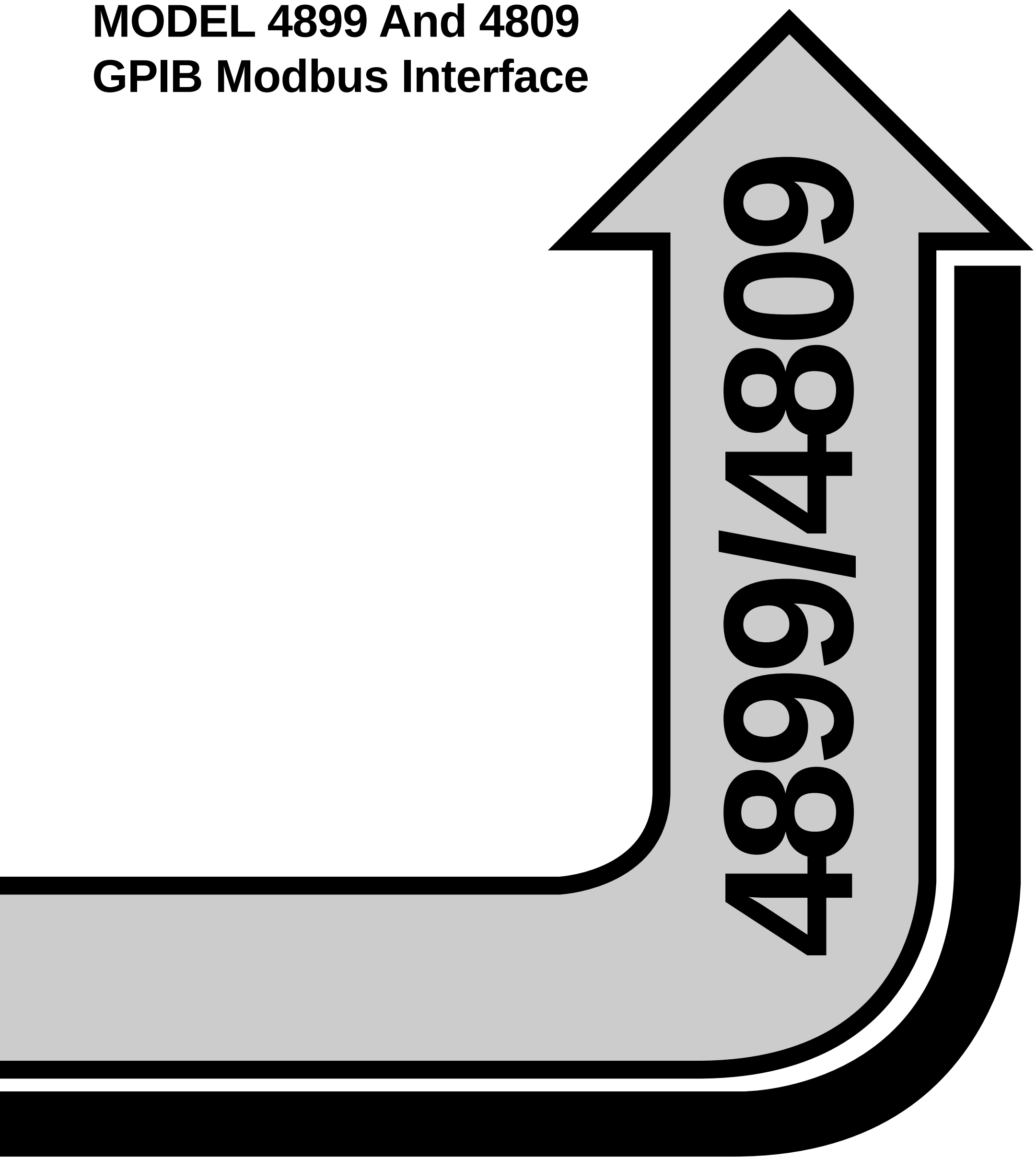
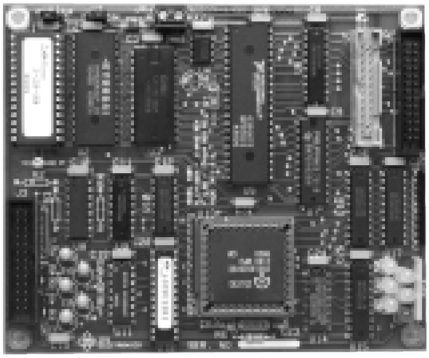
ICS
ELECTRONICS

MODEL 4899 And 4809
GPIB Modbus Interface



MODEL 4899/4809
GPIB ⟺ Modbus Interface
Instruction Manual



ICS
ELECTRONICS
a division of Systems West Inc.

473 Los Coches Street
Milpitas, CA 95035
Phone (408) 263-5500, Fax (408) 263-5896
Web Site http://www.icselect.com

Publication Number 120148
March 2000 Edition   Rev 1

LIMITED WARRANTY

Within 12 months of delivery (14 months for OEM customers), ICS Electronics will repair or replace this product, at our option, if any part is found to be defective in materials or workmanship (labor is included). Return this product to ICS Electronics, or other designated repair station, freight prepaid, for prompt repair or replacement. Contact ICS for a return material authorization (RMA) number prior to returning the product for repair.

CERTIFICATION

ICS Electronics certifies that this product was carefully inspected and tested at the factory prior to shipment and was found to meet all requirements of the specification under which it was furnished.

EMI/RFI WARNING

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. The Model 4899 has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of the FCC Rules and to comply with the EEC Standards EN 55022 and EN 50082-1, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference. The Model 4809 should be tested for RFI/EMI compliance as a component in the user's equipment.

CE Certificate of Compliance reproduced in Figure 1-4.

TRADEMARKS

The following trademarks referred to in this manual are the property of the following companies:

HP is a trademark of Hewlett-Packard Corporation, Palo Alto, CA
ICS is a trademark of Systems West Inc, Milpitas, CA
Watlow is a trademark of Watlow Electric Manufacturing Company, Winona, MN

Contents

General Information 1
Product Descriptions, Specifications, Factory Configuration, Physical Characteristics, Certifications and Accessories List.

Installation 2
Shipment Check, Installation Guide, Configuration Instructions, GPIB and Serial Connections, Internal Jumper Settings and Rack Mounting instructions.

Operation 3
Operation Description, GPIB Addressing, 488.2 Commands, SCPI Commands, Modbus Commands, Programming Guidelines and OEM Documentation.

Theory of Operation 4
Block diagram Description

Maintenance, Troubleshooting and Repair 5
Maintenance, Troubleshooting guide, Self test Error Codes and Repair Directions

Appendices A
A1 IEEE 488 Bus Description, IEEE 488.2 Message Formats, Common Commands and SCPI Commands
A2 Serial Communication and Interfacing
A3 GPIB Connector/Switch Board Description and Installation

Index I

1

1

General Information

1.1 INTRODUCTION

This section provides the specifications for ICS's Model 4899 and 4809 GPIB<->Modbus Interfaces and their accessory items. The Model 4899 is an enclosed minibox™ product designed for bench use with other equipment. The Model 4809 is a PC board assembly designed for mounting inside another piece of equipment. Both products are functionally equivalent. Wherever the text refers to the Model 4899, it also applies to the Model 4809 unless otherwise stated. Any Model 4809 differences are noted in parenthesis or in separate paragraphs.

1.2 DESCRIPTION

The Model 4899/4809 GPIB<->Modbus Interfaces are IEEE-488.2/ GPIB to Serial Interfaces with special firmware to control Modbus Slave devices from the GPIB or HP-IB bus. The 4899 and 4809 let the user send simple commands with decimal values over the GPIB bus to control and query slave Modbus devices. The 4899/4809 converts the GPIB commands into the Modbus protocol and adds the CRC checksum to make a complete Modbus RTU message. The messages are sent serially over a RS-232 link or over a RS-485 network. Responses are checked and valid response data from a query is returned to the GPIB bus when the 4899/4809 is next addressed to talk.

1

The 4899 and 4809 contain a number of advanced features that increases its flexibility and simplifies its use in system applications. Both have IEEE-488.2 compatible interfaces with expanded Status Reporting Structures that comply with the SCPI standard. SCPI commands are used to set the unit's GPIB address, to set the serial configuration, and to enable bits in the Status Reporting Structure to generate SRQs. The user can also enter his own IDN message to personalize the unit as part of his assembly. All settings are saved in nonvolatile memory.

The Model 4899 is packaged in a small minibox™ metal case that is less than 1U in height (1.6 inches) The front panel contains the power switch and LEDs which indicate the unit's status. The rear panel contains the GPIB and serial connectors and a DC power jack. The 4899 accepts a wide range of DC voltages and is shipped with an adapter for the local power lines.

The Model 4809 is a small, low-profile PC assembly that has two flat ribbon connectors for its GPIB signals and one for its serial signals. The Model 4809 can be connected to a standard GPIB connector on a pin-to-pin basis or to an small PC assembly that mounts an external address switch and GPIB connector on the host chassis rear panel. ICS provided GPIB Connector/Address Switch Assemblies are available in two mounting styles. (See Appendix 3) The 4809 has the same diagnostic LEDs as the 4899 and uses 5 volt regulated power. The 4809's GPIB address can be set by a GPIB bus command or by an external address switch.

1.3 MODEL 4899/4809 SPECIFICATIONS

The following specifications apply to all models. Options for your unit may be found by comparing the list below to those listed on the serial label on your unit.

4899 - X — General Model Number

Option Codes

-1	Special Crystal
-3	Custom Paint
-5	Custom Front Panel
-7	Special Program
-9	Factory Installed Rack Mount Kit
-A	Ship with Australian 230 Vac Adapter
-B	Ship with British 230 Vac Adapter
-E	Ship with European 230 Vac Adapter

Base Model Number

4899 - Minibox Product
4809 - Board Product

Note that because the Model 4809 is a board level product that uses only + 5 Vdc, options -3, -5, -9, and the AC adapters do not apply to the Model 4809.

1.4 IEEE 488 INTERFACE

1.4.1 488.1 Capabilities

1

The 488 Bus interface meets the IEEE STD 488.1-1987 standard and has the following capabilities:

SH1, AH1, T6, L3, SR1, PP0, DC1, RL0, DT0, C0 and E1/E2 drivers.

1.4.2 Address Ranges

Primary addresses	0 - 30

1.4.3 Buffers

GPIB input	2 kbytes
GPIB output	1 kbytes
Serial input/output	256 bytes

1.4.4 488.2 Common Commands

Standard 4899/4809s conform to IEEE STD 488.2-1987. When addressed to listen in the command mode, the unit responds to the following 488.2 commands:

*CLS, *ESE, *ESE?, *ESR?, *IDN?, *OPC, *OPC?, *PSC *RCL, *RST, *SAV, *SRE, *SRE?, *STB, *TST?, and *WAI.

1.4.5 SCPI Parser

Standard 4899/4809s include an extended SCPI parser that complies with the SCPI Standard Version 1995.0.

1.5 SERIAL INTERFACE

The 4899/4809's serial interface provide RS-232 single ended and RS-485 (RS-422) differential signals. Signals are selected by internal jumpers. The 4899 uses a DB-25F connector, the 4809 uses a 26-pin flat-ribbon header. Signal pinouts conform to EIA RS-530 specification and are listed in Table 2-2.

1.5.1 Baud Rates:

Baud Rate: Any rate from 50 to 115,200 baud. (The 4809 is limited to 38400 baud). Parser selects closest rate to specified rate when a nonstandard rate entered. Standard rates are: 50, 110, 300, 600, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 57600, 76800, 92160 and 115200 baud.

1.5.2 Data Character Formats:

Data bits	8 data bits per character
Parity	none
Type	Asynchronous character
Stop bits	1 or 2 stop bits per character

1.5.3 Modbus RTU Message Format

Messages conform to the Modbus RTU format and include the device address, command, register number, data and CRC formatted as binary bytes.

Packet Format:

Addr	Cmd	Registers.... .data	CRC

Figure 1-1 Modbus RTU Message

1

1.5.4 RS-232 Specifications

The 4899/4809 has single-ended RS-232C drivers and receivers and is designed to operate as a DTE device with up to 50 feet of cable.

Transmit Levels	+10 Vdc = Logic "0" or On -10 Vdc = Logic "1" or Off
Receive	±1.5 Vdc minimum, ±25 Vdc Maximum
Signals	AA, AB, BA, BB, CA, CB, CD and CF

1.5.6 RS-422/RS-485 Specifications

The 4899/4809 has balanced RS-485 line drivers and receivers that are RS-422 and RS-485 compatible. The line drivers and receivers are designed to operate with up to 1200 meters of twisted-pair cable. The transmitter can be set for continuous on operation or it can be tristated when not transmitting.

