INSTRUCTION MANUAL FOR

RSTL 488
PROGRAMMER

83-468-004 Revision J

SERIAL NUMBER _____

LAMBDA EMI

405 ESSEX ROAD, NEPTUNE, NJ 07753

TEL: (732) 922-9300
FAX: (732) 922-9334

LAMBDA⟨λ⟩ EMI

MANUFACTURER'S PRODUCT DECLARATION

INTENDED PURPOSE (USE)

The Power Supplies described by this manual are defined by Lambda EMI as a component for use in the composition of an apparatus as defined in Article 1 (1) of the EMC Directive (89/336/EEC). These products, as individual components, do not perform in themselves a direct function for the user of the end product. They are not intended to be placed on the market with a direct function to a final user! As such, the products described by this manual are not subject to the provisions of the EMC Directive (89/336/EEC, with amendment 92/31/EEC).

The products described by this manual are intended for incorporation into a final product by a professional assembler. It is the responsibility of the assembler to ensure that the final apparatus or system incorporating our products complies with all relevant EMC standards for that final product.

OPERATING ENVIRONMENT

The operating environment as defined by Lambda EMI, for the products described by this manual is stated as follows:

The Power Supplies described by this manual are intended for use in a protected industrial environment or in proximity to industrial power installations. These locations are often referred to as industrial locations containing establishments that are not connected to the low voltage public mains network.

Industrial locations are characterized by the existence of one or more of the following conditions:

1) industrial, scientific and medical (ISM) apparatus are present;
2) heavy inductive or capacitive loads are frequently switched;
3) currents and associated magnetic fields are high;
4) location supplied by their own transformer.

These components are not intended for connection to a public mains network, but are intended to be connected to a power network supplied from a high or medium-voltage transformer dedicated for the supply of an installation feeding manufacturing or similar operations. They are suitable for use in all establishments other than domestic and those directly connected to a low voltage power supply network which supplies buildings used for domestic purposes.

LAMBDA⚠ EMI

Description of symbols used in product labeling

SYMBOL	PUBLICATION	DESCRIPTION
$\mathbf{C}\boldsymbol{\epsilon}$	EC Council directive 93/68/EEC	European Community Conformity Assessment Product Mark
⚠	IEC 348	Attention, consult accompanying documents
⚡	IEC 60417-1-5036	Dangerous voltage
⏚	IEC 60417-1-5019	Protective earth (e.g power line earth ground)
⏛	IEC 60417-1-5017	Functional earth (e.g. chassis ground)
⚠	IEC 60417-1-5134	Electrostatic Discharge (ESD) Sensitive Device

TWO YEAR WARRANTY

Lambda EMI (405 Essex Road, Neptune, N.J. 07753), warrants that the unit is free from defects in material or workmanship for a period of TWO YEARS from the date of initial shipment. Lambda EMI will service and, at its option, repair or replace parts which prove to be defective. This will be done free of charge during the stated warranty period. This warranty excludes defects resulting from misuse, unauthorized modification, operation outside the environmental or safety specifications of the power supply, or improper site preparation or maintenance. The customer shall contact Lambda EMI, for warranty service or repair as described in the RETURNING EQUIPMENT section. The customer shall prepay shipping charges. If the repair is under the warrantee agreement, then Lambda EMI shall pay the return shipping charges.

The "WARRANTY", "CLAIM FOR DAMAGE IN SHIPMENT", and "RETURNING EQUIPMENT" information applies to equipment purchased directly from Lambda EMI. End users receiving equipment from a third party should consult the appropriate service organization for assistance with these issues.

CERTIFICATION

All test and measuring equipment used by Lambda EMI for Final Acceptance Testing are traceable to primary standards certified by the National Institute of Standards and Technology.

   LETHAL VOLTAGES PRESENT!

All power supplies contain hazardous voltage and energy. The power supply must only be operated by qualified personnel who have read this operator's manual and are familiar with the operation, hazards and application of the power supply. Proper care and judgment must always be observed.

1. Before connecting input AC power, ensure all covers are in place and securely fastened. Ensure the required safety ground to chassis is installed and sufficient cooling is supplied.

2. Proper grounding from the input AC power is required to reduce the risk of electric shock, and to comply with safety agency and code requirements.

3. Use extreme caution when connecting input AC power. Only apply the input voltage specified on the rating label.

4. Use extreme caution when connecting any high voltage cables. Never handle any output cables when the power supply is operating.

5. After a power is switched OFF, its output section will retain a charge which may be lethal. Allow sufficient time for self-discharge before handling anything connected to the output. The discharge time specified in the Safety Notes does *NOT* include extra time required to discharge the energy stored in the user's load.

6. When user serviceable fuses are present, always replace fuses with the same type and Volt/Amp rating.

7. Never attempt to operate the power supply in any manner not described in this manual.

8. Never remove DANGER or WARNING labels from the power supply. Replace lost or damaged labels immediately. Contact Lambda EMI Customer Service for replacement labels.

9. The power supply may be serviced only by Lambda EMI factory qualified service personnel. Breaking the warranty seal will void the warranty. Prior to opening the power supply, contact Lambda EMI Customer Service for a written Service Waiver and a replacement warranty seal.

ELECTRICAL STANDARDS

All company primary standards are either certified or are traceable to certification by the National Institute of Standards and Technology.

CLAIM FOR DAMAGE IN SHIPMENT

This instrument received comprehensive mechanical and electrical inspection before shipment. Immediately upon receipt from the carrier, and before operation, this instrument should be inspected visually for damage caused in shipment. If such inspection reveals damage in any way, a claim should be filed with the carrier. A full report of damage should be obtained by the claim agent and this report should be forwarded to us. We will then provide a disposition of the equipment and arrange for repair or replacement.

When referring to this equipment, always include the model and serial numbers.

> **The "WARRANTY", "CLAIM FOR DAMAGE IN SHIPMENT", and "RETURNING EQUIPMENT" information applies to equipment purchased directly from Lambda EMI. End users receiving equipment from a third party should consult the appropriate service organization for assistance with these issues.**

RETURNING EQUIPMENT

Before returning any equipment to the factory, the following steps shall be taken.

1. Notify Lambda EMI at 732-922-9300 or service@lambda-emi.com. Give a full description of the difficulty including the model and serial number of the unit in question. Upon receipt of this information, we will assign a Return Material Authorization (RMA) number and provide shipping instructions.

2. The customer shall prepay shipping charges. Equipment returned to us must be packed in a manner to reach us without damage. The shipping container must be marked with the RMA number in an area approximate to the shipping label with numbers that are easy to read. All returned units that do not show the RMA number on the outside of the container will be refused.

    If the equipment is repaired within the warrantee agreement, than Lambda EMI shall pay for the return shipping to the customer.

3. For non-warranty repairs, we will submit a cost estimate for your approval prior to proceeding. The customer shall pay return shipping charges.

MECHANICAL INSTALLATION

Most power supplies are heavy and, when rack mounted, they should be supported by rails along the sides of the supply from front to rear. The rails must adequately support the unit and not block airflow. Do not support the power supply from the front panel only.

TABLE OF CONTENTS

TABLE OF CONTENTS

Description   Page

LIST OF FIGURES

1 OVERVIEW

This manual is divided into several sections with varying degrees of complexity. When designing a GPIB interface for instrumentation control there are complex hardware and software issues that must be considered, but not necessarily by everyone. This manual is written for the first time user and the more experience user. The following GETTING STARTED section will provide sufficient information to the user who simply wants to get the instruments "talking" to each other. Other sections in the manual will detail more complex commands and methods of operation for the more experienced user.

