PACE {NONE | XON}
Return Value: {XON | NONE}

Description: **Identifies whether data flow control via the serial interface is enabled (XON) or disabled (NONE)** (see PAR. 3.7.1.2). See PAR. 3.5.1 and Figure 3-23 for special programming considerations.

B.80 SYSTem:ERRor? QUERY

SYST:ERR?

Syntax: Short Form: SYST:ERR? Long Form: SYSTem:ERRor?

Return Value: <int_value,string>

Description: **Posts error messages to the output queue.** Returns the next error number followed by its corresponding error message string from the instrument error queue. The error queue is a FIFO (first in, first out) buffer that stores errors as they occur. As it is read, each error is removed from the queue and the next error message is made available. When all errors have been read, the query returns 0,"No error". If more errors are accumulated than the queue can hold, it will overflow. The oldest errors stay in the queue but the most recent errors are discarded. The last error in the queue will be -350,"Too many errors." Error messages are defined in Table B-6.

TABLE B-6. ERROR MESSAGES

ERROR MESSAGE	ESR ERROR BIT SET (SEE PAR. A.5)	EXPLANATION
0,"No error"	None	No error
-100,"Command error"	Command Error bit 5	Command or data that followed command was not understood.
-109,"Missing Parameter"	Command Error bit 5	Command understood, required information was not included or incorrect.
-120,"Numeric data error"	Command Error bit 5	Expected number but other characters were detected.
-203,"Command Protected"	Execution error bit 4	Password must be CENAbled.
-221,"Settings Conflict"	Execution error bit 4	Calibration state not enabled but CALibrate command received; or LIST:DIR set to DOWN.
-222,"Current, Voltage or Data out of range"	Execution error bit 4	Value (current or voltage) exceeds power supply rating or (data) exceeds acceptable command parameters.
-223,"Too Much Data"	Execution error bit 4	During a LIST command, the list became full, preventing all the data from being added to the list.
-226,"Lists not same length"	Execution error bit 4	During a LIST command, number of DWEL list entries was not equal to 1 and did not match number of LIST:VOLT or LIST:CURR entries.
-240,"Hardware error"	Execution error bit 4	Power supply did not respond to command.
-241,"Hardware missing; Discovery Only"	(no ESR bit set)	Access to the visa :instr:instr VXI-11 interface was attempted. This functionality is for discovery of the device only, use in MAx manual entry to establish the socket type device at port 5025.
-301,"Voltage Error"	Device Error bit 3 (1)	Voltage output not within error window; unit set to output off -- only valid for DIAG:ERR 1, 2 or 3 mode (see PAR. 3.9.1 and Table 3-2).
-302,"Current Error"	Device Error bit 3 (1)	Current output not within error window; unit set to output off -- only valid for DIAG:ERR 1, 2 or 3 mode (see PAR. 3.9.1 and Table 3-2).
-311,"Memory Error"	Device Error bit 3 (1)	*SAV (save), *RCL (recall), or CALibrate:STORe error.
-350,"Queue Overflow"	Device Error bit 3 (1)	Error queue was full, error events have been lost.
-400,"QueryError"	Query Error bit 2	Data requested was lost due to 253 character limit of BIT 4886 output buffer.
-420,"Query Unterminated"	Query Error bit 2	Controller sent query but did not accept data from power supply. See VOLT? (PAR. B.58), CAL:SAVE (PAR. B.58) commands

(1) The Device error bit may be set when the status monitoring functions of the power supply detect an overvoltage/undervoltage condition.

B.81 SYSTem:ERRor:CODE? QUERY SYST:ERR:CODE?

Syntax**:** Short Form: SYST:ERR:CODE?] Long Form: SYSTem:ERRor:CODE?

Description: **Returns the three character error code without the ASCII definition string.** The error codes are defined in table B-6 (See example, Figure B-8.)

B.82 SYSTem:ERRor:CODE:ALL? QUERY SYST:ERR:CODE:ALL?

Syntax**:** Short Form: SYST:ERR:CODE:ALL?] Long Form: SYSTem:ERRor:CODE:ALL?
Return Value:

Description: **Returns a comma-separated list of all error codes.** A maximum of 15 codes will be returned; if the queue is empty, the power supply returns 0.

B.83 SYSTem:PASSword:CENable COMMAND SYST:PASS:CEN

Syntax**:** Short Form: SYST:PASS:CEN <val>
Long Form: SYSTem:PASSword:CENable <val>

Description: **Sets the password enable state if the value matches the current password.** This command allows other commands such as DIAG:SAV and CALibrate to operate. To reset forgotten passwords, contact the factory.

B.84 SYSTem:PASSword:CDISable COMMAND SYST:PASS:CDIS

Syntax**:** Short Form: SYST:PASS:CDIS <val> Long Form: SYSTem:PASSword:CDISable <val>

Description: **Clears the password enable state if the <value> matches the current password.**

B.85 SYSTem:PASSword:NEW COMMAND SYST:PASS:NEW

Syntax**:** Short Form: SYST:PASS:NEW <old password>,<new password>
Long Form: SYSTem:PASSword:NEW <old password>,<new password>

Description: **Establishes new password.** The old (current) password is checked, then replaced by the new password. (See example, Figure B-8.)