Modes	Transmitter always on or tristated when not transmitting
Transmit Levels	+5 Vdc differential for binary 0 or On -5 Vdc differential for binary 1 or Off
Receive Levels	±0.2 Vdc minimum, ±25 Vdc maximum, differential or single-ended input with other input line biased at mid-range.
Signals	SD, RD, RS, CS, RR and TR signal pairs

1

1.6 PROGRAMMABLE FUNCTIONS

4899/4809s use IEEE 488.2 and SCPI commands to change their programmable functions and jumpers to select the serial signals. Table 1-1 lists the programmable functions and their factory default settings. Table 1-2 lists the factory's jumper settings

TABLE 1-1 FACTORY CONFIGURATION

Command	**Functions**	**Factory Setting**
:ADDRess	Sets GPIB bus address	4
:EXT	Enables External GPIB Address Switch	OFF (4809 only)
:BAUD	Sets transmit/receive baud rate	9600
:PARity	Sets parity type	NONE
:CHECK	Enables parity checking	OFF
:BITs	Sets number of data bits per character	8
:SBITs	Sets number of stop bits/per character	1
:RS485	Tristate transmitter enabled	OFF
:FORMat	Sets talk format for response data	ASCii
*ESE	Enables Standard Event Status Register bits	0
*SRE	Enables Status Byte Register bits	0

TABLE 1-2 FACTORY JUMPER SETTINGS

Jumpers	**Functions**	**Factory Settings**
4899 W1, W2, W3	Selects RS-232 or RS-422 Signals	RS-232
4809 W4, W5	Selects RS-232 or RS-422 Signals	RS-232

1.7 INDICATORS

The 4899/4809s have six LEDs that display the following conditions:

PWR	-	Indicates power on
RDY	-	Indicates unit has passed self test
TALK	-	Indicates unit has recognized its talk address
LSTN	-	Indicates unit has recognized its listen address
SRQ	-	On when receiving serial data
ERR	-	On when either data buffer is full

When the 4899 is turned on, it performs an internal self test which takes about 0.5 seconds. Only the PWR indicator is on during self test. At the end of the self test the 4899 displays its current GPIB address by blinking the front panel LEDs for one-half second. The LED bit weights are:

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

Any errors found during self test are indicated by a repeated blinking of the error code pattern. Refer to paragraph 5.4 for a description of the errors and their possible causes.

At power turn-on, the ERR LED indicates that the unit has not been calibrated since the memory was last initialized. During normal operation, the ERR LED indicates that there was a problem with the GPIB command or the Modbus communication.

1.8 PHYSICAL

1.8.1 4899 Minibox

Size	-	7.45"L x 5.57"W x 1.52"H (18.92 cmL x 14.15 cmW x 3.86 cmH) (See Figure 1-2)
Weight	-	3 lbs (1.4 kg) including adapter
Temperature	-	Operating -10° C to +55° C Storage -40° C to +70° C
Humidity	-	0-90% RH without condensation
Shock/Vibration	-	Normal handling only
Construction	-	All metal case
Power	-	9 to 32 Vdc @ 3.5 VA
Connectors	-	IEEE 488 Interface Amphenol 57-20240 with metric studs Serial Interface Cinch DB-25F with lock studs

1

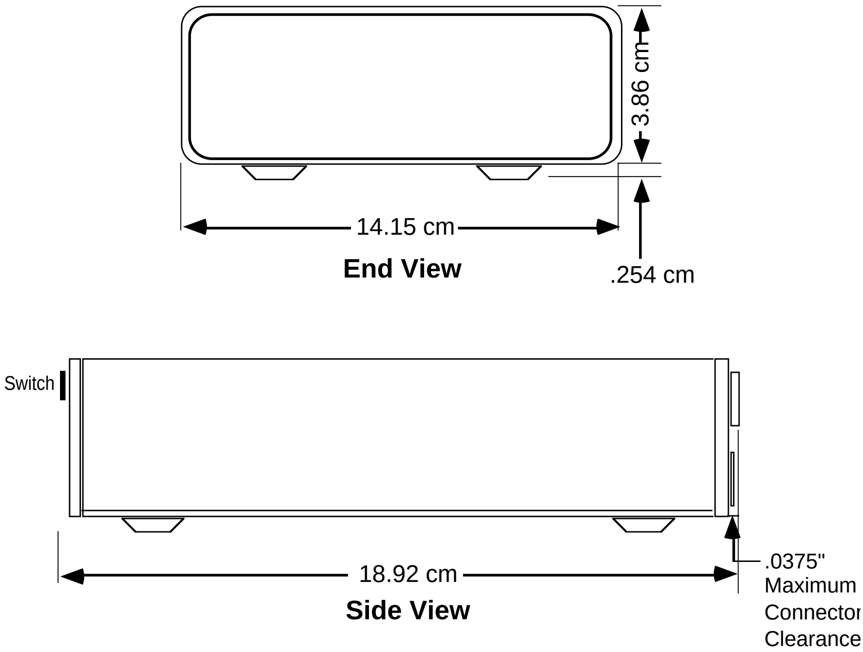


Figure 1-2 4899 Outline Dimensions

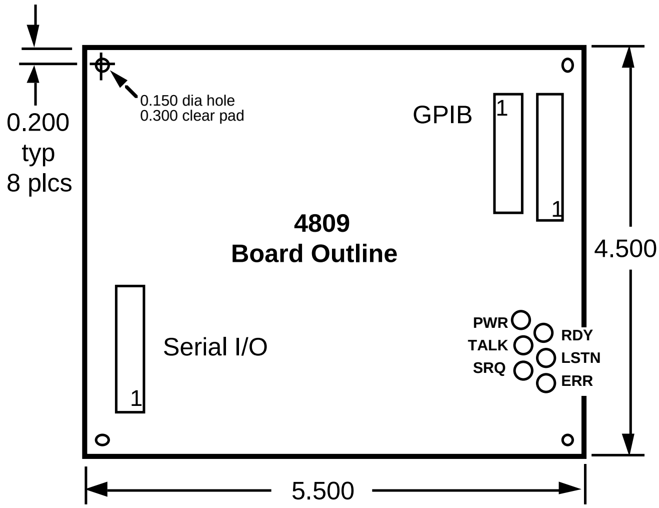


Figure 1-3 4809 Outline Dimensions

1.8.2 4809 Board Assembly

Size	-	5.50"L x 4.50"W x 0.5"H (139.7 mmL x 114.3 mmW x 12.7 mmH) (See Figure 1-3)
Weight	-	6 oz. (0.17 kg)
Temperature	-	Operating -10° C to +55° C Storage -40° C to +70° C
Humidity	-	0-90% RH without condensation
Shock/Vibration	-	Normal handling only
Construction	-	Flame-retardant printed circuit board
Power	-	5 ± 0.25 Vdc @ 400 MA (typical)
Connectors	-	GPIB - 24 pin male 3M 2524 connector
	-	GPIB/Address Sw - 26 pin male 3M 2526 connector
	-	Serial - 26 pin male 3M 2526 connector

1.9 4899 CERTIFICATIONS OR APPROVALS

EMI/RFI CE	Meets limits for part 15, Class A of US FCC Docket 20780 and complies with EEC Standards EN 55022 and 50082-1. CE Certificate of Compliance reproduced in Figure 1-4.
UL/CSA/VDE	AC Wall adapter has applicable UL/CSA/VDE and CE approvals

Certificate of Conformity

Application of Council Directives 89/336/EEC

Standard(s) to which Conformity is Declared ..... EN 55022, EN50082-1

Manufacturer's Name — ICS Electronics div
Systems West, Inc.

Manufacturer's Address — 473 Los Coches Street
Milpitas, CA 95035
USA

Importer's Name

Importer's Address

Type of equipment — GPIB to Serial Converter

Model No. — 4899

Serial No.________Thru__________ , Year of manufacturer 1999

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directive(s) and Standard(s) by virtue of its similarity with the Model 4894.

Place Milpitas, CA, USA

Date Sept. 14, 1999

[signature]
Signature

Gerald K. Mercola
(Full Name)

President
Title

Figure 1-4 4899 Certificate of Conformance

1.10 ACCESSORIES

1.10.1 4899 Included Accessories

Instruction Manual
AC Wall Adapter, with applicable plug
3.5" Minibox Configuration Program Disk for PC and PC compatible computers.

1.10.2 4809 Included Accessories

120117	4899/4809 Instruction Manual
123045	Minibox GPIB Configuration Disk for PC and PC compatible computers.

1.10.3 Optional Accessories

120117	4899/4809 Instruction Manual
123045	Minibox GPIB Configuration Disk
113640-L	Horizontal GPIB Connector/Address Switch Assembly (Dash number is cable length from 10 to 90 CM long. 90 CM standard)
113642-L	Vertical GPIB Connector/Address Switch Assembly (Dash number is cable length from 10 to 90 CM long. 90 CM standard)
114439-L	GPIB Flat Ribbon Extension Cable. (Dash number is cable length from 10 to 90 CM long. 90 CM standard)
114256-L	Serial Flat Ribbon Extension Cable from 4809 to DB-25P male connector. (Dash number is cable length from 10 to 90 CM long. 90 CM standard)

Installation

2.1 UNPACKING

When unpacking, check the unit(s) for signs of shipping damage (damaged box, scratches, dents, etc.) If the unit is damaged or fails to meet specifications, notify ICS Electronics or your local sales representative immediately. Also, call the carrier immediately and retain the shipping carton and packing material for the carrier's inspection. ICS will make arrangements for the unit to be repaired or replaced without waiting for the claim against the carrier to be settled.

2.2 SHIPMENT VERIFICATION

Take a moment to verify the shipment. Each shipment includes:

(1) Model 4899 or Model 4809 GPIB<->Modbus Interface
(1) AC Power Adapter (Model 4899 only)
(1) 4899/4809 Instruction Manual
(1) Minibox Configuration Program Disk

Board only 4809 orders (Part# 114442) do not include manuals or configuration disks unless ordered separately.

Take a moment to check ICS's website at http://www.icselect.com for any manual errata or the latest configuration programs.

2.3 4899 INSTALLATION GUIDE

The following steps should be used as a guide to setting up and using your 4899.

1. If the 4899 is to be used with RS-422 or RS-485 signals, change the jumper settings as directed in Section 2.10.

2. See Section 2.9 to select and/or design the serial cable. CAUTION - So called 'standard' serial cables should not be used with the 4899.

3. If the unit is to go into a rack mounting kit, disconnect all cables from the unit. Follow the instructions in Section 2.11 to install the unit it in the rack mounting kit.

4. Connect the AC adapter to the 4899 and to the AC power. Turn the unit on and verify that it passes its selftest and that it indicates the correct GPIB address. Query its IDN message to verify GPIB communication.

5. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured. If the 4899 needs to be reconfigured, follow the instructions in Section 2.6 or 2.7 to change its configuration.

6. Turn the 4889 off. Connect the unit to the Modbus device. Turn both units on and verify that they passes their selftest. Use a GPIB keyboard or similar program to query Modbus device's model number or some other known value. Follow the example in section 3.8.3. Try out the other Modbus commands in Section 3.7 that apply to the Modbus device.

2.4 4809 INSTALLATION GUIDE

The following steps should be used as a guide to the 4809 installation.

1. If the 4809 is to be used with RS-422 or RS-485 signals, change the jumper settings as directed in Section 2.10.

2. Review Sections 2.8 and 2.9 to select and/or design the GPIB and serial interface cables.

3. Select a convenient location to mount the 4809. Do not mount it directly over a heat producing surface. Provide a 0.1 inch (2.5 mm) clearance underneath the 4809 or use an insulator if the 4809 is being mounted on a metal surface.

4. Use a twisted pair of #24 wires to connect the 4809's power terminals to the host's +5 Vdc power supply. Connect the 4809 directly to the power supply to avoid noise problems. Attach the wires into the terminal block on the 4809 PCB. Bypass diode CR1 if necessary so TP+5 equals 5 ± 0.2 Vdc.

5. Plug in the GPIB and serial cables and connect the unit to the GPIB controller. Turn the unit on and verify that it passes its selftest and indicates the correct GPIB address. Query its IDN message to verify GPIB communication.

6. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured. If the 4809 needs to be reconfigured, follow the instructions in Section 2.6 or 2.7 to change its configuration.

7. Turn the 4889 off. Connect the unit to the Modbus device. Turn both units on and verify that they passes their selftest. Use a GPIB keyboard or similar program to query Modbus device's model number or some other known value. Follow the example in section 3.8.3. Try out the other Modbus commands in Section 3.7 that apply to the Modbus device.

2.5 CONFIGURATION PROGRAMS

When shipped, Model 4899s and 4809s are configured as shown in Table 1-1. The configuration is stored in nonvolatile E^2ROM and can be changed from any GPIB bus controller with SCPI commands. If your GPIB controller is a PC with DOS or Windows 3.1, and has one of the GPIB controller cards listed below, you can use the programs on the supplied Minibox Configuration Disk to walk you through a menu driven configuration procedure. Follow the instructions in Section 2.6 to install and use the DOS configuration programs.

Program	Supported GPIB Card
mconfig.exe	ICS 488-PC2 Card National Inst. GPIB-PC2a Card (Set to address 2E1 and to 7210 emulation) or any NEC 7210 compatible GPIB Controller Card that is set to 2E1.
niconf.exe	National Instruments AT-GPIB Card (Set to address 2C0H)
hpconf.exe	Hewlett-Packard HP-IB Card (Set to address DC000)

If you are using a PC with Windows 95 and have one of the following cards installed in your computer, you can download the niconfig program from our web site at http://www.icselect.com and use it to configure the unit. Niconf_z.exe is a self-exploding zip file that installs a configuration program that makes National Instrument type calls to control the GPIB bus.

Program	Supported GPIB Card
niconfig	ICS 488-PCI or 488-PCMCIA Any National Inst Card Any Computer Boards Card

Follow the instructions in Section 2.7 to configure the 4899/4809s from other computers.