1.1 QUICK START

The RSTL is designed to functionally duplicate the voltage and current controls on the front panel of a power supply. These controls select the programming levels and provide metering of the supply output. The RSTL, when commanded to go to REMOTE mode, switches out front panel control pots (the meters still read), and asserts control of the power supply.

1.1.1 AC INPUT SELECTION

The input AC voltage has two selectable ranges: 95-130 VAC or 190-260 VAC 1Ø 47-63 Hz. The selector switch is inside the RSTL. To change the input range <u>DISCONNECT THE POWER *FIRST*</u> and then set the switch to the proper range.

1.1.2 RSTL CONNECTION TO POWER SUPPLY

The RSTL is designed to program and measure the voltage and current of an E/M power supply. The RSTL has a built in CMOS switch that allows the selection of either Local (front panel) or Remote (RSTL D/A output) control of the supply.

The connections between RSTL and an EMS or TCR or EMHP series power supply is as follows: The programming terminal strip is TB-101 on the single phase TCR, TB2 on the three phase TCR, TB3 on the EMHP and TB1 on the EMS.

- On the power supply, find the jumpers on the terminal strip at the rear of the supply.

- Remove the jumper between terminal 3, terminal 4, and terminal 5.

- Remove the jumper between terminal 9, terminal 10, and terminal 11.

NOTE: DO NOT OPERATE THE SUPPLY WITHOUT EITHER THE JUMPER LINKS INSTALLED ON THE REAR TERMINAL BLOCK OR THE RSTL CONNECTED. DAMAGE TO POWER SUPPLY MAY RESULT IF PROGRAMMING TERMINALS ARE LEFT "OPEN."

The RSTL should be connected to the power supply as follows:

FUNCTION	RSTL - J1 DB-25 PIN#	POWER SUPPLY TERMINAL #	E/M SUPPLIED CABLE COLOR
VPROG OUT	1	4	YELLOW
IPROG OUT	5	10	BLK
IREAD(-)	7	12	GRN
VREAD(+)	8	Positive Output Terminal	WHITE
VPROG(-)	14	6	BLU
IPROG(-)	15	12	VIOLET
VPROG LOCAL	16	3 & 5	ORN
IPROG LOCAL	17	9 & 11	WHT
IREAD(+)	19	14	BROWN
VREAD(-)	20	7	RED

The jumper between terminal 1 & 2 must remain on the terminal strip because remote sensing is not allowed with an RSTL option.

The jumper between terminal 6, 7, & 8 must remain on the terminal strip.

The jumper between terminal 13 & 14 must remain on the terminal strip.

(See Figure 1 for Connection Diagram.)

1.1.3   RSTL CONNECTION TO COMPUTER

The RSTL is designed to be connected to a computer via a GPIB or RS232 cable. The switch on the back of the RSTL (red DIP switch) is used to set the GPIB address and RS232 data rate. The usual default setting when shipped from the factory is address #6 and 9600 bits per second (see section 2.2 for details).

1.1.4   CONTROLLING THE POWER SUPPLY

This section should allow you to begin programming the supply, reading back its output and querying its status. There will most likely be some inaccuracy because of calibration disparities between the RSTL and the power supply under control. Each power supply is unique and the RSTL must be calibrated for each supply. Section 3.2 has all the calibration details, but this should be performed after you are familiar with the basic operation of the RSTL.

Each command sent to the RSTL must be followed by a CR LF (ASCII 13 and 10). Normally this is taken care of during the configuration of the IEEE-488 card installed in your computer. For example, the National Instruments cards have an End Of String (EOS) identifier that is controlled by the user. Try using the default setting of the IEEE-488 card, if you have problems then look to see if the end of string terminator is set correctly.

Send the command "?M" and read the result. The RSTL will return the firmware revision, model and serial number. In addition, the voltage and current of the power supply being programmed will be returned. If the current does not match your unit (which is likely) then it will need to be changed using the following commands:

"Set * Current dddd" = "S*Cdddd" = Set scaling current to dddd Amps.
dddd = 0000 to 1000 (must use four digits).

NOTE: THE VOLTAGE OF AN RSTL IS SPECIFIED WHEN THE RSTL IS ORDERED. IF THE RSTL VOLTAGE DOES NOT MATCH THE VOLTAGE OF THE POWER SUPPLY THEN STOP AND CALL ELECTRONIC MEASUREMENTS.

After the voltage and current are correctly identified, the following commands will be sufficient to test out the basic functionality of the RSTL control of the power supply:

- **"SR"** tells the RSTL to put the supply in REMOTE operating mode. In this mode the front panel controls are switched out of the control circuit and the RSTL digital to analog converters provide the control signal to the supply. These D/A converters are controlled by the following two programming commands (among others - see section 2.5 for other commands):

- **"PV%50"** tells the RSTL to program the voltage output of the supply to 50% of full scale (see 2.5 for further details).

- **"PC%100"** tells the RSTL to program the current output of the supply to 100% of full scale (see 2.5 for further details).

- **"MV"** tells the RSTL to measure the voltage output of the supply (see 2.6 for further details).

- **"MC"** tells the RSTL to measure the current output of the supply (see 2.6 for further details).

1.1.5 CONCLUSION

Now that you are able to program to and read back from the power supply it is time to move on. The next step is calibration of the RSTL to the connected power supply. Section 4.2 details this calibration. The calibration is essentially a matching of the RSTL gain and full-scale profiles (programming and readback) to the errors within the power supply. After calibration you will see very accurate tracking of the supply to your commands in both programming and readback modes.

2 INTRODUCTION

2.1 SCOPE OF MANUAL

The remainder of the manual offers increased technical details, commands and features that expand upon the few you now know.

This section of the manual contains all the instructions necessary for operation of the RSTL Controller Board. This board is a factory-installed, or stand alone, option for remote computer control of the TCR, EMS, ESS and other power systems manufactured by Electronic Measurements, Inc.

2.2 GENERAL DESCRIPTION

The RSTL provides a means to remotely control Electronic Measurements Inc. power supplies. This board is either factory installed within the supply enclosure or within its own enclosure. The RSTL communicates via the General Purpose Interface Bus (GPIB) or the RS232C serial communications link.

The RSTL allows remote control of the power supply voltage and current programming functions. Front panel voltage and current controls are remotely locked out by the user via CMOS switches on the RSTL.

Programming of the Voltage and Current channels is via 12 bit D/A converters. The offset (zero) and gain (span) trimming of the programming signals is accomplished by four 8-bit D/A's pumping current into the 12 bit D/A programming loops. The values used to trim the unit are stored in an EEPROM in the microcontroller. At power up the trim levels are restored.

Readback is via a 21-bit A/D converter. The software routines used for calibration "use up" some of this resolution, leaving an apparent 16-bit resolution. This is more accurate than needed for the great majority of supplies. Trim adjustments for OFFSET and GAIN, in both VOLTAGE and CURRENT channels, for PROGRAM and READBACK are implemented in software. Readback trim settings may be disabled at any time, the default mode at power up is TRIM ON. Trim settings are stored in nonvolatile EEPROM address space.

2.3 ELECTRICAL SPECIFICATIONS

OPERATING TEMPERATURE: 0 TO 40 DEGREES CENTIGRADE.
ISOLATION from IEEE(RS232) to POWER SUPPLY: 1000VAC.
AC INPUT: 95-130 VAC or 190-260 VAC 1Ø 47-63 Hz
IEEE-488.1: SWITCH SELECTABLE PRIMARY ADDRESS.
RS232C : 8-BIT WORD, 1 STOP BIT, 1 START BIT.
               SWITCH SELECTABLE BAUD RATE 150 TO 9600 Bd.