B.86 SYSTem:PASSword:STATe? QUERY SYST:PASS:STAT?

Syntax**:** Short Form: SYST:PASS:STAT? Long Form: SYSTem:PASSword:STATe?
Return Value: <int_value> 0 or 1

Description: **Returns a 1 if the password state is enabled or a 0 if it is disabled.**

B.87 SYSTem:REMote COMMAND SYST:REM

Syntax**:** Short Form: SYST:REM {ON | OFF} or {1 | 0}
Long Form: SYSTem:REMote {ON | OFF} or {1 | 0}

Description: **Used during serial (Telnet) communication to set the unit to remote (1 or ON) or local (0 or OFF) mode.** This command must be issued prior to commands that affect the power supply output (e.g., VOLT 10;:OUTP ON) to ensure the unit is in remote mode. See PAR. 3.7.2 and Figure B-7.

B.88 SYSTem:REMote? QUERY SYST:REM?

Syntax**:** Short Form: SYST:REM? Long Form: SYSTem:REMote?
Return Value: {1 | 0}

Description: **Identifies whether unit it in remote mode (1) or local mode (0) during serial (Telnet) communication.** See PAR. 3.7.2 and Figure B-7.

B.89 SYSTem:SECurity:IMMediate COMMAND SYST:SEC:IMM

Syntax**:** Short Form: SYST:SEC:IMM Long Form: SYSTem:SECurity:IMMediate

Description: **Initializes all NVRAM variables to factory defaults.** Empties all memory locations. This command is password protected and requires that SYST:PASS:CEN be issued before sending this command.

*IDN?	Unit responds with KEPCO,BOP 20-20,E1234,1.66 (typical).
OUTP?	Unit responds with 0 indicating output is off
SYST:REM?	Unit responds with 0 indicating unit is in local mode.
SYST:REM 1	PUTS UNIT IN REMOTE MODE.
OUTP ON	Enables output
OUTP?	Unit responds with 1 (output on).
SYST:REM 0	Unit set to local mode.

FIGURE B-7. SETTING THE UNIT TO REMOTE MODE VIA RS 232 (SERIAL) PORT

B.90 SYSTem:SET COMMAND — SYST:SET

Syntax**:** Short Form: SYSTem:SET {CM0 | CM1 | DC0 | DC1 | LF0 | LF1 | RO0 | RO1}
Long Form: SYSTem:SET {CM0 | CM1 | DC0 | DC1 | LF0 | LF1 | RO0 | RO1}

Description: **Establishes Device Clear, Line Feed, and Reset functions. Sending SYST:SEC:IMM sets LF1, DC0, and RO0 (as if CM0 was sent)**

- DC0 — Device Clear functions per IEEE 488.2 Standard: No effect on the device (power supply), only clears internal status registers.
- DC1 — Device Clear functions identical to *RST: Output set to 0V, voltage mode and output set to OFF except if RO1 (see below) is set.
- LF0 — Line Feed not provided upon empty buffer condition.
- LF1 — Line Feed provided if buffer is empty and a read is performed.
- RO0 — *RST and power up set output to OFF, 0V, 0A, voltage mode.
- RO1 — *RST sets output to ON, 0V, 0A, voltage mode. Subsequent VOLT and CURR commands are immediately present at the output without sending OUTP ON (OUTP OFF and OUTP ON function normally to turn the output off or on).
- CM0 — Establishes DC0, LF0, RO0 conditions described above (SCPI 1997 Standard compliance).
- CM1 — Establishes DC1, LF1, RO1 conditions described above. (Compatible with software versions 1.2 and earlier.) (See example, Figure B-8.)

B.91 SYSTem:SET? QUERY — SYST:SET?

Syntax**:** Short Form: SYST:SET? Long Form: SYSTem:SET?
Return Value: DC<n>,LF<n>,RO<n> where n = 0 or 1

Description: **Identifies functions established by SYST:SET command.** (See example, Figure B-8.)

B.92 SYSTem:VERSion? QUERY — SYST:VERS?

Syntax**:** Short Form: SYST:VERS? Long Form: SYSTem:VERSion?
Return Value: <int_value>.<int_value> (YYYY.V)

Description: **Identifies SCPI Version implemented.** Returns SCPI Version number:
YYYY = year, V = Revision number for specified year. (See example, Figure B-8.)

SYST:VERS?	Unit returns 1997
SYST:SET?	Unit returns DC0,LF0,RL0
SYST:PASS:NEW DEFAULT,OKAY	Unit changes password to be OKAY
SYST:SET LF1,RL1,DC1	Configures BIT 4886 card as BIT 4882
DIAG:SAV;:SYST:ERR:CODE?	BOP returns -231 indicating command is protected
SYST:PASS:CEN OKAY	Password enabled commands are accepted
DIAG:SAV	Unit saves the LF1 state for next power on cycle.

FIGURE B-8. USING SYSTEM COMMANDS AND QUERIES