2.6 USING THE CONFIGURATION PROGRAMS

The Minibox configuration programs walk the user through a menu driven program to configure the 4899/4809's power-on settings. The configuration programs on the Configuration Disk run on any IBM type PC or compatible clone with DOS 6.0 or Windows 3.1 or later versions of the operating system. A Windows 95 version of the configuration program can be downloaded from http://www.icselect.com.

2

2.6.1 Installing the DOS Configuration Program from the Disk

Perform the following steps to install the DOS configuration programs on your hard disk.

1. Turn on the computer and select the directory where you want the configuration program.

2. Load the configuration disk into the floppy disk drive.

3. Read the README file to see if there are any changes to the program that may affect the configuration procedure.

4. Copy the selected configuration program from the floppy disk to your selected directory. Use the mouse to drag the desired .exe program to the directory on the hard disk or use the DOS copy command. Substitute the correct floppy drive letter for the letter a in the copy command.

   ```
   >cd c:/newdir                 'go to new directory
   >copy a: mconfig.exe c:       'copy file
   ```

   When the installation is complete, remove the configuration disk from your floppy disk drive.

2

2.6.2 Installing a Downloaded Configuration Program

Perform the following steps to download and install the configuration program on your hard disk.

1. Use your internet browser to access the config.html page at http://www.icselect.com and to download niconfig.exe.

2. Place the file in a temporary directory and double click on it with the mouse to explode the file. Two of the exploded files will be a setup.exe and a readme file.

3. Follow the instructions in the included readme file to install the program in your computer.

2.6.3 Running the Configuration Program

The configuration programs support the standard configurable items. Special 4809 settings such as the user's IDN message will have to be entered with a live keyboard program or as part of the user's program. See section 2.7.

1. Connect the 4899/4809 to the GPIB controller card in the PC as shown in Figure 2-1. Plug the AC adapter plug into the DC jack on the 4899's rear panel. Connect the AC adapter to an AC outlet.

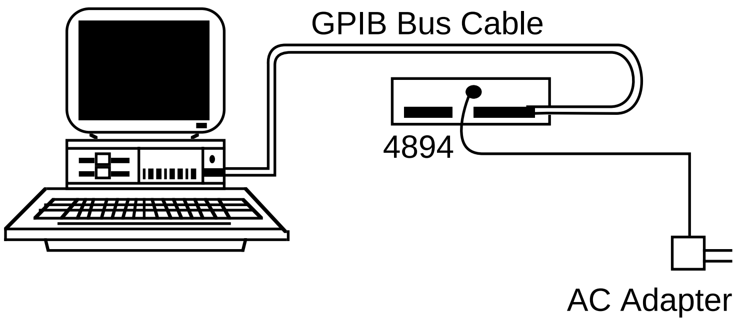


Figure 2-1 4899 Configuration Connections

2. Turn the 4899's power switch on or apply power to the 4809. After 1 second, the unit should blink its GPIB bus address on the LEDs. The selftest ends with the PWR and RDY LEDs both on and the other LEDs off.

3. Run the configuration program. This may be done by double clicking on the program name or by typing the program's name at the DOS prompt or in the Windows Run command box.

   > c:\new_directory\MCONFIG <return>

4. Product Selection

   The program will display a list of model numbers. Enter or select the number that corresponds to the model that you are configuring and press return

   e.g. 4899 <return> 'selects Model 4899

   The program will ask that you turn the unit off and back on. Press the Return key when the unit has finished its self test.

5. GPIB Address

   The program branches to the selected product menu and asks for the unit's current GPIB address. Enter a one or two digit value ; i.e., 4, 04, 10. The factory default setting is 4. If you do not know the unit's GPIB address, turn the unit off and back on. The unit will blink its GPIB address on the front panel LEDs at power turn-on. Add the bit weights to get the GPIB address.

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

2

6. Configuration Choices

The configuration program steps through each parameter and displays the current setting and configuration choices. The user should refer to the command definitions in Tables 3-2 and 3-3 to understand the command choices and their affect on the unit's operation. All setting changes are made by entering one of the displayed choices and pressing the Return key or clicking the Enter box in the Visual Basic versions. Pressing the Return key or clicking Enter without entering a new choice causes the program to skip to the next parameter leaving the current setting unchanged.

7. Saving the New Settings

After the last choice, the program will give you several configuration choices.

The program may give you the opportunity to set the SRE and ESE enable bit registers and to save the values so the unit can generate a SRQ at power turn-on. Enter **Y** to set PSC 0; **N** to set PSC 1 or click the appropriate boxes.

The program may ask if you want to lock the parameters so that they cannot be changed by the end user. The configuration program automatically unlocks the parameters whenever it is run. Enter **Y** to lock; **N** to continue or click the appropriate box.

The program will ask if you want to save the current configuration. Enter **Y** to save; **N** to continue or click the appropriate box.

8. Configuring other units

The program will ask if you want to configure another unit. Enter **Y** to configure another unit; **N** to exit.

2.7 CONFIGURING FROM OTHER CONTROLLERS

2.7.1 General Instructions

The 4899/4809 can be configured from any GPIB bus controller by using the following procedure. The following example commands are shown in HP BASIC for easily conversion to another language.

2

1. Connect the unit to the bus controller as shown in Figure 2-1 Use an Abort, REN or a take control type command to have the bus controller assert REN. Then turn the unit on.

2. Determine the unit's GPIB address:

   a) For new units use the factory setting of 04.

   b) For other units, turn the unit off and back on. At the end of its self test, the unit blinks its GPIB address on the front panel LEDs.

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

3. Verify communication to the unit by sending it the *IDN? query and reading back the unit's IDN message.

4. Use Table 3-2 to put together the SCPI command for the parameter you want to change. Use an OUTPUT, SEND or WRITE type statement in your GPIB Controller Card library to send the new configuration value to the unit. Follow each configuration statement with a query to verify that the unit accepted the new setting. The following example shows how to change and query the baud rate.

```
OUTPUT 704, "SYST:COMM:SER:BAUD  2400"
OUTPUT 704, "SYST:COMM:SER:BAUD?"
ENTER 704, B$
PRINT B$
```

2

5. Use caution when changing the unit's GPIB address. The change takes place immediately when the command is executed. Provide a 0.1 second delay before querying the new address setting.

   i.e., to change the GPIB address to 20

```
OUTPUT 704, "SYST:COMM:GPIB:ADDR 20"
WAIT 0.1
' address the unit by its new address
OUTPUT 720, "SYST:COMM:GPIB:ADDR?
ENTER 720, A$
PRINT A$
```

6. Use the ***SAV 0** command to save the new values in the unit's nonvolatile memory. The ***SAV 0** command also stores the current I/O configuration as the power-on values.

   e.g. OUTPUT 704, "*SAV 0", END 'HP BASIC command

2.8 GPIB CONNECTIONS

2.8.1 4899 GPIB Connections

The 4899 has a standard 24-pin IEEE-488 connector on its rear panel. The IEEE-488 connector mates with the standard IEEE 488/GPIB bus cables. Signal-pin assignments for the standard IEEE-488 connector are shown in Figure A-2 in the Appendix.

2

2.8.2 4809 GPIB Connections

The 4809 has two connectors that can be used to connect the 4809 to the GPIB bus. The 4809 only requires that one of the connectors be used to connect it to the GPIB bus. The unused connector can be left open.

2.8.2.1 GPIB Connector J1

Connector J1 is a 24-pin connector that is designed for direct connection to a GPIB bus connector using a flat ribbon cable. The GPIB Signal-pin assignments for J1 are identical to the standard IEEE-488 connector shown in Figure A-2 of the Appendix. The connector layout is shown in Figure 2-2 (a). Use a flat ribbon cable with a 24-pin plug on one end and a GPIB connector on the other end to mount the GPIB connector on the rear panel of the chassis. GPIB cables with metric lock studs are available from ICS as P/N 114439. Specify length in cm.

TABLE 2-1 4809 GPIB/ADDRESS CONNECTOR SIGNALS (J2)

Signal	Pin Number	Wire Color	Bit Weights
GROUND	1	BRN 1	
ADSW5	2	RED 1	16 (MSB)
T SW	3	ORG 1	not used
L SW	4	YEL 1	not used
ADSW4	5	GRN 1	8
SI SW	6	BLU 1	0
ADSW1	7	VIO 1	1
ADSW3	8	GRY 1	4
ADSW2	9	WHT 1	2
NRFD	10	BLK 1	GPIB Signal
REN	11	BRN2	
DAV	12	RED 2	
IFC	13	ORG 2	
NDAC	14	YEL 2	
EOI	15	GRN 2	
ATN	16	BLU 2	
SRQ	17	VIO 2	
DIO1	18	GRY 2	
DIO2	19	WHT 2	
DIO3	20	BLK 2	
DIO4	21	BRN 3	
DIO5	22	RED 3	
DIO6	23	ORG 3	
DIO7	24	YEL 3	
DIO8	25	GRN 3	
GROUND	26	BLU 3	GPIB Signal

1	13
2	14
3	15
4	16
5	17
6	18
7	19
8	20
9	21
10	22
11	23
12	24

(a) J1 GPIB Layout

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26

(b) J2 GPIB/Addr Sw Layout

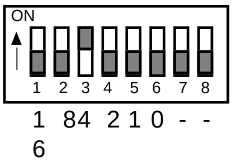
Note: Check Figure 1-3 for true connector orientation

Figure 2-2 4809 GPIB Connector Pin Layouts

2.8.2.2 GPIB/Address Switch Connector J2

Connector J2 is a 26-pin connector that contains the external address switch input signals as well as the GPIB bus signals. The connector layout is shown in Figure 2-2(b). Signal-pin assignments for J2 and flat-ribbon wire colors are listed in Table 2-1. The external address switch inputs are low true signals with pullup resistors on the 4809. At power turn-on, the 4809 reads the five address lines (ADSW1-ADSW5) if the external address switch is enabled.

Connector J2 mates with ICS's GPIB Connector/Address Switch Assemblies. These assemblies are small business card size assemblies that mount a GPIB connector and a 8-bit rocker switch to the rear panel of a chassis. They have a 26 conductor, flat ribbon cable that plugs into J2 on the 4809. The assemblies are available in two layout styles. Refer to Appendix A3 for dimensions and installation instructions and silkscreen. Switch rockers functions are shown in Figure 2.3.



Switch set to address 4

Figure 2-3 Address Switch Rocker Assignments

TABLE 2-2 SERIAL CONNECTOR PIN ASSIGNMENTS

Pin	RS-232	RS-422 RS-485	Signal	Direction In Out
1	AA	—	Chassis	
2	BA	SD(A)	Send Data (A)	→
3	BB	RD(A)	Receive Data (A)	←
4	CA	RS(A)	Request-to-Send (A)	→
5	CB	CS(A)	Clear-to-Send (A)	←
6			Data Set Ready	
7	AB		Ground	
8	CF	RR(A)	Signal Detected (A)	←
9				
10		RR(B)	Signal Detected (B)	←
11				
12				
13		CS(B)	Clear-to-send (B)	←
14		SD(B)	Send Data (B)	→
15				
16		RD(B)	Receive Data (B)	←
17				
18				
19		RS(B)	Request-to-send (B)	→
20	CD	TR(A)	Data Terminal Rdy (A)	→
21				
22				
23		TR(B)	Data Terminal Rdy (B)	→
24				
25				
26				

Note: Pin 26 is only available on the 4809

1	14
2	15
3	16
4	17
5	18
6	19
7	20
8	21
9	22
10	23
11	24
12	25
13	26

Note: Check Figure 1-3 for true connector orientation

Figure 2-4 4809 Serial Connector Pin Layout

2.9 SERIAL INTERFACE CONNECTIONS

The 4899 and 4809 provide RS-232 and RS-422/RS-485 signals on a single connector. Signal selection is done by setting two jumpers on the unit's PCB. Either the RS-232 or RS-422/RS-485 signals can be used to connect the unit to Modbus device(s). The RS-232 signals are recommended for short connections to a single Modbus device. The RS-485 signals are recommended for long cables or for connections to multiple Modbus devices.

2

2.9.1 Signal Assignments

The 4899's serial port is a DTE (Data Terminal Equipment) interface on a DB-25F female connector. The 4809 has a similar interface but uses a 26-pin male ribbon connector (Figure 2-4) that connects on a pin-to-pin basis to a 25-pin connector. Both connectors contain RS-232 and RS-485 signals in accordance with EIA-STD - RS-530. RS-232 and RS-485 signal selection is made by setting jumpers inside the 4899 or on the 4809's PC board. Refer to section 2.10 for jumper setting instructions. Table 2-2 shows the serial signal-pin assignments and the signal directions for both units.