PROGRAMMING RESOLUTION: 12-BIT
PROGRAMMING LINEARITY ERROR: ± 1.5 LSB.
PROGRAMMING FS. DRIFT ERROR: ± 200 PPM/degree C.
PROGRAMMING ZERO DRIFT ERROR: < 100uV/degree C.
PROGRAMMING SETTLING TIME: 100uS

READBACK RESOLUTION: > 15 bits (depends on readback setpoints).
READBACK LINEARITY ERROR: < 0.005 % FSO.
READBACK DRIFT ERROR: ± 2 LSB MAX. (0 TO 40 degrees C).
READBACK CONVERSION RATE: < 100 mS.

PROGRAMMING CALIB.:   SOFTWARE (USING TRIMDACS - NO POTS)
READBACK CALIB: SOFTWARE (NO POTS)
REMOTE SHUTDOWN: DISABLE SUPPLY PWM

2.4 MECHANICAL SPECIFICATIONS

RSTL 488-1: Length x Width x Height = 16.64" x 1.94"(1/8 rack) x 5.22" (3 RU)

RSTL 488-2: Length x Width x Height = 16.562" x 19" x 1.719" (1 RU)

3 OPERATING INSTRUCTIONS

3.1 GENERAL

The RSTL is designed to functionally duplicate the voltage and current controls on the front panel of a power supply. These controls select the programming levels and provide metering of the supply output. The RSTL, when commanded to go to REMOTE mode, switches out front panel control pots (the front panel meters still read), and asserts control of the power supply.

3.1.1 AC INPUT SELECTION

The input AC voltage has two selectable ranges: 95-130 VAC or 190-260 VAC 1Ø 47-63 Hz. The selector switch is *inside* the RSTL. To change the input range disconnect the power *first* and *then* set the switch to the proper range.

3.1.2 CONNECTION TO POWER SUPPLY

The RSTL is designed to program and measure the voltage and current of an E/M power supply. The power supply manual has detailed descriptions of "Remote Programming by External Voltage" modes of operation. Essentially, the control of a power supply by applying an external (to the supply) voltage source to each (voltage and current) programming input channel. The RSTL provides these signals to the rear terminal block on the power supply. The connections between an RSTL and an EMS, TCR or EMHP series power supply is as follows:*

FUNCTION	RSTL - J1 DB-25 PIN#	POWER SUPPLY TERMINAL #	E/M SUPPLIED CABLE COLOR
VPROG OUT	1	4	YELLOW
IPROG OUT	5	10	BLK
IREAD(-)	7	12	GRN
VREAD(+)	8	Positive Output Terminal	WHITE
VPROG(-)	14	6	BLU
IPROG(-)	15	12	VIOLET
VPROG LOCAL	16	3 & 5	ORN
IPROG LOCAL	17	9 & 11	WHT
IREAD(+)	19	14	BROWN
VREAD(-)	20	7	RED

* This table only applies to EMHP Power Supplies which were NOT ordered with the "RSTL" Option.

The jumper between terminal 1 & 2 must remain on the terminal strip.

The jumper between terminal 6, 7, & 8 must remain on the terminal strip.

The jumper between terminal 13 & 14 must remain on the terminal strip.

Remove the jumper between terminal 3, 4, & 5.

Remove the jumper between terminal 9, 10, & 11.

(See Figure 1 for Connection Diagram)

3.1.3 REQUIRED SYSTEM SETUP

The RSTL, right out of the box, is able to control a power supply. However, there are some initial steps that must be taken by the user in the case where an RSTL is NOT an option on a power supply. The following applies to stand-alone RSTL programmers:

1) Connect the unit to the computer and establish communication.

2) Determine the Voltage and Current settings by sending the "?M" command and reading back the result (see 2.4). The RSTL voltage setting is part of the model number (so E/M can select the correct voltage readback components) but the current setting will usually be at 100 Amps. Current channel readback on E/M supplies is usually measured as 0-100mV across a shunt, regardless of the actual current output of the supply. It is likely that the current setting is incorrect for your power supply! The Voltage and Current settings returned by the RSTL should match those of the power supply to be controlled. If the current does NOT match, it may be changed by using the following command (see 2.3):

   S*Cdddd where dddd = 0000 to 1000 (must use four digits).

3) Calibrate the RSTL (see 3.2, 2.7 & 2.8). The RSTL is calibrated at the factory using precise voltage standards and meters to set the readback and programming parameters. This provides an accurate way of testing the RSTL, yet it does not ensure that the RSTL is perfectly calibrated to *your* unit. There are often some system offsets and gain differences in power supplies. These differences are easily calibrated out by the RSTL. This calibration should be performed as soon as proper operation is established.

3.2 GPIB AND RS232 INTERFACES

The RSTL interfaces to a computer using either GPIB or RS232C data transfer methods. See IEEE Std 488.1 for mechanical details of the GPIB connection. The RS232 port follows the 9-pin IBM AT serial port standard for DCE devices. The RSTL accepts a command string from either port and processes that string. If any messages are generated by the command string (e.g.. Voltage Readback) the message is placed in an output buffer.

After the command string is processed the contents of the output buffer are transmitted over the RS232 lines. At any time the output buffer may be read using the GPIB. All messages are terminated with a carriage-return and a line-feed character.

The RSTL may be a Talker or a Listener.

The response of the RSTL to a Device Clear Command is to zero the voltage and current programming DACs. If the power supply is in remote mode it will be zeroed. If in local mode there will be no change in supply output. However, unless new voltage and current values are programmed in, the power supply will be zeroed upon returning to remote operation.

The GPIB address and the RS232 baud rate are selected using the 8-position DIP switch on the rear of the supply. The settings are as follows:

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	FUNCTION
Bd3	Bd2	Bd1	AD5	AD4	AD3	AD2	AD1	
			X	X	X	X	X	IEEE ADDRESS
X	X	X						RS232 BAUD RATE
0	0	0						RS232 DISABLED
0	0	1						150 BAUD
0	1	0						300 BAUD
0	1	1						600 BAUD
1	0	0						1200 BAUD
1	0	1						2400 BAUD
1	1	0						4800 BAUD
1	1	1						9600 BAUD

The baud rate is user selectable. However, the RSTL requires an 8-bit word, 1 stop, and 1 start bit for proper communications. There is no hardware handshaking, a properly received character will be echoed back over the RS232 line, assuming that the echo function is enabled. The default is to echo all characters, modifiable using the SET BACKTALK command:

"SB1" = echo on, "SB0" = echo off.

Since there is a finite amount of time required to perform a command, the user should not send a second command until the first command is complete. The unit is ready for the second command when bit 4 in the serial poll register is set.

The serial poll register is used as follows:

b0 set = Not used

b1 set = Not used

b2 set = Not used

b3 set = Not used

b4 set = supply ready to accept commands, when this bit is set after a command the results of the command (e.g. MV results) are now in the output buffer.

b5 set = not used

b6 set = service request sent.

b7 set = power on initialization, reset by CLR or device clear finish. This bit simply indicates that the unit has been turned on. This bit is reset by a device clear command. If the bit then becomes set again, it is because the controller has reinitialized for some reason.

As an alternative method, the RSTL is able to assert a service request upon the completion of a command that results in some return message. The command "SQ1" will enable this feature, "SQ0" will disable it. The default is NO service request.

"SQ0" = Don't assert SRQ when ready for next command (def).
"SQ1" = Assert SRQ when ready for next command.

3.3 SET-UP COMMANDS

There are several commands that may be issued to the RSTL to select operating conditions. Each must be followed by a CR LF (ASCII 13, 10).