2.9.2 RS-232 Connections to a Modbus Device

The 4899/4809's minimum RS-232 connection uses just three lines to connect the unit to a Modbus slave device. The lines are transmit data (SD), receive data (RD), and Ground. Figure 2-5 shows an RS-232 connection to a Watlow F4 Temperature Controller

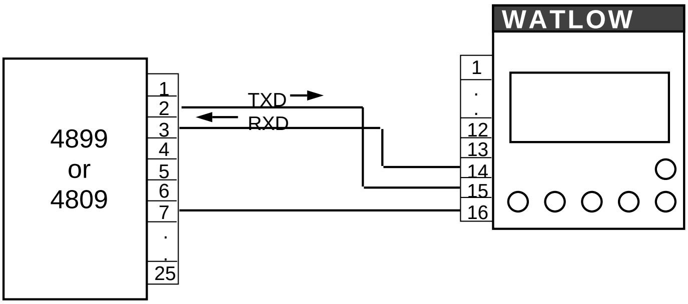


Figure 2-5 RS-232 Connections to a Modbus Device

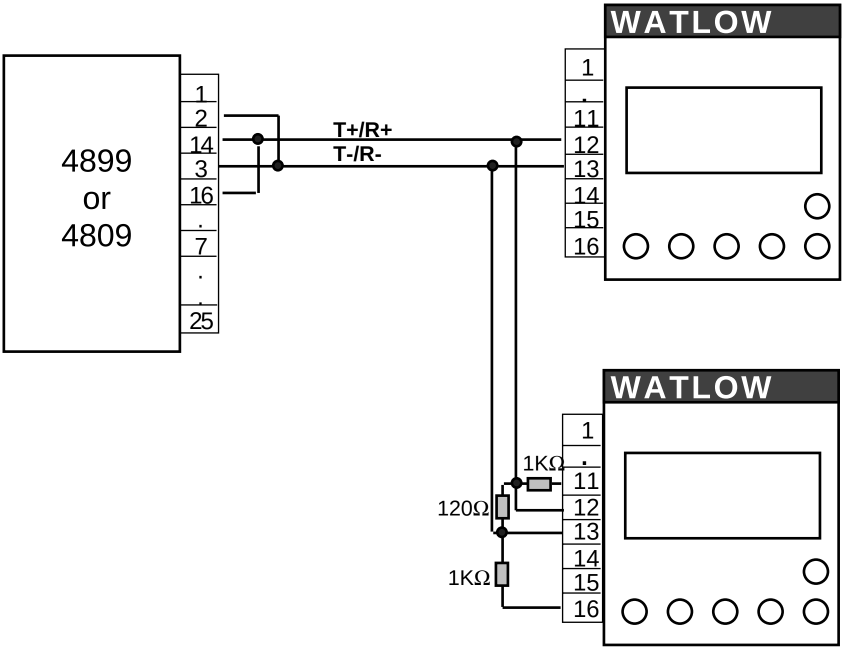


Figure 2-6 RS-485 Network Connections to Watlow to Watlow F4 Controllers

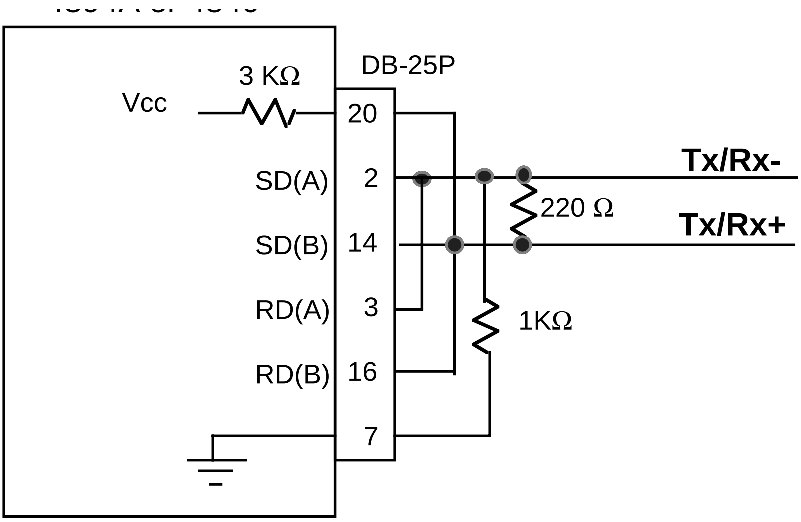


Figure 2-7 4899/4809 Pullup/Pulldown Resistor Connections

2.9.3 RS-485 Connections to a Modbus Device

The 4899 and 4809's serial interface provides a transmit (SD) and a receive (RD) pair of RS-422/RS-485 signals. Because most RS-485 networks use just two wires, the SD and RD signal pairs have to be jumpered together in the cable connector. The 4899 and 4809 have to be configured for RS-485 operation when used on a RS-485 network. Use the SYST:COMM:SER:RS485 ON command. This causes the unit to tristate its serial transmitter when not transmitting which free's the network so the Modbus can respond to the message.

Two wire RS-485 networks also need pullup, pulldown and load resistors to bias the lines in the 'mark' state when neither unit is transmitting. Set the bias voltages to 2 Vdc and 2.5 Vdc. Use resistors with an approximate value of 500 ohms/volt. Figure 2-6 shows an example of single 4899 or 4809 driving two Modbus Controllers over a RS-485 network. In Figure 2-6, the termination network uses 5 Vdc and ground provided on the Watlow controlllers.

Figure 2-7 shows an alternate method of making the RS-485 connection to a 4899 or 4809. Figure 7 uses the 4899/4809 ground and internal pullup resistor available at pin 20. Jumper the internal pullup resistor to the SD(B) and RD(B) lines. Use a 1 Kohm resistor from SD(A) and RD(A) to ground as the pulldown resistor. A 220 ohm load resistor completes the circuit.

2.9.4 4809 Serial Connection Methods

Here are some suggestions for connecting to the 4809's 26-pin serial connector. Method #1 uses a 26-wire flat-ribbon female connector crimped onto a open-end, rainbow colored flat-ribbon cable. Peel apart the open end and solder the desired lines to a connector that mates with the Modbus device. Method #2 uses an 114256-L cable from ICS which gives you a DB-25P male connector. L is cable length in cm. Method #3 uses individual wires with female clips crimped to wires that can slide down on the 4809's connector pins. The other end of the wires connect to the Modbus device. Use female pins similar to AMP part number 86016-5.

2

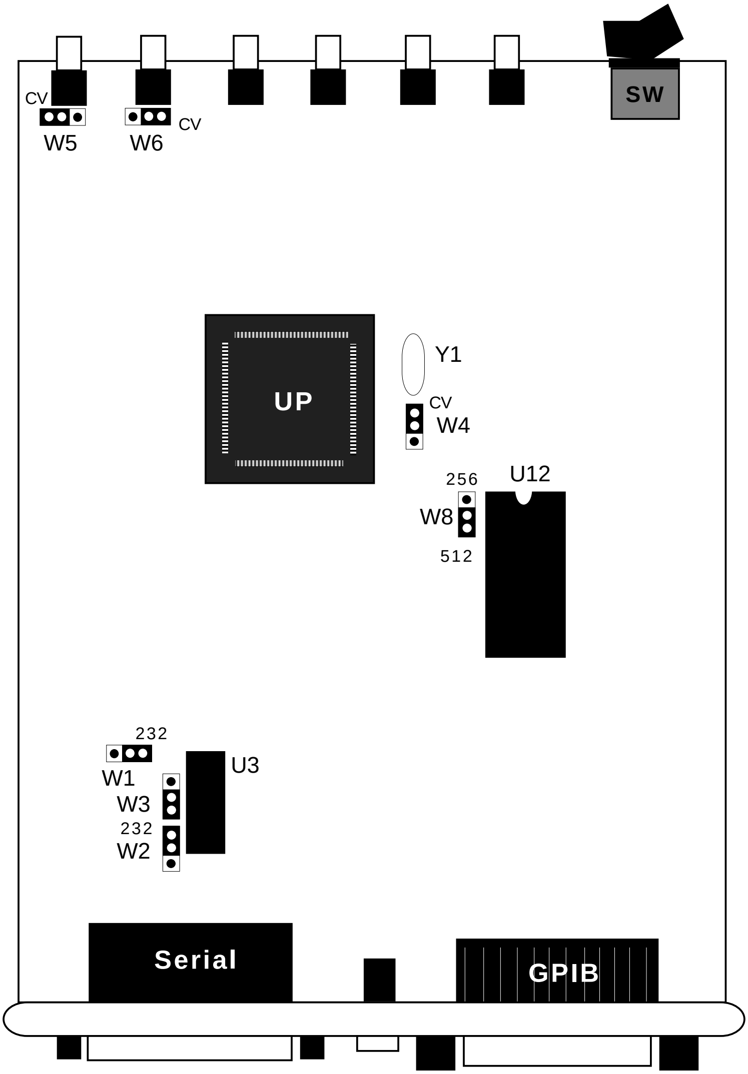


Figure 2-8 4899 Factory Jumper Setting for RS-232 Serial Signals

2.10 JUMPER SETTINGS

2.10.1 4899 Jumpers

The 4899 has seven jumpers on its PC board as shown in Figure 2-8. Jumpers W1, W2 and W3 are used to select RS-422 or RS-485 signals. Figure 2-8 shows these jumpers in the RS-232 position. Leave jumpers W4, W5, W6 and W8 in their factory set positions.

To change the jumpers, disconnect all cables from the unit. Undo the two rear panel screws and slide the unit out of its case. Locate and change the jumpers W1, W2 and W3 to the positions shown in Figure 2-9. Replace the unit in its case and tighten the two screws taking care not to over tighten them.

Reverse the above procedure to go from the RS-485 to the RS-232 signal setting.

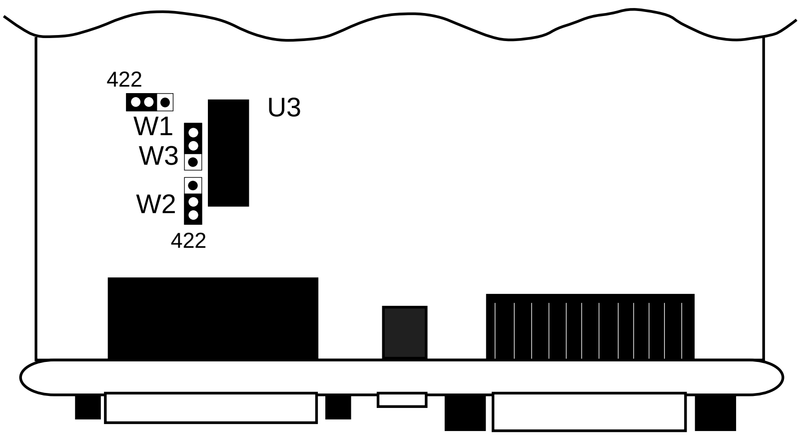


Figure 2-9 4899 Internal Jumpers set for RS-422/RS-485 Serial Signals

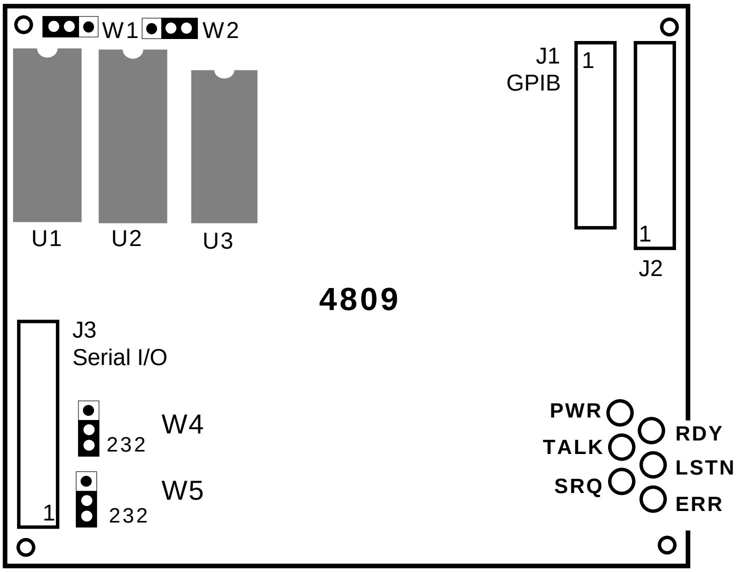


Figure 2-10 4809 W4 and W5 Jumpers set to RS-232 Position

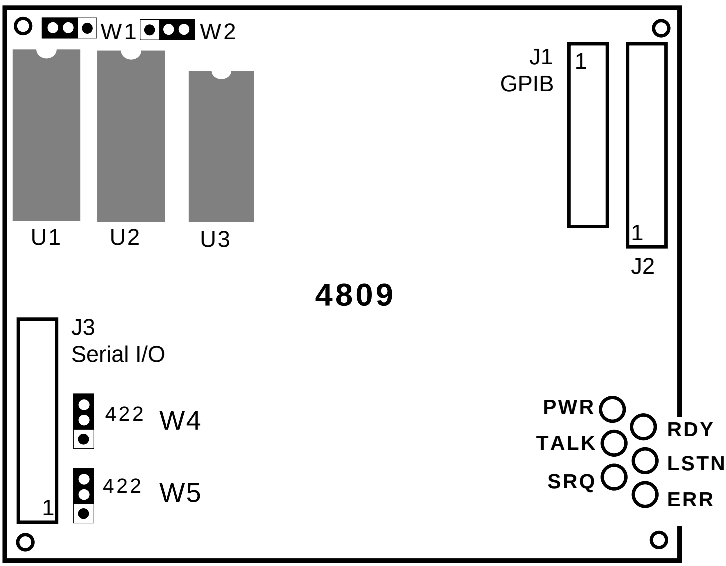


Figure 2-11 4809 W4 and W5 Jumpers set to RS-422 Position

2.10.2 4809 Jumpers

The 4809 has two jumpers to select the type of serial output signals. Jumpers W4 and W5 are factory set for RS-232 output signals as shown in Figure 2-10. For RS-485 signals, position both jumpers as shown in Figure 2-11.