Upon receipt of the new RSTL the following two commands may be needed. They set the scaling values used by the RSTL in calculating correct programming and readback values. These have no effect in the HEX or %full scale modes. If not set properly the RSTL will return erroneous readback data and will program the supply incorrectly. These need only be set once by the user, the result is stored in nonvolatile memory.

"Set * Voltage dddd" = "S*Vdddd" = Set scaling voltage to dddd Volts.

"Set * Current dddd" = "S*Cdddd" = Set scaling current to dddd Amps.
             dddd = 0000 to 1000 (must use four digits).

For RSTL units ordered with a non-integer voltage (e.g 7.5 Volts), the RSTL has been modified to include an <u>implied decimal point</u>. Therefore, a 7.5 Volt setup command would be "S*0075". Note the implied decimal point between the third and fourth digit.

The set commands are as follows:

"Set Local" = "SL" = local operation (def)
"Set Remote" = "SR" = remote operation

"Set Backtalk 0" = "SB0" = RS232 echo off
"Set Backtalk 1" = "SB1" = RS232 echo on (def)

"Set Message length 0" = "SM0" = short output
"Set Message length 1" = "SM1" = verbose output (def)

"Set Trim 0" = "ST0" = disable trim adjustments
"Set Trim 1" = "ST1" = enable trim adjustments (def)

These commands may be explicitly spelled out in the command string sent to the RSTL, but only those letters that are capitalized are actually needed. The rest are ignored.

e.g. "Set Message 0" is the same as "SM0"
      "Set Remote" is the same as "SR"

Due to the limited WRITE/ERASE endurance of the EEPROM in the RSTL's microcontroller, EMI cautions users from executing any of the following commands more than 10,000 times

set_routine ( )	SD00	S*S
set_char( )	SD01	SR
do_trim_progs ( )	SD10	SL
TRVO	SD11	SFXX
TRVG	S*DV0	PVL
TRCO	S*DV1	PCL
TRCG	S*DC0	
TPVO	S*Vdddd	
TPVG	S*Cdddd	
TPCO	S*L0	
SGCXX	S*L1	
SGVXX	S*XR	

3.4 INQUIRY COMMANDS

There are several inquiry commands used to check the operating conditions of the RSTL. Each command must have as its first character a question mark (?). Each must be followed by a CR LF (ASCII 13, ASCII 10). Each of these commands will result in a message being placed in the output buffer. The length of the returned message is determined by the setting of the BACKTALK bit (see 2.3).

The inquiry commands are as follows:

"? Model" = "?M" = Place Software Revision, Model and Serial number in output buffer

"? String" = "?S" = Previous command string ?

"? Operation" = "?O" = Local or remote operation?

"? Voltage channel " = "?V" = Voltage DAC programming value (decimal)?
"? Voltage channel heX" = "?VX" = Voltage DAC programming value (hex)?

"? Voltage channel Limit" = "?VL" = Voltage programming soft limit (decimal)?
"? Voltage channel Limit heX" = "?VLX" = Voltage programming soft limit (hex)?

NOTE: Models over 1000 Volts, the limit will not set above 999.9 Volts.

"? Current channel" = "?C" = Current DAC programming value (decimal)?
"? Current channel heX" = "?CX" = Current DAC programming value (hex)?

"? Current channel Limit" = "?CL" = Current programming soft limit (decimal)?
"? Current channel Limit heX" = "?CLX" = Current programming soft limit (hex)?

NOTE: Models over 1000 Amps, the limit will not set above 999.9 Amps.

"? Trim Voltage " = "?TV" = Trimming values for voltage channel ?

"? Trim Current" = "?TC" = Trimming values for current channel ?

These commands may be explicitly spelled out in the command string sent to the RSTL controller, but only those letters that are capitalized are actually needed. The rest are ignored.

"?M" will put a message in the output buffer of the following form:
"Rev 3.0 RSTL 10-1000 Serial 91A-1234"

This is of the form : Rev #1 RSTL voltage-current Serial #2, where #1 is the firmware revision number and #2 is the unit's serial number.

"?S" will return the last command string sent to the RSTL. This can be useful for debugging command sequences or verifying correct command reception.

"?O" will put one of the following messages in the output buffer:
"L operation" or "L"
"R operation" or "R"

if unit is in SHUTDOWN mode:

"R operation SHUTDOWN" or
"R SHUTDOWN"

"?V" will put the following message in the output buffer: "PVoltage = ddd.d Volts"
"?C" will put the following message in the output buffer: "PCurrent = ddd.d Amps"

"?VL" will put the following message in the output buffer: "PVoltage Limit = ddd.d Volts"

"?CL" will put the following message in the output buffer: "PCurrent Limit = ddd.d Amps"

"?VX" will put the following message in the output buffer: "Voltage = xxx"

"?CX" will put the following message in the output buffer: "Current = xxx"

"?VLX" will put the following message in the output buffer: "PVoltage Limit = xxx"

"?CLX" will put the following message in the output buffer: "PCurrent Limit = xxx"

xxx = the hex programming values of 000 to FFF.

"?TV" will put the following message in the output buffer: "1 ww xx yyyy zzzz"

Where the first character indicates whether or not the trim adjustments are disabled:
(1 = enabled, 0 = disabled).

ww = the hexadecimal value of the voltage programming offset trim value.

xx = the hexadecimal value of the voltage programming gain trim value.

yyyy = the hexadecimal value of the voltage readback offset trim value.

zzzz = the hexadecimal value of the voltage readback gain trim value.

"?TC" will put the following message in the output buffer: "1 ww xx yyyy zzzz"

This gives the same result for the current channel as "?TV" gives for the voltage channel.

3.5 MEASUREMENT COMMANDS

The RSTL is capable of measuring the output voltage and output current of the power supply. The optional A/D converter provides over 16-bit resolution. The readback message is returned as a measure of Volts, Amps, or for fastest operation as simply the hexadecimal output of the A/D converter. The length of the returned message is determined by the setting of the BACKTALK bit (see 2.3)

The measurement commands are as follows:

"Measure V" = "MV" = measure output voltage and return result in Volts format.

"Measure V heX" = "MVX" = measure output voltage and return result in hex format.

"Measure C" = "MC" = measure output current and return result in Amps format.

"Measure C heX" = "MCX" = measure output current and return result in hex format.

"MVC" = measure output voltage and currennt and return result in Volts and Amps format.

These commands may be explicitly spelled out in the command string sent to the RSTL controller, but only those letters that are capitalized are actually needed. The rest are ignored.

Assuming a 10 Volt, 1000 Amp ESS supply at 10 Volts and 500 Amps, the follow messages would be returned using the following commands:

"MV" will put one of the following messages in the output buffer:
"Voltage = +10.000 Volts" or "+10.000"

"MVX" will put one of the following messages in the output buffer:
"Voltage = ffff" or "ffff"

"MC" will put one of the following messages in the output buffer:
"Current = 500.0 Amps" or "500.0"

"MCX" will put one of the following messages in the output buffer:
"Current = 8000" or "8000"

"MVC" will put the following message in the output buffer:

"10.000, 500.0"

Note the value of the hex string: Zero = 0000 hex
                                      Full Scale = ffff hex

3.6 PROGRAMMING AND SOFT LIMIT COMMANDS

The Voltage and Current programming channels have resolutions of 12-bits.   In addition to these two channels there are two corresponding "soft limit" values that can be programmed. The soft limit is simply a programmed maximum or limit point. If, for example, the Voltage channel is programmed to some value in excess of the soft limit, the voltage output will not be allowed to exceed the soft limit. If the soft limit is increased then the output will increase until the initial programming level is reached. The default soft limit for both channels is full scale.
NOTE: Soft limits will not set above 999.9 Volts or 999.9 Amps.

There are several methods available to program the control and limit channels, these are:
> Programming in Volts or Amps.
> Programming as a percentage of full scale.
> Programming the DAC directly in hexadecimal.

The programming channel commands are of the following forms:

3.6.1 DIRECT PROGRAMMING (VOLTS OR AMPS)

"Program Voltage 10.000" = "PV10.000" = Program Voltage to 10.000 Volts.
"Program Current 485" = "PC485" = Program Current to 485 Amps.

"Program Voltage Limit 10.000" = "PVL10.000" = Program Voltage Limit to 10.000 Volts.
"Program Current Limit 485" = "PCL485" = Program Current Limit to 485 Amps.
NOTE: Soft limits will not set above 999.9 Volts or 999.9 Amps.

3.6.2 PERCENTAGE FULL SCALE PROGRAMMING (0 TO +%99.99)

"Program Voltage %50.00" = "PV%50" = Program Voltage to %50 of full scale.
"Program Current %99.99" = "PC%99.99" = Program Current to %99.99 of full scale.
"Program Current -%0.25" = "PC-%.25" = Program Current to negative %0.25 full scale.

"Program Voltage Limit %50.00" = "PVL%50" = Program Voltage Limit to %50 of FS.
"Program Current Limit %99.99" = "PCL%99.99" = Program Current Limit to %99.99 FS.

"Program Current Limit -%0.25" = "PCL-%.25" = Program Current Limit to neg. %0.25 FS.

3.6.3 HEXADECIMAL PROGRAMMING (0000H TO FFFFH)

"Program Voltage heX 0000" = "PVX0" = Program Voltage DAC to 000h (zero).
"Program Voltage heX 7ff" = "PVX7ff" = Program Voltage DAC to 7ffh (half scale).
"Program Current heX fff" = "PCXfff" = Program Current DAC to fffh (full Scale).

"Program Voltage heX Limit 0000" = "PVXL0" = Program Voltage Limit to 000h (zero).
"Program Voltage heX Limit 7ff" = "PVXL7ff" = Program Voltage Limit to 7ffh (half scale).
"Program Current heX Limit fff" = "PCXLfff" = Program Current Limit to fffh (full Scale).

3.7 TRIM PROGRAM COMMANDS (SOFT PROGRAMMING CALIBRATION)

The RSTL does not have any potentiometers to adjust the zero and full scale programming levels of the current and voltage channels. The calibration, or "trimming" function is accomplished by using TRIMDACS to adjust these parameters.

The four trim levels are programmed using hexadecimal input coding. The adjustment range is between 00h and ffh. The nomenclature of the four trim programming parameters is (hh = hex programming level):

> TRIM VOLTAGE CHANNEL OFFSET hh
> TRIM VOLTAGE CHANNEL GAIN hh
> TRIM CURRENT CHANNEL OFFSET hh
> TRIM CURRENT CHANNEL GAIN hh

The trim programming commands are of the following forms:

"Trim Program Voltage Offset 80" = "TPVO80"=
> Set voltage offset programming D/A to hex 80 (1/2 scale).

"Trim Program Voltage Gain ff" = "TPVGff"=
> Set voltage gain programming D/A to hex ff (full scale).

"Trim Program Current Offset 40" = "TPCO40"=
> Set current offset programming D/A to hex 40 (1/4 scale).

"Trim Program Current Gain c0" = "TPCGc0"=
> Set current gain programming D/A to hex c0 (3/4 scale).

3.8 TRIM MEASURE COMMANDS (SOFT READBACK CALIBRATION)

The measurement, or readback, of the power supply's output voltage and current is accomplished by a high resolution A/D converter. To allow for software trimming of the readback levels the input to the A/D is pulled up slightly above zero and the full scale range slightly exceeds the actual signal to be measured. By taking a "snapshot" of the voltage and current sense signal at their zero and full scale points the RSTL can calculate all measured values. There are four commands used to take this snapshot, they must only be used when the output of the supply is at the appropriate level. The four trim readback parameters are:

"Trim Read Voltage Offset" = "TRVO" = Set voltage read zero point
"Trim Read Voltage Gain" = "TRVG" = Set voltage read full scale point
"Trim Read Current Offset" = "TRCO" = Set current read zero point
"Trim Read Current Gain" = "TRCG" = Set current read full scale point

The procedure to use these commands is as follows:

3.8.1 VOLTAGE READBACK ZERO

Set the voltage output of the supply to zero using the "PV" command or equivalent. Send the command "TRVO". This will store the voltage readback offset value in the EEPROM for later use in readback calculations.

3.8.2 VOLTAGE READBACK FULL SCALE

Set the voltage output of the supply to full scale using the "PV" command or equivalent. Send the command "TRVG". This will store the voltage read gain value in the EEPROM for later use in readback calculations.

3.8.3 CURRENT READBACK ZERO

Set the current output of the supply to zero using the "PC" command or equivalent. Send the command "TRCO". This will store the current readback offset value in the EEPROM for later use in readback calculations.

3.8.4 CURRENT READBACK FULL SCALE

Set the current output to full scale using the "PC" command or equivalent. Send the command "TRCG". This will store the current read gain value in the EEPROM for later use in readback calculations.

If the readback seems to be *correct toward the full scale end but erroneous toward the zero end*, then repeat the zero procedure. If the readback seems to be *correct toward the zero end but erroneous toward the full scale end*, then repeat the full scale procedure.

4 SERVICE PROCEDURES

4.1 GENERAL

The RSTL controller board has no user serviceable parts on it. Any operational difficulty with the board should be reported to Electronic Measurements Inc. The RSTL controller is shipped precalibrated from the factory, if it becomes necessary to recalibrate the operating parameters (Programming, Readback, etc...) this may be done by the user.

4.2 CALIBRATION

There are several end-user calibrations that may be made, these include:

> Programming Voltage Zero
> Programming Voltage Gain (full Scale)
>
> Programming Current Zero
> Programming Current Gain (full Scale)
>
> Readback Voltage Zero
> Readback Voltage Gain
>
> Readback Current Zero
> Readback Current Gain

The programming channels may be trimmed by approximately ± 10% using the 8-bit trim adjustments DACs. The DACs can be programmed in two ways, either directly using a hexadecimal input (00 to ff), or as a percentage of full scale (0% to 99%). Direct hex programming provides greater resolution and is the preferred method.

The trim setting should only be adjusted after the unit under test has fully warmed up (see supply manual).

The readback channels are trimmed quite differently from the programming channels. Internal to the RSTL is an AD7710 21-bit A/D converter. The input to this converter is slightly offset in the positive direction to force a zero input to cause some slight reading. The gain of the buffer-amplifier stage before the A/D is set to provide a full-scale signal at some level less than the full-scale range of the converter. This allows a software manipulation to "normalize" the readback data to the actual supply output levels.

Readback trimming is limited to about +10% max on offset adjustments, and 20% max on gain adjustments.

Items needed for RSTL controller calibration:

- RSTL Controller installed in or connected to a power supply.
- 4 1/2 digit voltmeter.
- Calibrated external shunt (optional - internal shunt is very accurate).
- Shorting strap or other load capable of handling the full current of the power supply.
- Computer with IEEE-488 or RS232 port (see 3.3 for demonstration programs).

4.2.1 INITIAL SETUP

Turn circuit breaker to OFF position.

Disconnect AC power from the power supply and RSTL (if separate unit).

If the RSTL is a separate unit, connect it to the power supply per the instructions in the power supply manual (remote voltage programming of the voltage and current channels) and paragraph 2.1.2 of this manual.