2.11 4899 RACK MOUNTING INSTRUCTIONS

The Model 4899 is held in its rack mounting kit with a winged-'U' shaped bracket. Perform the following steps to install a 4899 in a rack mounting kit:

1. Hold the 4899 at a 30 degree nose down angle and place the front bezel through the rack mount kit from the rear of the kit. Push it forward through the opening until the rubber feet line up with the holes in the rack mounting kit. Push the unit down until it rests flat on the kit and the feet are in the four holes.

2. Repeat step 1 for a second unit if two units are being held in one rack mounting kit.

3. Aline the unit(s) so the bezels are parallel with the front of the rack mount kit and protrude equally through the front panel of the rack mounting kit.

4. Set the bracket so its two holes line up with the holes in the rack mounting kit extrusion. Use the supplied 4-40 screws to hold the bracket to the extrusion. Do not overtighten.

5. Use the supplied 10-32 screws to bolt the rack mounting kit into the rack.

Operation

3.1 INTRODUCTION

This section describes how the 4899/4809 operates and controls Modbus RTU controllers from the GPIB bus. This section also describes the SCPI commands used to configure and control the 4899/4809. Wherever the text refers to the Model 4899, it applies equally to the Model 4809 unless otherwise noted.

3.2 OPERATION

Commands received at the GPIB port are parsed and used to either control the 4899/4809 or are used to create messages that are transmitted serially to the Modbus slave devices. Both the 4899 and the 4809 have a SCPI parser that accepts IEEE-488.2 commands, SCPI commands and Modbus commands. The IEEE-488.2 and SCPI commands are used to setup and configure the 4899 or 4809's Status Reporting Structure or its GPIB and Serial interfaces. Any commands that end in a '?' are a query and the unit responds by outputting the response on the GPIB bus the next time it is addressed as a talker.

Modbus commands are used to set the Modbus device address, to write data to the device's registers or to query the contents of the registers. When the 4899/4809 receives a Modbus read or write command, it assembles a RTU style message that contains the slave Modbus device address, register number and the data to be sent to the

Modbus slave device. The 4899/4809 appends a CRC checksum to the message and transmits it serially to the Modbus device.

If the message is successfully received by the Modbus device, the Modbus device will generate a response message that either confirms receipt of the message or that contains the requested data. The 4899/4809 receives the response message and compares its checksum against the message's CRC value. If the response message is a valid response message to a read command, the returned data is held in the GPIB transmit buffer and will be output on the GPIB bus the next time the unit is addressed to talk. If the message is an acknowledgment message, there is no further action.

3

The 4899/4809 expects to receive a response from the Modbus device within a preset time period or it declares a timeout error. The timeout period is programmable and is factory set to 100 milliseconds.

If the message was not a valid message, or was an exception message, or was missing, then the 4899/4809 sets the appropriate bit(s) in the Questionable Register and puts a decimal value in the Modbus Error register. Both registers are part of the unit's Status Reporting Structure. If the corresponding enable bits were true, then the 4899/4809 will generate a Service Request by asserting the SRQ line. The SRQ line stays asserted until the unit is serial polled or until the bits that caused the SRQ are reset.

3.3 ADDRESSING THE UNITS

3.3.1 4899 and 4809 Internal GPIB Address

The 4899 and 4809s can be set to any unused GPIB primary address between 0 and 30. The Bus Controller will use the primary address to address the unit as a talker or as a listener. Bus addresses of 0 and 21 are not recommend as these addresses are customarily used by Bus Controllers as their own address.

The internal GPIB address can be set or queried with the SCPI SYST:COMM:GPIB:ADDR command. The change takes affect when the command is executed so any subsequent commands will need to address the unit at its new address. Use the IEEE-488.2 common command *SAV 0 to save the new address value in the unit's nonvolatile memory.

If you have forgotten the GPIB address, momentarily turn the unit off and back on. At the end of the self test, the unit will blink its GPIB address on the front panel LEDs using the following bit weights:

PWR	RDY	TALK	LSTN	SRQ	FULL
-	16	8	4	2	1

3

3.3.2 4809 Address External Address Switch

In addition to the internal GPIB bus address method described above, the 4809's GPIB address can also be set by connecting the 4809 to an external address switch. Connector J2 contains the external address switch input lines in addition to the GPIB bus signals. J2 mates to ICS's GPIB Connector/Address Switch Assemblies. These assemblies are small business card size assemblies that make it easy to mount a GPIB connector and an address switch on the rear panel of the host chassis.

The 4809's external address switch is enabled with the SCPI SYST:COMM:GPIB:ADDR:EXT ON command. When enabled, the 4809 reads the external address switch at power turn-on. If the address switch is changed, the unit must be powered off and back on before it will respond to the new address. When the external address switch is enabled, the internal address value is ignored.

3.4 488.2 STATUS REPORTING STRUCTURE

The 4899/4809 includes the expanded IEEE-488.2 status reporting structure shown in Figure 3-1. The expanded status reporting structure conforms to the SCPI 1994.0 Specification and builds on the IEEE 488.2 Standard status structure with the addition of the Questionable, Operation and Modbus Error registers. The Event and Status registers are controlled and queried with the IEEE-488.2 common commands. The Status Byte Register may also be read by serial polling the 4899/4809. The added Questionable and Operation registers are controlled and queried with SCPI commands. The Modbus Error register is read and cleared with the Modbus E? command.

As shown in Figure 3-1, IEEE 488.2 SRQ generation is a multilevel function and is determined by the occurrence of an event that has its corresponding enable bit set to '1'. The register outputs are summarized in the Status Byte Register which generates the Service Request and pulls the SRQ line low. SRQs are used to signal the bus controller that an event has occurred and/or that the 4899/4809 needs service. There are four major sources of SRQs, each of which is summarized in a bit in the Status Byte Register. Three of the sources are event registers with their own enabling bits and the fourth is the Output Queue. The Event registers and the Output Queue are cleared when read or by the *CLS command.

3.4.1 Event Registers

An event register **captures 0 to 1 transitions** in its associated condition register or in the standard event register. An event bit becomes TRUE (1) when the associated condition bit makes <u>logical 0 to 1 transition</u>. Once an event bit is set it **is held** until the event register is read or cleared with the ***CLS** command.

Each event register contains eight or sixteen bits. When the register is read, its response is a decimal number that is the sum of the binary bit weights of the bits that are logical 1s.

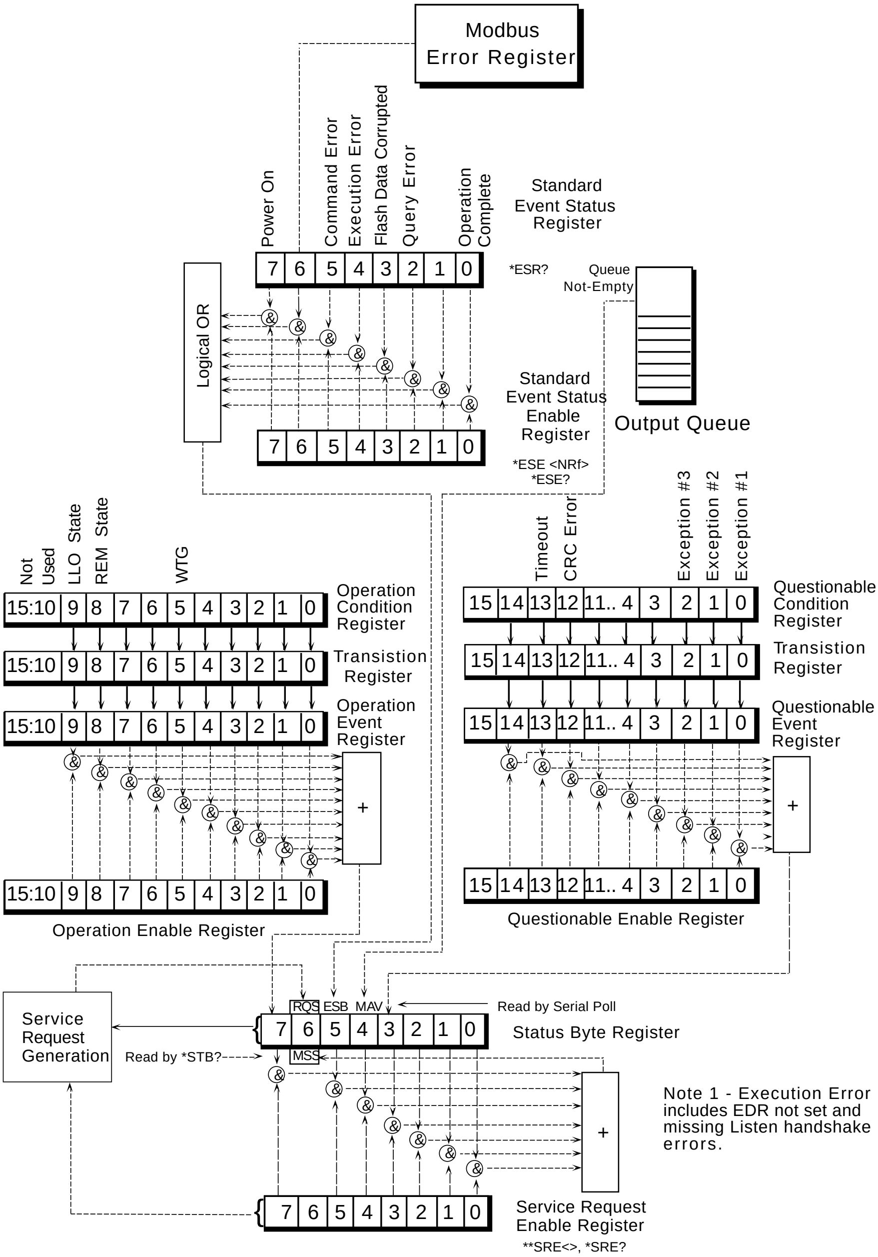


Figure 3-1 4899/4809 Status Reporting Structure

e.g., 23 decimal = 0001 0111 or 0000 0000 0001 0111 binary

Each event register bit has a corresponding enable bit. The enabling bits are ANDed with the state of the event bits to create the summary condition in the Status Byte Register. Unwanted conditions can be blocked from generating SRQs by setting their corresponding enabling bit to a '0'. The enabling bits are set by writing the value equal to the sum of all of the desired logic 1 bits to the enabling register. The value is normally decimal but can be expressed in HEX, OCTAL or BINARY by prefixing the number with a #H, #O or #B.

3.4.2 Event Status Register

3

The Event Status Register reports events that are common to all 488.2 devices. This includes events such as self test errors, command errors, execution errors, power on and operation complete. The Power-on event occurs at power turn-on and can be used to signal a power off-on occurrence. In the 4899/4809, the Modbus Error Register is summarized into the Event Status Register as Bit 6. The 488.2 Operation Complete event has no meaning for either units.

The Event Status Register is read with the ***ESR?** query. Use the ***ESE** commands to set the Event Status Enable Register as shown in the following example:

***ESE 60**	'enables error bits 2 through 5 for errors
***ESE 124**	'enables error bits 2 through 5
***ESE?**	'queries the enabling register setting

3.4.3 Modbus Error Register

The Modbus Error Register reports a decimal value of the last error detected with the Modbus message transmission or reported back from the Modbus slave device. This register is cleared when read by the Modbus E? command. The *CLS and *RST commands have no affect on this register. Refer to Table 3-4 for the Modbus Error Register values.

3.4.4 Questionable Registers and Digital Inputs

The Questionable Registers lets the user read bits that report CRC errors, Exception message types or a timeout (no response message received). Bit alignments are shown in Figure 3-2. The Questionable Transition Register filters the inputs and passes only the enabled state changes to the Questionable Event Register. The Questionable Event Register bits becomes true (1) when the positive transition bit is enabled and the associated condition register bit makes a 0 to 1 transition. When both transitions are selected for the same bit, the corresponding Questionable Event Register bit sets whenever the digital input changes state. The Questionable Event Register is cleared when it is read.

The Questionable Registers are queried with the SCPI STATUS branch commands.

The 4899/4809 can be set to monitor the bits in the Questionable Register and generate a SRQ when they change state. The following example sets the Questionable Event register to monitor the CRC and Timeout bits by capturing a positive transition on bits 12 and 13. The decimal value for bit 12 is 4096 and the decimal value for bit 13 is 8192.

```
STAT:QUES:PTR 12298      'enables bits 12 and 13 to set on
                          a positive transition
```

Because summing large decimal values is confusing, it is better to use HEX values that are easier to write. i.e.

```
STAT:QUES:PTR #h3000    'same as 12298 decimal
```

The Questionable Enable Register enables set Event bits to be included in the summary output to the Status Byte Register. The following example enables bits 12 and 13:

```
STAT:QUES:ENAB #h3000  'enables Event bits 12 and 13
```

Note that the Questionable Event Register has to be cleared after an SRQ is generated either by reading the register or with the *CLS command. If the register is not cleared, the event bits will remain set and they will not generate another SRQ when the input again goes true.