Connect IEEE or RS232 ports to computer or other IEEE-488 Controller.

Connect the AC power source to the supply and the RSTL (the RSTL will now be powered up if it is a separate unit).

Turn the power supply's AC circuit breaker to the ON position.

Establish communications between the computer and the RSTL.

Turn the power supply's AC circuit breaker to the OFF position.

4.2.2 VOLTAGE CHANNEL CALIBRATION

Either disconnect the load from the power supply (open circuit) or adjust it such that the supply will be able to put out its full voltage.

Place meter across output terminals, observing proper polarity.

Turn the circuit breaker to ON position.

Send command "SR" (see 2.3).

4.2.3 VOLTAGE PROGRAM ZERO

Send commands "PVX005" and "PCXFFF" (see 2.6).

Calculate the proper output voltage by dividing the full scale voltage of the power supply by 4096 and multiplying the result by 5 ((Vfs/4096) * 5 = zero calibration point).

The Meter should read the calculated zero calibration value ± 0.1% F.S.

Adjust TRIM PROGRAM VOLTAGE OFFSET until meter reads correct voltage. (See 2.7)

4.2.4 VOLTAGE PROGRAM FULL SCALE:

Send command "PVXFFF" and "PCXFFF" (see 2.6).

Meter should read F.S. Voltage ± 0.1% F.S.

Adjust TRIM PROGRAM VOLTAGE GAIN until meter reads correct voltage.(See 2.8)

4.2.5 VOLTAGE READBACK ZERO

Program the supply to zero voltage (e.g. "PV0").

When the output voltage is zero ± 0.1 %, send the command: "TRVO". This will store the voltage read offset value in the EEPROM for later use in readback calculations (See 2.8).

Send command "MV" to initiate a read of the voltage channel.

Check to make sure the reading is now correct (voltage should equal zero).

4.2.6 VOLTAGE READBACK FULL SCALE

Program the supply to full-scale voltage.

When the output voltage is at full-scale ± 0.1 %, send the command : "TRVG". This will store the voltage read gain value in the EEPROM for later use in readback calculations.

Send command "MV" to initiate a read of the voltage channel.

Check to make sure the reading is now correct (voltage should equal full scale).

Program and measure the voltage for several points throughout the supply's range. The programming and readback should track closely.

4.2.7 CURRENT CHANNEL CALIBRATION

Turn circuit breaker to OFF position.

Either connect the load to the power supply and adjust it such that the supply will be able to put out its full current or connect the shorting strap to the power supply. If using an external shunt connect the shunt in series with the load or shorting strap.

Turn circuit breaker to ON position.

Send command "SR".

4.2.8 CURRENT PROGRAM ZERO

Send the commands "PCX005" and "PVXFFF".

Calculate the proper output current by dividing the full scale current of the power supply by 4096 and multiplying the result by 5 ((Ifs/4096) * 5 = zero calibration point). NOTE: The full scale current value corresponds the voltage across the shunt being measured when the output current is at full scale. The typical E/M internal shunt will measure 100miliVolts at full output current. The zero calibration point is therefore equal to (100/4096)*5 = 0.122 miliVolts.
The meter should read the calculated zero calibration value ± 0.1% F.S.
Adjust TRIM PROGRAM CURRENT OFFSET until meter reads correct current (See 2.8).

4.2.9 CURRENT PROGRAM FULL SCALE

Send command "PCXFFF".
Meter should read F.S. Current ± 0.1% F.S.
Adjust TRIM PROGRAM CURRENT GAIN until meter reads correct current.

4.2.10 CURRENT READBACK ZERO:

Program the supply to zero current.

When the output current is zero ± 0.1 %, send the command: "TRCO". This will store the current read offset value in the EEPROM for later use in readback calculations.

Send command "MC" to initiate a read of the voltage channel.

Check to make sure the reading is now correct.

4.2.11 CURRENT READBACK FULL SCALE:

Program the supply to full-scale current.

When the output current is at full-scale ± 0.1 %, send the command : "TRCG". This will store the current read gain value in the EEPROM for later use in readback calculations.

Send command "MC" to initiate a read of the current channel.

Check to make sure the reading is now correct.

Program and measure the current for several points throughout the supply's range. The programming and readback should track closely.

4.3 DEMONSTRATION GWBASIC PROGRAM

The following is a GWBASIC program to operate the RSTL controller card. This program requires the use of a National Instruments GPIB interface card in an AT-type PC. In addition, the Universal Language Interface program (ULI - included with National Card) should be run prior to running this GWBASIC program. See the National Instruments manual for proper installation and setup of the IEEE-488 Interface Card.

Line 2000 is the serial poll wait routine. After a command is sent the program looks for b4 to be set in the serial poll output register.

```
10 CLS
20 OPEN "GPIB0" FOR OUTPUT AS #1
30 OPEN "GPIB0" FOR INPUT AS #2
40 PRINT #1, "ABORT""
50 PRINT #1, "GPIBEOS CR LF"
60 PRINT "ESS TEST......"
70 PRINT "UNIT UNDER TEST MUST BE AT ADDRESS 6"
80 PRINT:PRINT "Enter Command..."
90 INPUT CMD$
110 IF CMD$ = "S" THEN GOSUB 1005
111 IF CMD$ = "C" THEN GOSUB 3000
120 PRINT #1,"OUTPUT 6 ;",CMD$
125 GOSUB 2000
140 PRINT #1, "ENTER 6"
150 INPUT #2, RD$
160 PRINT RD$
170 GOTO 80
1005 PRINT#1,"SPOLL 6"
1006 INPUT#2,SPR%
1007 PRINT SPR%
1009 GOTO 1005
1010 RETURN
2000 PRINT#1,"SPOLL 6"
2010 INPUT#2,SPR%
2015 PRINT HEX$(SPR%)
2020 IF SPR% AND 16 THEN 2040
2030 GOTO 2000
```

```
2040 RETURN
3000 PRINT#1,"CLEAR 6"
3010 RETURN
```

4.4 DEMONSTRATION QUICK-C PROGRAM

The following is a QuickC program to operate the RSTL controller card. This program requires the use of a National Instruments GPIB interface card in an AT-type PC. See the National Instruments manual for proper installation and setup of the IEEE-488 Interface Card.

```
/*
            SIMPLE INTERACTIVE RSTL PROGRAM USING
                            MICROSOFT QUICK-C &
                            NATIONAL INSTRUMENTS PCII-A CARD
*/
#include <c:\qc25\include\stdio.h>
#include <c:\qc25\include\math.h>
#include <c:\qc25\include\ctype.h>
#include <c:\qc25\include\dos.h>
#include <c:\qc25\include\graph.h>
#include <c:\qc25\include\time.h>
#include <c:\qc25\include\sys\types.h>
#include <c:\qc25\include\sys\timeb.h>

/* The following is provided on the National Instruments distribution disk */
#include <c:\gpib-pc\decl.h>

/*------------------------------------------------*/
/* IEEE functions and definitions */
/*------------------------------------------------*/

#define ERR    (1<<15)    /* Error detected        */
#define TIMO   (1<<14)    /* Timeout               */
#define RQS    (1<<11)    /* Device needs service    */

extern int    ibsta,
              iberr,
              ibcnt;

/*----------------------------------------------------------------------*/
/* Application program variables passed to GPIB functions */
/*----------------------------------------------------------------------*/

char     rd[512];        /* read data buffer */
int      rstl,           /* RSTL identifier */
         brd0;           /* National GPIB board identifier */
char     spr;            /* serial poll response byte */

main()
{
void          read_rstl();
char          command[40];

       _clearscreen(_GCLEARSCREEN); /* clear display screen */
```

```c
/*--------------------------------------------------------------------------*/
/* initialize the National Controller at address 0 on the GPIB */
/*--------------------------------------------------------------------------*/

        brd0 = ibfind ("GPIB0");

        if(brd0 < 0)
        {
                printf("Didn't find brd0\r\n");
                finderr("GPIB0");
        }

        ibeos(brd0,"\n");

/*----------------------------------------------------------------*/
/* initialize the RSTL at address 6 on the GPIB */
/*----------------------------------------------------------------*/

        rstl = ibfind("DEV6");

        if(rstl < 0)
        {
                printf("Didn't find rstl at dev6\r\n");
                finderr("DEV6");
        }

        if (ibclr(rstl) & ERR)
        {
                error(); /* clear rstl, check for error */
        }
        printf("Initialising RSTL...\r\n");
        write_rstl("sr\r\n");
        for(;;)  /* simple command loop */
        {
                printf("\nEnter Command >>> ");
                gets(command);
                write_rstl(command);

                for(sp_byte = 0x00; (sp_byte & 0x10) == 0x00 ;) /* wait for ready bit to be set */
                {
                  ibrsp(rstl,&sp_byte);
                  printf("serial poll = 0x%02x\r\n",sp_byte);
                }

                read_rstl();
                printf("\n%s",rd);
                strncpy(rd," ");
        }
} /*              END OF MAIN()                  */
```

```c
/*-----------------------------------------------------------------*/
/* The following routine will write a string to the RSTL */
/*-----------------------------------------------------------------*/

int
write_rstl(char message[])
{
        if (ibwrt (rstl,message,strlen(message)) & ERR)
                error();

        if (ibwrt (rstl,"\r\n",2) & ERR)
                error();

        ibcmd (brd0,"_?",2); /* UNT UNL */
}

/*-----------------------------------------------------------------*/
/* The following routine will read a string from the RSTL */
/*-----------------------------------------------------------------*/

void
read_rstl()
{
        if (ibrd(rstl,rd,40) & ERR)
                error();
        ibcmd (brd0,"_?",2); /* UNT UNL */
}

/*-----------------------------------*/
/* National error routines */
/*-----------------------------------*/

int
finderr(char device[]) /* ibfind failure error routine */
{
   printf("Ibfind error on %s; does device or board\n",device);
   printf("name given match configuration name?\n");
}

int
error() /* general error handling routine */
{
   printf("GPIB function call error:\n");
   printf("ibsta=0x%x, iberr=0x%x,",ibsta,iberr);
   printf(" ibcnt=0x%x\n",ibcnt);
}

int
rstlerr() /* device error handling routine */
{
   printf("Device error\n");
   printf("ibsta=0x%x, iberr=0x%x,",ibsta,iberr);
   printf(" ibcnt=0x%x\n",ibcnt);
}
```