STAT:QUES:COND?	'reads the questionable inputs

3.4.5 Operation Registers

The 488.2 Operation Registers lets the user read device specific status conditions and detect any changes in the device's status. The Operation Registers are similar to the Questionable Registers described in paragraph 3.4.3.

3

In the 4899/4809, the Operation Condition Register reports the WTG (Waiting for Trigger) status and the Local Lockout and Remote GPIB interface states. The WTG bit is true when the 4899/4809 has been armed and is waiting for a trigger. The following commands demonstrate some possibilities of the Operation Registers:

STAT:OPER:PTR 32	'enables bit 5 to set on a positive transition of WTG
STAT:OPER:ENAB 32	'enables Event bit 5
STAT:OPER:COND?	'querys the Operation Condition Register

3.4.6 Output Queue

The Output Queue is used by the 4899/4809 to send IEEE 488.2 messages back to the bus controller. These messages are responses to 488.2 and SCPI queries sent to the unit by the bus controller. The Output Queue reports a '1' in bit 4 of the Status Byte Register when it contains a message(s) to be read by the bus controller. Reading the contents of the Output Queue clears its summary bit. The Output Queue is read by addressing the 4899/4809 to talk at its GPIB address. If the Output Queue is not read before sending another query, its contents will be lost and an error reported.

3.4.7 Status Byte Register

The 4899/4809 generates a service request (SRQ) whenever any of the enabled bits in the Status Byte Register become true and the 4899/4809 is not addressed as a talker. The Status Byte Register may be read by a Serial Poll or with the ***STB?** query. A Serial Poll resets the RQS bit; the ***STB?** query does not change the bit. The Status Byte Register is enabled by setting the corresponding bits in the Service Request Enable Register with the ***SRE** command. e.g.

***SRE 160**	'Sets the SRE Register to 1010 0000 which enables just the Event Status and Questionable summary bits to generate SRQs.

3.4.8 Saving the Enable and Transition Register Values

The Enable and Transition Register values can only be saved and recalled at power turn-on by disabling the PSC flag. **The *SAV command does not save the Enable and Transition register values.** Use the *PSC 0 command to disable the PSC flag and save the current Enable and Transition register values. The following example saves the current settings which enables bits in the Operation and Event Status Registers to generate a SRQ at power turn-on. e.g.

STAT:OPER:ENAB 1	'enables bit 1
STAT:OPER:NTR 1	'enables neg transition
ESE 192; SRE 32;*PSC 0	'saves Power-on and EDR bits and current registers values as power on settings.

Note that the enable and transition commands must be on the same line or set prior to the *PSC 0 command to be saved. A later *PSC 1 command sets the PSC flag which will cause the registers to be cleared at the next power turn-on.

3

3.4.8 488.2 Differences from 488.1 Devices

The IEEE 488.1 Device Clear command **does not** reset the unit's input-output settings as would be expected of a 488.1 device. To reset the unit's input-output settings, use the *RST (Reset) or *RCL 0 command.

3.5 488.2 CONFORMANCE INFORMATION

3

The IEEE 488.2 Standard mandated a list of common commands that are common to all IEEE 488.2 compatible devices. The 4899 and the 4809 respond to all of the mandated common commands and to some optional commands defined in IEEE-488.2. Table 3-1 lists the IEEE-488.2 commands that apply to this unit, and describes the affect they have on the 4899/4809 and its status reporting structure.

TABLE 3-1 IEEE-488.2 COMMON COMMANDS

COMMAND	NAME	DESCRIPTION
*** CLS**	Clear Status	Clears all event registers summarized in the status byte, except for "Message Available," which is cleared only if *CLS is the first message in the command line.
***ESE <value>**	Event Status Enable	Sets "Event Status Enable Register" to <value>, an integer between 0 and 255. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated. EXAMPLE: decimal 16 converts to binary 00010000. Sets bit 4 (EXE) in ESE to 1.
***ESE?**	Event Status Enable Query	4899 returns the <value> of the "Event Status Enable Register" set by the *ESE command. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***ESR?**	Event Status Register Query	4899 returns the <value> of the "Event Status Register" and then clears it. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***IDN?**	Identification Query	4899 returns its identification code as four fields separated by commas. These fields are: manufacturer, model, six-digit serial number and version of firmware - e.g. **ICS Electronics, 4899, S/N 012123, Rev. 00.10 Ver 99.01.27**
***OPC**	Operation Complete Command	Causes the 4899 to generate the operation complete message in the Standard Event Status Register when all pending selected 4899 operations have been finished.
***OPC?**	Operation Complete Query	Places an ASCII character 1 into the 4899's Output Queue when all pending selected 4899 operations have been finished.

TABLE 3-1 IEEE-488.2 COMMON COMMANDS (CONTINUED)

COMMAND	NAME	DESCRIPTION
***RCL 0**	Recall	Restores the state of 4899 from a copy stored in its memory by *SAV command.
***RST**	Reset	The 4899 restores its power-up state except that the state of IEEE-488 interface is unchanged, including: 1) instrument address, 2) Status Byte and, 3) Event Status Register.
***SAV 0**	Save	Saves current 4899 configuration in non-volatile memory. *SAV 0 saves the current setting as the new power on setting.
***SRE <value>**	Service Request Enable	Sets the "Service Request Enable Register" to <value>, an integer between 0 and 255. The value of bit six is ignored because it is not used by the Service Request Enable Register. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated.
***SRE?**	Service Request Enable Query	4899 returns the <value> of the "Service Request Enable Register" (with bit six set to zero). <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***STB?**	Read Status Byte	4899 returns the <value> of the "Status Byte" with bit six as the "Master Summary" bit. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***TST?**	Self-Test Query	Returns status of the last power turn-on self test. A zero response indicates no test failures. Other responses are listed in Table 5-2.
***WAI**	Wait-to-continue	Prevents the 4899 from executing any further commands or queries until the No-Operation-Pending flag is TRUE.

3.6 SCPI CONFORMANCE INFORMATION

The 4899/4809 accepts SCPI commands and command extensions to configure its GPIB/Serial interfaces, to set the data formats and to transfer data. The SCPI commands conform to SCPI Standard 1995.0 and provide an industry standard, self-documenting form of code that makes it easy for the programmer to maintain the application program.

Table 3-2 shows the 4899/4809's SCPI command tree. The command tree uses portions of the SCPI SYSTEM, STATUS, FORMAT, INITIATE, ABORT and CALIBRATE subsystems. The 4899 and 4809 follow SCPI's hierarchal 'tree like' structure which starts with a root keyword and branches out to the final action keyword. Each command can be used as a query except where noted. The SCPI commands are **not** case sensitive. The portion of the command shown in capitals denotes the abbreviated form of the keyword. Either the abbreviated or whole keyword may be used when entering a complete command. Bracketed keywords are optional and may be omitted. There must be a space between the command and the parameter or channel list.

3

e.g., **STATus:QUEStionable?** is the same as
STAT:QUES:EVEN? or
stat:ques?

Table 3-3 lists the SCPI keywords and describes their functions in detail. Keywords other than those listed in the table or locked keywords will have no effect on the 4809's operation and a command error will be reported. Refer to Appendix A-1 for additional information about SCPI commands.

Note: A SCPI command that ends with a question mark '?' is a query. All queries should be followed by reading their response to avoid data loss.

TABLE 3-2 SCPI COMMAND TREE

Keyword	Parameter Form	Notes & Short Form Commands
SYSTem		**System Address**
:COMMunicate		
:GPIB		
:ADDRess	0 - 30 [4]	
:EXTernal	0|1 or OFF|ON [0]	4809 only
:SERial		
:BAUD	<numeric value> [9600]	
:PARity	EVEN | ODD | [NONE]	
:BITS	7 | [8]	
:SBITs	[1] | 2	
:UPdate	no value-command only	
:RS485	0|1 or OFF|ON [0]	
:ERRor?	(0, "No error")	
:VERSion?	(1994.0)	
STATus		
:OPERation		Status Inputs, WTG
[:EVENt]?	bit 0,1 and 5 active (0)	
:CONDition?	bit 0,1 and 5 active (0)	
:ENABle	bit 0,1 and 5 active (0)	
:ENABLE?		
:PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:QUEStionable		Modbus Error Bits
[:EVENt]?	bits 0-2, 12, 13 active (0)	
:CONDition?	bits 0-2, 12, 13 active (0)	
:ENABle	bits 0-2, 12, 13 active (0)	
:ENABLE?		
: PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:PRESet		

TABLE 3-2 SCPI COMMAND TREE (CONT'D)

Keyword	Parameter Form	Notes & Short Form Commands
FORMat		**Format Strings**
[:DATA]		
:TALK	ASCii | HEXL	
	[ASCII]	FT
INITate		**Trigger**
[:IMMeditate]		TI
:CONTinuous	1(On)| 0(Off) [0]	TC
ABORt		TA
CALibrate		**Calibrate**
:IDN	string	
:DATe	mm/dd/yy	
:DEFault		
:LOCK	1(On)| 0(Off) [0]	

Notes:

1. Parameter enclosed by [ ] - denotes factory default
2. Parameter enclosed by ( ) - denotes power on default
3. SCPI name ends with ? - denotes query only
4. Unless otherwise noted SCPI command is also a query
5. Keyword enclosed by [ ] - denotes optional use
6. Only a configuration command that has one of its parameters enclosed by [ ] can change its parameter setting and have this setting stored in the 4809's E^2ROM (with the *SAV command).
7. The format for a SCPI list is (@1,2, n) or (@ 1:n). There must be a space between the @ and the first number and parenthesis are required. A list of numbers is separated by commas or uses a colon to denote a range of numbers.
8. Numeric entries conform to IEEE-488.2 section 7.7.2.4 for decimal numeric parameters.
9. ASCII formatted data is a series of decimal values (0-255) for each byte separated by commas. e.g. 64, 132, 8
10. The CAL:DATe commands stores the CAL:IDN and CAL:DATe parameters in the 4809's E^2ROM.
11. The CAL:DEFault command resets the E^2ROM memory to it factory settings. Caution - All user settings will be overridden by this command.
12. Most parameters can be output in various numeric formats (radix). The parameters with decimal 0-255 value ranges may also be output as HEX using #h00-#hFF or Binary using #b00000000-#b11111111. Conversely, the parameters shown with HEX (#h) values can also be output in Decimal.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES

3

Keyword	Default Value	Description
SYSTem	-	Starts System command branch.
:COMMunicate	-	Identifies communication subsystem commands
:GPIB	-	Controls GPIB (IEEE 488) port settings
ADDRess(1)	04	Sets GPIB primary address. Values = 0 to 30 for Single mode, 0-29 for Dual mode and 1 to 30 for Secondary addressing. **Provide 70 ms delay after an address change before next command** Note - The GPIB address and the serial address are the same parameter. Changing either one, changes both settings.
:ADDRess?		Returns 0 - 30 for 4809's primary address.
:EXTernal	OFF	On enables the 4809's address switch inputs to be used to input a switch setting for the GPIB address. Off uses the value saved in E^2ROM memory. Does not apply to the 4899. Values = 0|1 or OFF|ON.
:SERial		Controls Serial Interface settings
:BAUD	9600	Sets serial baud rate. Vallues for the 4809 are 300 to 38400 baud. Values for the 4899 are 300 to 115200 baud.
:PARity	NONE	Sets serial parity. Values = EVEN, ODD or NONE.
:BITS	8	Sets number of data bits per character. Values= 7 | 8.
:SBITs	1	Sets minimum number of stop bits between characters. Value = 1 | 2.
:UPDATe	-	Sets UART with new serial values. User must reprogram the serial controller's COM port after this command.

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
RS485	OFF	Tristates transmitter when not transmitting for two wire networks. Values are ON and OFF.
:ERRor?	0, "No error"	Requests next entry in 4809's error/event queue. Error messages are: 0, "no error" -100, "Command error" -200, "Execution error" -400, "Query error"
:VERSion?	1994.0	4809 returns the <value> of the applicable SCPI version number.
STATus	-	Starts Status Reporting Structure
:OPERational	-	Identifies Operational registers.
:QUEStionable	-	Identifies Questionable registers.
[:EVENt?]		Returns contents of the event register associated with the command.
:CONDition?		Returns contents of the condition register associated with the command.
:ENABle	0	Sets the enable mask which allows the true conditions in the associated event register to be reported in the summary bit.
:PTRansition	#h7FFF	Sets positive transition enable register. Value = 0 to #h7FFF in decimal or HEX.
:NTRansition	0	Sets the negative Transition register. Values = 0 to #h7FFF in decimal or HEX.
:PREset		Sets the selected Enable Register, PTR and NTR registers to their default values (0, #h7FFF and 0 respectively) so the 4809 detects a positive changes

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

3

Keyword	Default Value	Description
FORMat		Starts string format branch.
DATA		Optional digital data identifier
TALK	ASCii	Sets talk string and data query response format. ASCII expresses a words input bit pattern as a decimal value equal to the binary sum of the data. Multiple words are separated by commas. HEXL converts each four bit nibble into the ASCII characters 0-9 and A-F. TABLE allows the user to define his own character set. All talk strings end with a linefeed. Values are ASCii | HEXL |. i.e. ASCii example = 128,5,255 HEXL example = 8000, 05FF
INITiate		Starts Trigger branch
[:IMMediate]		Enables a single trigger operation
:CONTinuous	OFF	Enables ongoing external triggers. Values = 0|1 or OFF|ON.
ABORt		Disables trigger function
CALibrate		Starts calibrate branch
:IDN <string>		Sets user IDN message. String is up to 72 characters and consists of four fields (manufacturer, model code, serial number and firmware revision) separated by commas. e.g. **ICS Electronics, 4809, S/N 990102, Rev 00.10, Ver 99.01.27.**
:DATe <date>		Saves IDN message and date. The save operation lights all the LEDs. Date is in mm/dd/yyyy format. A *CLS will clear the ERR LED after a CAL:DATE command.