4.5 DEMONSTRATION LABWINDOWS FOR DOS PROGRAM

The following is a LabWindows for DOS program to operate the RSTL controller card. This program requires the use of a National Instrument LabWindows software package and a National Instruments GPIB interface card in an AT-type PC. See the National Instruments manual for proper installation and setup of the IEEE-488 Interface Card.

```
/*
/*      SIMPLE INTERACTIVE LABWINDOWS FOR DOS PROGRAM
/*
/*

#include "C:\LW\INCLUDE\lwsystem.h"
#include "C:\LW\INCLUDE\utility.h"
#include "C:\LW\INCLUDE\gpib.h"
#include "C:\LW\INCLUDE\formatio.h"
#include "C:\LW\INCLUDE\userint.h"
#include "C:\LW\INCLUDE\graphics.h"
#include "C:\LW\INCLUDE\analysis.h"
#include "C:\LW\INCLUDE\vxi.h"
#include "C:\LW\INCLUDE\rs232.h"

/*------------------------------------------------------------*/
/* The following are IEEE functions and definitions */
/*------------------------------------------------------------*/

#define ERR    (1<<15)    /* Error detected       */
#define TIMO   (1<<14)    /* Timeout              */
#define RQS    (1<<11)    /* Device needs service     */

int ibsta,
    iberr,
    ibcnt;

/*------------------------------------------------------------*/
/* Application program variables passed to GPIB functions */
/*------------------------------------------------------------*/

char  command[80];        /* command input buffer */
int   rstl,               /* RSTL */
      brd0;               /* National GPIB board */

int   spr;        /* serial poll response byte */

int   status;

long  stptme,strtme;
long  pretme,eltme;
long  strtime,ptime;

main()
{
```

```c
/*--------------------------------------*/
/* Local variable assignment */
/*--------------------------------------*/

void    supply_init();

char    *read_rstl();

/*------------------------------------------------------------------*/
/* initialize the National Controller at address 0 on the GPIB */
/*------------------------------------------------------------------*/

brd0 = ibfind ("GPIB0");

if (brd0 < 0)
{
    FmtOut("Didn't find board and GPIB0\r\n");
    finderr("GPIB0");
}

/*-----------------------------------------------------*/
/* Set end of string character to LineFeed, 0x0a */
/*-----------------------------------------------------*/

ibeos(brd0,0x0a);

/*-----------------------------------------------------*/
/* initialize the RSTL at address 6 on GPIB */
/*-----------------------------------------------------*/

FmtOut("Initialising RSTL...\r\n");

rstl = ibfind ("DEV6");

if(rstl < 0)
{
    FmtOut("Didn't find device at DEV6\r\n");
    finderr("DEV6");
}

if (ibclr(rstl) & ERR)   /* Clear device at address 6 */

{
    error(); /* clear rstl, check for error */
}

for(;;) /* Infinite Command and Message Readback loop */
{
    FmtOut("Enter command : ");
    ScanIn("%s",command);
    FmtOut("      Command: %s\r\n",command);
    write_rstl(command);
    FmtOut ("      Return: %s \r\n", read_rstl() );
}

}
```

```c
/*--------------------------------------*/
/* RSTL-Specific subroutines */
/*--------------------------------------*/

/* The following routine will write a string to the RSTL */
int
write_rstl(char message[])
{
    if (ibwrt (rstl,message,StringLength(message)) & ERR)
        error();
    ibwrt (rstl,"\r\n",2);
}

/* The following routine will read a string from the RSTL */
char
*read_rstl()
{
    int    xyz;
    char   read_data[80];

    for(xyz = 0 ; xyz < 80; xyz++)
        read_data[xyz]='\0';

    if (ibrd(rstl,read_data,80) & ERR)
        error();
    return(read_data);
}

/* National error routines */
int
finderr(char device[]) /* ibfind failure error routine */
{
    FmtOut("Ibfind error on %s; does device or board\n",device);
    FmtOut("name given match configuration name?\n");
}

int
error() /* general error handling routine */
{
    FmtOut("GPIB function call error:\n");
    FmtOut("ibsta=0x%x, iberr=0x%x,",ibsta,iberr);
    FmtOut(" ibcnt=0x%x\n",ibcnt);
}

int
rstlerr() /* device error handling routine */
{
    FmtOut("Device error\n");
    FmtOut("ibsta=0x%x, iberr=0x%x,",ibsta,iberr);
    FmtOut(" ibcnt=0x%x\n",ibcnt);
}
```
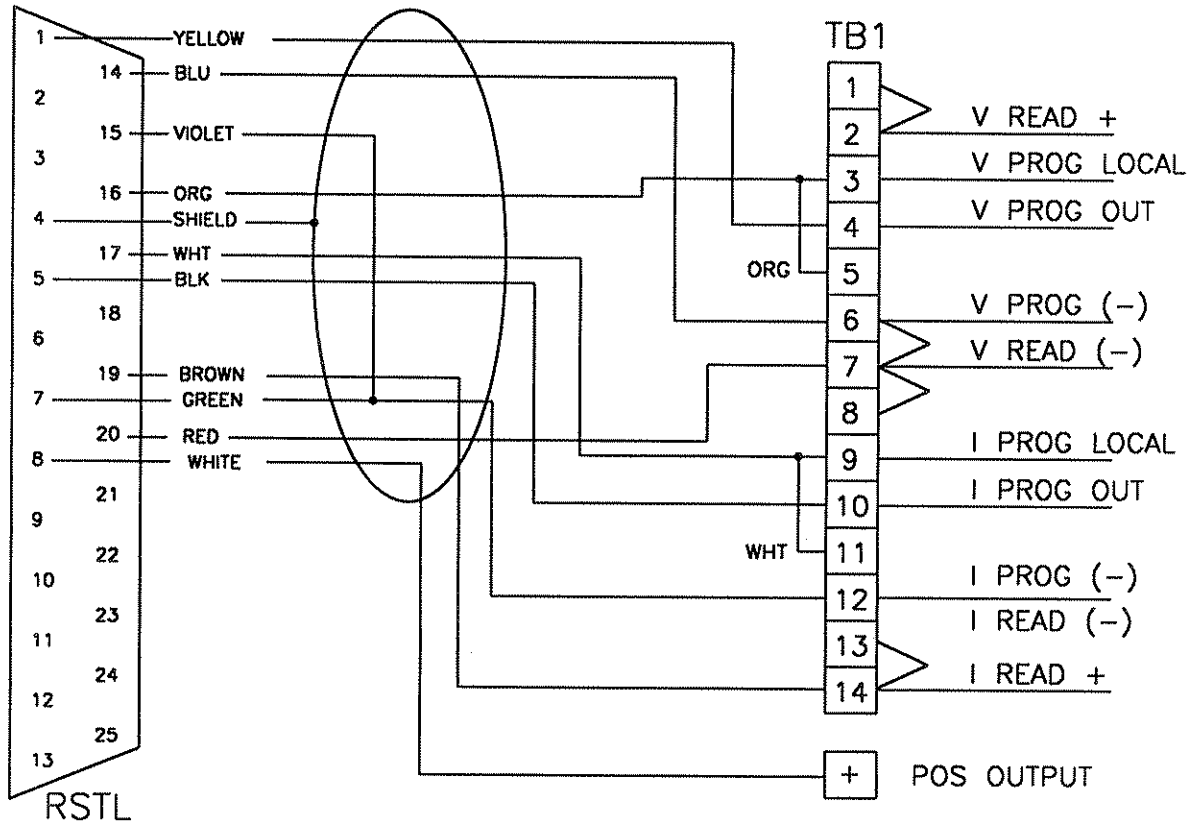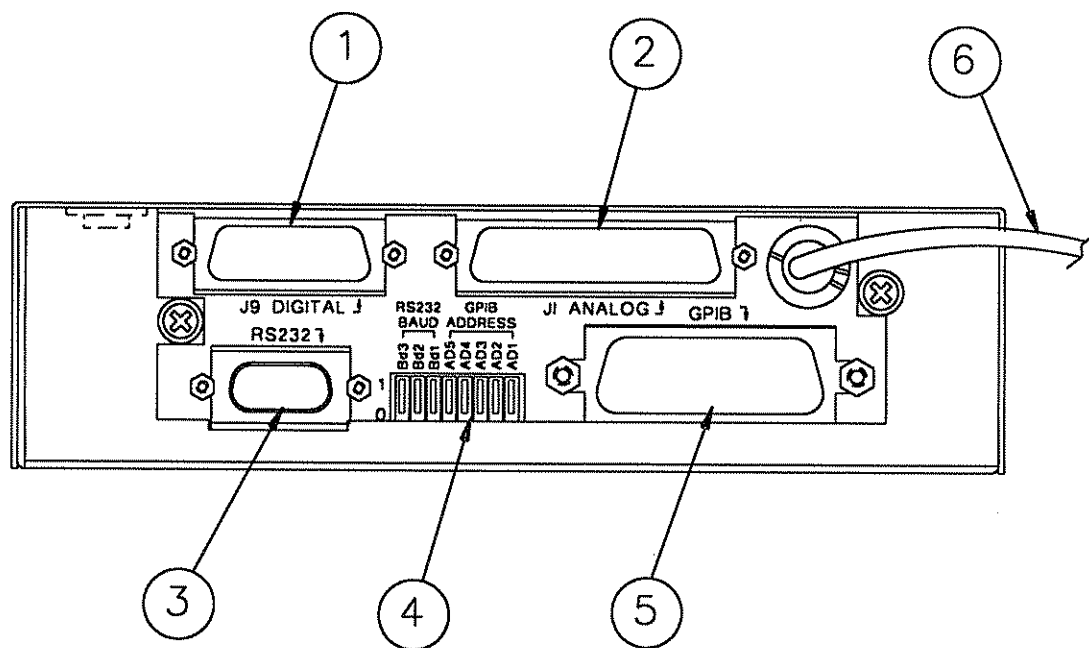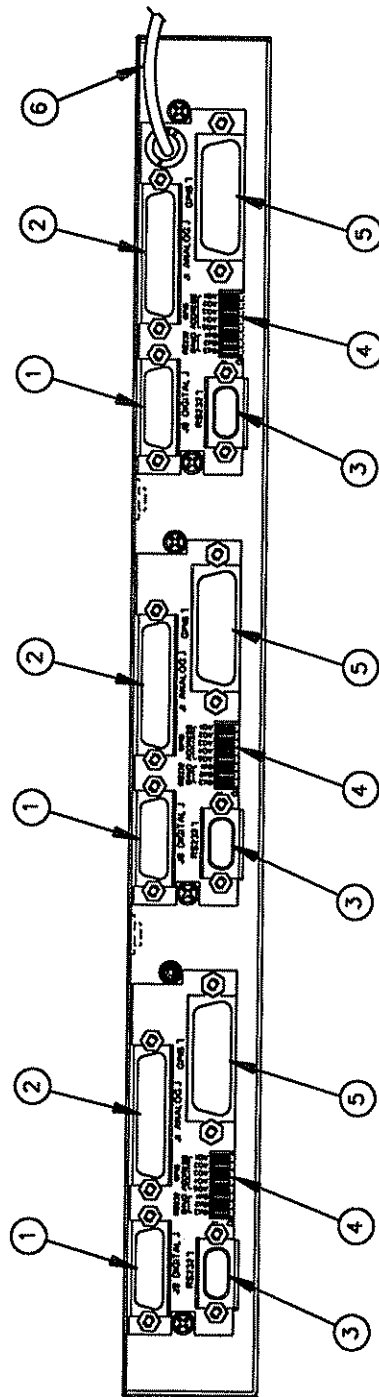
CONNECTION DIAGRAM (RSTL TO POWER SUPPLY)

Figure 1 Connection Diagram (RSTL to Power Supply)

Designator	Function
1	Digital I/o
2	Analog I/O
3	RS232
4	GPIB Address and RS232 Baud Rate Switches
5	GPIB
6	AC Input line Cord

Figure 2 Rear Panel RSTL 488-1

Figure 3 Rear Panel RSTL 488-2

Designator	Function
1	Digital I/o
2	Analog I/O
3	RS232
4	GPIB Address and RS232 Baud Rate Switches
5	GPIB
6	AC Input line Cord