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:DATe?		Queries the calibration date. The response is 00/00/0000 for factory default settings.
DEFault		Sets E^2ROM memory to factory settings.
:LOCK	0	Disables configuration commands when On. Values = 0|1 or OFF|ON. Table 1-4 lists the locked commands.

3.7 Modbus Commands

The following commands are used to Control Modbus slave devices.

TABLE 3-4 MODBUS COMMANDS

Syntax	Default	Meaning
C addr	1	Modbus Address Command. Sets Modbus slave device address for subsequent commands. Value for *addr* is 1 to 255.
L[?] w	-	Loopback Command. Writes a 16-bit word, *w*, out to a Modbus device and returns a single response word to the GPIB bus. The question mark is optional. Value for *w* is 0 to 65535.
R[?] reg, num	-	Read Register Command. Reads one or multiple Modbus device registers. User specifies starting register *reg* and number of registers to be read *num*. The [?] is an optional symbol so programs like ICS's GPIBKybd program can recognize the comand as a query and automatically read the response. Values for *reg* are 0 to 32767. Values for *num* are 1 to 64. Responses are returned as 16-bit decimal or HEX values separated by commas. Output format selected with the Format command. i.e.. R? 0,1 reads Watlow Model Number. Response is 5270 for Watlow Model F4 R? 0,3 reads three successive registers. Response is 5270,0,123 for the Watlow F4 Controller.
W reg, w	-	Write Register Command. Writes a 16-bit value, *w* to a single Modbus device register, *reg*. Values for reg are 0 to 32767. Values for *w* are 0 to 65535. An example is: W 100, 55 writes the decimal value 55 to register 100.

3

TABLE 3-4 MODBUS COMMANDS (CONTINUED)

Syntax	Default	Meaning
WB reg, num, w(0)..w(n)		Write Block Command. Writes multiple 16-bit words, *w(i)* to multiple registers. Starting register, *reg*.. Number, *num* specifies how many words are to be written. Values for *reg* are 0 to 32767. Values for *num* are 1 to 64. Values for *w* are 0 to 65535.
D time	100	Timeout Command. Sets timeout value of Modbus response message in milliseconds. Timeout is the total time for the message to be received by the 4899 or 4809. Value for *time* is 1 to 65,535 milliseconds. Default is 100.
D?		Queries the current timeout setting.
E?	-	Read Error Command. Reads and clears the Modbus Error Register and bit 6 in the Event Status Register. Returns a error code whose value is 0 to 255. Current error values are: 0 No errors present 1 Exception Code 1 2 Exception Code 2 3 Exception Code 3 100 CRC Error 101 Timeout Error indicates no characters received in response message. 2nn Partial or corrupted message re ceived. nn is the number of received bytes.

Notes:
1. All values are in decimal. To enter HEX values, the value must be preceded with a #h . i.e. 100 decimal = #h64
2. Response parameter format set by SCPI FORMat command. Default is ASCii

3

3.8 PROGRAMMING GUIDELINES

The following section provides information on how to program the 4899 and 4809 to set its configuration and how to send commands to the Modbus slave device.

3.8.1 General Configuration Guidelines

New units are factory set so that they are ready to be used when received. Table 1-1 lists the Factory Configuration. To change the configuration, the user should follow the sequence outlined below:

3

Send IFC	'gets control, asserts REN
Send SCPI commands and queries	'read and change settings
Send*ESR? **Read response**	'query the Event Status Register to be sure that an error did not occur.
Send "*SAV 0"	'save the new setup

The *SAV 0 command will cause the 4899/4809 to blink all but one of its LEDs. Pay close attention to the ERR LED when sending commands. If it comes on, the unit's parser detected a problem with the command and DID NOT execute it.

3.8.2 Setting the Modbus Device Address

The first step is to set the Modbus device address so the 4899/4809 can send messages to the correct Modbus device. If the 4899/4809 is being used with only one Modbus device, the address can be set and saved as part of the unit's power on configuration.

C 1	'sets device address
***SAV 0**	'optional save if controlling only one Modbus device

3.8.3 Querying a Modbus Device

The second step is to send the query to the 4899/4809 and read back the response from the Modbus device. The following commands set device #1 and read 1 value from register 0. With Watlow products, register 0 is the Watlow Model number register. The '?' is optional and is included so programs like ICS's GPIB Keyboard control programs can automatically read back and display the response from a query. i.e.

```
C 1             'sets Modbus Address to 1
R? 0,1          'reads Watlow model number
```

3

A more realistic command might be to read the measured temperature from register 100. (Register numbers will vary with the Modbus device)

```
C 7             'sets Modbus Address to 7
R? 100,1        'reads temperature
```

The device address and read command can be concatenated on one line by separating them with semicolons. i.e.

```
C 7: R? 100,1   'concatenated command
```

3.8.4 Writing to the Modbus Device

The nature of the command depends upon the specific Modbus device. Simple writes are handled with the W command. In the following example, a value of 50 is written to register 150. i.e.

```
W 150, 50       'sets temperature setpoint

C7; W 150, 50   'concatenated command
```

Writes to multiple registers are possible with the WB command.

3.8.5 Setting Modbus Device Timeouts

The time that the 4899/4809 waits to receive a response from the Modbus device is programmable by the D command. If the 4899/4809 does not receive a response within the timeout period, it assumes that the Modbus device is not responding and sets the timeout error. The factory default for the 4899/4809 timeout is 100 milliseconds. The command to change the timeout period is:

D 300 'sets timeout period to 300 ms

3

The default time period has proved to be satisfactory for some Watlow controllers but should be checked carefully for your specific Modbus device. Some devices fail to respond within the default time period because they perform periodic calibrations. The recommendation is that your program should have a built-in recovery routine to handle the occasional timeout error.

3.8.6 Locking Setup Parameters

All of the 4809/4899's configuration parameters can be locked to prevent accidental change by the end user. These lockable parameters are noted by a # symbol in Tables 1-1. Locked parameters cannot be queried or changed while locked. Any command that addresses a locked parameter is not executed, the Command Error bit in the Event Status Register is asserted and the ERR LED is lit. The lock function is saved by the *SAV 0 command. An example is:

CAL:LOCK ON 'blocks unauthorized changes
***SAV 0** 'saves lock condition

CAL:LOCK OFF 'unlocks the setup parameters

While lock is enabled, the end-user can change and save any non-locked parameter.

3.8.7 Generating SRQs from Modbus Errors

Figure 3-1 shows the 4899/4809's Status Reporting Structure. All Modbus Error codes are placed in the Modbus Error Register at the top of the figure. If the proper Event Status and Status Byte register bits are enabled, any Modbus Error code will generate a SRQ. The commands to enable the bits are:

***ESE 64**	'enables ESR bit 6
***SRE 32**	'enables Status Byte bit 5

Some Modbus Errors set specific bits in the Questionable Event Register. To generate a SRQ from a specific event, its bit must be enabled. The following commands enable SRQs for Timeouts and CRC errors only:

STAT:QUES:PTR #h3000	'enables positive going bits 12 and 13 to set bits in the Questionable Event Register
STAT:QUES:ENAB #h3000	'enables Event bits 12 and 13
***SRE 8**	'enables Status Byte bit 3

In both cases, the user needs to reset the event cause and clear the SRQ so another error will cause another SRQ. In case one, this is done by reading the Modbus Error Register with the E? query. In case two, the Questionable Event Register must be read to clear the set event bits.

3.8.8 Personalizing the Unit's IDN Message

The 4899/4809's IDN message is changed with the CALIBRATE subsystem commands. Change the IDN message when you want to personalize the unit, to identify the overall assembly as being from your company or to record product history or revision dates. The IDN message is a lockable parameter and if locked, needs to be unlocked before being changed. The format for the IEEE 488.2 IDN message is four fields (company, model#, serial number and revision) separated by commas and a maximum of 72 characters long. The word "model" may not be used in an IEEE-488.2 IDN message. An example IDN message change sequence is:

3

CAL:LOCK OFF	'unlocks all parameters
CAL:IDN Acme Mfgr Co, 101, s/n 007, Rev 10 0/08/99	
CAL:DATe 01-15-1999	'saves new IDN message 'Note-use the current date
CAL:LOCK ON	'relocks all parameters
***SAV 0**	'saves lock status

3.8.9 Saving the Configuration

The 4809 uses the 488.2 *SAV 0 command to save the current configuration in nonvolatile Memory. This includes all configuration settings and the current I/O settings. The saved configuration is recalled and the I/O settings restored to their saved state at power turn-on or by the *RCL 0 command. **WARNING - Because the Nonvolatile Memory has a finite number of write cycles, the *SAV command should not be used inside a program loop.** Be sure all settings are correct before saving.

***SAV 0**	'saves current values and configuration
***RCL 0**	'recalls the saved configuration

3.9 OEM Documentation Guidelines

OEM users of the 4809 and 4809 boards should provide the end user with the necessary instructions to operate the complete system. In most cases this includes directions for:

1. Setting the product's GPIB Address or serial address.
2. How to use the 4809/4809 Modbus commands to control the host device. (Includes sending outputs and reading inputs if applicable). The OEM needs to define the commands in terms of what they do to the host unit and how the end user should use them.
3. Using the trigger functions if applicable.
4. Using the 488.2 Status Reporting Structure. The OEM needs to define what the digital inputs mean if they are part of the system, how to enable SRQs and how to read the registers.

The SCPI Standard requires that the SCPI command tree and SCPI conformance information be passed on to the end user. This only means the active or applicable commands. Locked commands become invisible to the end user and should be omitted from the end user's SCPI command tree and list.

OEM users are hereby given permission to copy any portion of this manual for the purpose of documenting systems that incorporate the Model 4809 or a 4809 OEM Board. Reproduction of this manual for other purposes without the expressed written consent of ICS Electronics is forbidden.

4

Theory of Operation

4.1 INTRODUCTION

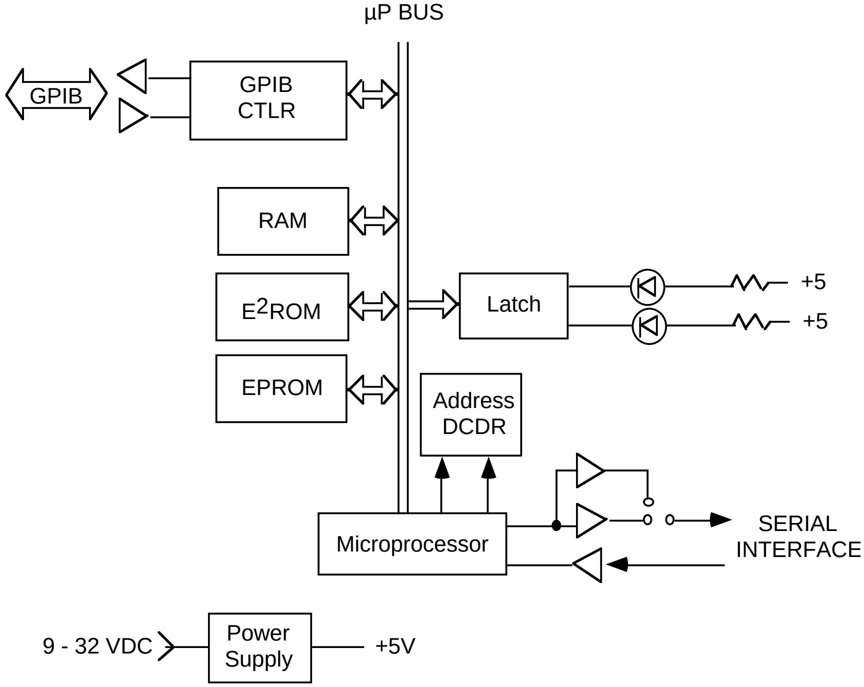
This section describes the theory of operation of the 4899/4809. Wherever the text refers to the Model 4899, it applies equally to the Model 4809 unless otherwise noted.

4.2 BLOCK DIAGRAM DESCRIPTION

A block diagram of the 4899 is shown in Figure 4-1. The 4899 is a microprocessor based device that transparently passes data between the GPIB (IEEE 488) bus and a Modbus slave device over a serial link. The 4899 is made up of seven major elements, most of which are interconnected to the microprocessor by a common data, address and control signal bus. The 4809's block diagram is similar to the 4899 but without the internal power supply.

Incoming GPIB bus data and commands are received by the GPIB controller chip. Each received character interrupts the microprocessor to place the characters in the GPIB received data buffer. When a complete message has been received, the parser checks the message for a valid command and then acts upon it. Invalid messages cause a bit to be set in the unit's Event Status Register and turn on the ERR LED.

Modbus commands are converted into a series of binary characters to make up the Modbus RTU message. The Modbus message includes



4

Figure 4-1 4899 Block Diagram

the slave device address, the command number, the registers and data (if any) that is being sent to the registers. A checksum is added to make up the complete Modbus RTU message. The Modbus message is then placed in the serial transmit buffer. From the serial transmit buffer, the data characters are sequentially placed in the microprocessor's UART where they are serialized, passed through the selected serial driver and outputted at the serial interface.

SCPI commands and IEEE-488.2 commands are parsed and used to set control parameters, perform an operation or query a parameter. Responses are placed in the GPIB buffer so they can be returned to the host controller when the unit is next addressed to talk.

Incoming serial data from the Modbus slave device is received, converted into TTL levels and applied to the UART's input. Each received character is temporarily stored in the serial received data buffer. The characters in the received message are counted and verified against the expected response character count. The message is then checksumed. If the received message is a valid response, any data is converted in to the correct format and placed in the GPIB buffer where can be transferred out onto the GPIB bus when the unit is next address to talk. Messages that contain errors or Exception messages cause the 4899/4809 to set bits in the Questionable Register and to place an error value in the Modbus Error Register. The 4899/4809 contains a multilevel Status Byte Register and Event Register structure enables the 4899 to generate a Service Request and interrupt the GPIB bus controller when errors are detected.

EPROM or Flash Memory contains all of the 4899's program instructions, command tables, and power turn-on/self test routines. At power turn-on, the 4899 performs a self test on each functional block to determine whether there is a gross system failure. Any self test error is displayed as a pattern of blinking LEDs on the front panel. The error pattern is repeated until the unit is turned off. Just after completing the self test routine, the 4899/4809 displays its current GPIB address setting on the front panel LEDs. Bit weights are read from right to left with the least significant bit on the far right. The RDY LED comes on to indicate a successful completion of the self test routine.

4

The 4899's and 4809's configuration settings, serial number and other parameters that are subject to change are saved in nonvolatile E^2ROM or Flash Memory. At power on time, the microprocessor copies the saved configuration to RAM where it is used to operate the unit. Any changes made to the settings during run time are <u>not</u> stored in the nonvolatile memory until the user sends the *SAV 0 command.

In the 4899/4809 , the RAM is a 8 bit wide memory that is primarily used for data storage, operating variables and configuration settings. The 4899/4809 data buffers are mechanized as straight buffers because of the Modbus command-response protocol. The buffers are several times larger than any anticipated message so no data loss ever occurs.

GPIB bus data is never lost since the 4899/4809 simply inhibits further Bus handshakes until there is room in the GPIB buffer for more data.

The 4899's power supply is a switching regulator that converts a unregulated 9 to 32 volt DC input to +5 Vdc to run the 4899's internal logic chips. A DC-DC converter in the RS-232 transmitter IC makes ± 9 Vdc to operate the RS-232 drivers.

4.3 4809 DIFFERENCES

The Model 4809 uses +5 to +4.3 Vdc power supplied from the host power supply. A diode in series with the power input pads reduces the input voltage by 0.3 volts. The voltage measured across the test points should be +5 ± 0.2 Vdc for proper operation. If the voltage is too low, the series diode should be bypassed with a jumper.

Troubleshooting and Repair

5.1 INTRODUCTION

This section describes the maintenance, troubleshooting and repair procedures for the Model 4899 and 4809 GPIB <-> Modbus Serial Interfaces. All comments and errors apply equally to both units unless otherwise stated.

5.2 MAINTENANCE

The 4899 and the 4809 do not require periodic calibration and have no internal adjustments. However, if the 4899 is used in an application where the IEEE 488 bus cables are frequently changed or if the input signals appear erratic, the 4899's GPIB connector may require cleaning to remove wax and dirt buildup. New bus and other 'blue ribbon' type connectors are shipped with a brightener on them. (The brightener is a thin wax like film) Depending upon cable usage, enough of the brightener may buildup on the 4899's connector to cause intermittent operation.

The brightener is an organic compound and may be cleaned off by washing the connector with a mild detergent solution followed by an alcohol wash.

5.3 TROUBLESHOOTING

Troubleshooting is broken down into self test errors and operating errors that are caused during usage.

5.3.1 Self Test Errors

The 4899 and 4899 indicate self test errors by blinking one or more of its LEDs at a 2 cps rate. Verify the error by turning the unit off for 10 seconds, disconnect the unit from any other equipment and then turn the power back on. If the error persists it is a true self test error. The self test error codes and their most likely problems are listed in Table 5-1.

5.3.2 Operating Failures

Use the fault isolation information in Table 5-2 to narrow the operational problem down to a specific area. The majority of installation faults can be fixed by following the table and making the necessary corrections to the installation wiring or the program. Failures after the unit has been running a while can be isolated by first substituting a known good unit or output/input channel.

WARNING

If the fault isolation procedure requires internal measurements, always remove power when disassembling or assembling the unit. Use extreme caution during troubleshooting, adjustments, or repair to prevent shorting components and causing further damage to the unit.

TABLE 5-1 4899/4809 - SELF TEST ERROR CODES

Blinking LED	Error	Possible Fault
All	E^2ROM Memory	E^2ROM Memory corrupted and unit reset E^2ROM to factory settings at power turn-on. Power unit off and back on to clear the blinking LEDs. See Table 5-2 if the ERR LED comes on when power is reapplied.
RDY	E^2ROM	Loose E^2ROM in socket. Check part for bent pins and press into socket. U11 in 4899 or U3 in 4809 Defective E^2ROM U8 that cannot be written to. Replace part U11 in 4899 or U3 in 4809 Defective decoder GAL. Test and/or replace GAL. U10 in 4899 or U12 in 4809.
TALK	EPROM	EPROM loose in its socket or has a bent pin. Check EPROM for a bent pin then press EPROM into its socket. EPROM Memory dropped a bit. Replace the defective EPROM with a known good EPROM . Defective decoder GAL. Test and/or replace GAL. U10 in 4899 or U12 in 4809.
LSTN	RAM	Defective RAM. Replace RAM with new IC. U26 or 27 in 4899 or U2 in 4809. Defective decoder GAL. Test and/or replace GAL, U13 in 4899, U13 in 4899.

5

TABLE 5-1 4899/4809 - SELF TEST ERROR CODES CONTINUED

Blinking LED	Error	Possible Fault
SRQ	GPIB	Defective GPIB controller chip. Replace GPIB Controller, U15 in 4899 or U5 in 4809.
ERR	Address Setting	Address value should be between 0 and 30. Check and or correct address setting.
SRQ + ERR	GPIB	Wrong firmware or GPIB chip for hardware configuration. Check hardware configuration and change firmware or replace GPIB Controller chip with NI 7210. U15 in 4899 or U5 in 4809.
ERR + LSTN	CPU type	Wrong CPU type. Should be Zilog Z8S18020VSC (SL1919 Enhanced Version)
ERR + LSTN + TALK	CPU type	Wrong CPU type. Should be Zilog Z8S18020VSC

Solid LED	Error	Possible Fault
PWR (After blinking address)	Program hung	Open GPIB chip selection line or grounded interrupt into Z180.
RDY off (After selftest)	DCD low	DCD input should be high. Check cable wiring andslave device is handshaking signals used in cable.

5

TABLE 5-2 TROUBLESHOOTING GUIDE

Symptom	Possible Fault	Action or Check
Unit will not turn on	Power cord not plugged in	Push power cord into DC receptacle
	Power at AC outlet	Check outlet and power adapter
	High output lines shorted to ground	Disconnect output signals and reapply power to test the unit. If it powers on, remove the short or put resistors in the offending circuit path.
Unit shows a blinking LED at power turn on	Self test fault	Check Self Test errors in Table 5-1
ERR LED on at power turn-on	E^2ROM data lost	Use *CLS to clear the LED. Use CAL:DATE command to accept default configuration and clear the error so the ERR LED will not come on at next power-on time. Reload your configuration and use the *SAV 0 command to save the new configuration. Recall the factory defaults with the CAL:DEFAULT command. Use *SAV 0 to save the factory configuration.
Unit fails to respond or responds wrong after an address change	No delay after an the address change	Provide a 70 ms minimum delay after changing 4899's GPIB address.
	Insufficient delay	Program running on faster CPU or in a compiled form runs faster. Change to a called time function and test CPU clock.

5

TABLE 5-2 TROUBLESHOOTING GUIDE CONTINUED

Symptom	Possible Fault	Action or Check
No response from GPIB commands	Wrong GPIB address	Turn unit off and back on. Watch LEDs to check address setting.
ERR LED comes on	Bad command	Check command syntax. Query the ESR register to determine the cause of the error.
	Bit set in ESR register	Check ESR register and use Figure 3-1 to determine the cause of the problem.
No communication with Modbus device	Device address wrong	Query the current Modbus device address with the C? query. Set Modbus device address with the C n command.
	Command syntax	Check access LED on Modbus deviceto verify serial message received by the Modbus device. Modbus device manual for applicable commands, registers and values. 4899/4809 Modbus Error register for Modbus problems.
No communication on RS-485 link.	RS-485 not enabled	4899/4809 RS-485 not enabled. Check setting with the SYST:COMM:SER:RS485? query.
Bad RS-485 data	No termination on RS-485 link.	Missing pullup or pulldown resistors. Add resistors to RS-485 network.

5

5.4 4899/4809 Factory Default Recovery

If a 4899 or 4809's configuration gets into an unknown setting, it can be restored to the factory default configuration with the following procedure:

1. Turn the 4899/4809's power off.

2. Place a jumper on W1 or short out the two posts on W1.

3. Turn the 4899/4809's power on. Wait until the LEDs stabilize and the 4809 has finished its save procedure (takes approximately 10 seconds).

4. Remove the jumper from W1 and operate the unit normally. All parameters should have been restored to the settings listed in Table 1-2 or 1-3.

5.5 REPAIR

Repair of the 4899 or 4809 is done by the user or by returning the unit to the factory or to your local distributor. Units in warranty should **always** be returned to the factory or else repaired only after receiving permission to do so from an ICS customer service representative.

When returning a unit, a board assembly, or other products to ICS for repair, it is necessary to go through the following steps:

1. Contact the ICS customer service department and ask for a return material authorization (RMA) number. An ICS application engineer will want to discuss the problem at this time to verify that the unit needs to be returned, or to assist in correcting the problem. We have discovered that one-third of the difficulties customers call about can be resolved over the phone as opposed to returning a unit for repair.

2. Write a description of the problem and attach it to the material being returned. Describe the installation, system failure symptoms, and how it was being used. If the item being returned is a board assembly, describe how you isolated the fault to it. Include your name and phone number so we can call you if we have any questions. Remember, we need to locate the problem in order to fix it.

3. Pack the item with the fault description in a box large enough to accommodate a minimum of two inches of packing material on all four sides, the top, and the bottom of the box. Securely seal the box.

4. Mark the shipping label to the attention of RMA#. The RMA number is very important since it is our way of identifying your unit in order to return it to you.

5. Ship the box to ICS freight prepaid. ICS does not pay freight to return the unit to ICS, but will prepay the freight to return the repaired item to you.

Appendix

Appendix **Page**

A1	IEEE 488 Bus Description	A-2
A1.1	IEEE 488.1 Bus	A-2
A1.2	IEEE 488.2 Standard	A-9
A1.3	SCPI Commands	A-13
A2	Serial Data Communications Background	A-17
A2.1	Introduction to Serial Communication	A-27
A2.2	RS-232 Standard	A-22
A2.3	RS-422 and RS-485 Standards	A-24
A2.4	RS-530 Pinouts	A-25
A2.5	Serial Interface Problems	A-25
A3	GPIB Connector/Switch Board Assemblies	A-27
A3.1	Board Descriptions	A-27
A3.2	Dimensions and Installation	A-28

A1 IEEE 488 BUS DESCRIPTION (IEEE 488.1, IEEE 488.2, SCPI)

The IEEE Std 488 Bus is a convenient means of connecting instruments and computers together to form a test system or to transfer data between two computers. The IEEE Std 488.1 covers the electrical and mechanical bus specifications and the state diagrams for each bus function. The IEEE Std 488.2 expanded on the original specification and established data formats, common commands for each 488.2 device and controller protocols. The SCPI standard developed a tree like series of standard commands for programmable instruments so that similar instruments by different manufacturers can be controlled by the same program.

The 488-PC2 card provides an IEEE 488 Interface for any IBM PC computer or compatible ISA bus clone. When used as a bus controller, the 488-PC2 drivers, driver libraries and windows DDL make the 488-PC2 operate as a 488.2 compatible controller. The controller protocols are built in to the PC2 drivers. All IEEE 488.2 common commands, queries and SCPI commands are placed in the output command string by the user and are not part of the PC2 driver software.

A1.1 IEEE 488.1 Bus

The IEEE Std 488 Bus, or GPIB as it is commonly referred to, provides a means of transferring data and commands between devices. The physical portion of the bus is governed by IEEE -Std 488.1 - 1978. The interface functions for each device are contained within that device itself, so only passive cabling is needed to interconnect the devices. The cables connect all instruments, controllers and other components of the system in parallel to the signal line as shown in Figure A-1. Eight of the lines (DIO1-DIO8) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner. Data and message transfer is asynchronous, coordinated by the three handshake lines (DAV, NRFD, NDAC). The other five lines control Bus activity.

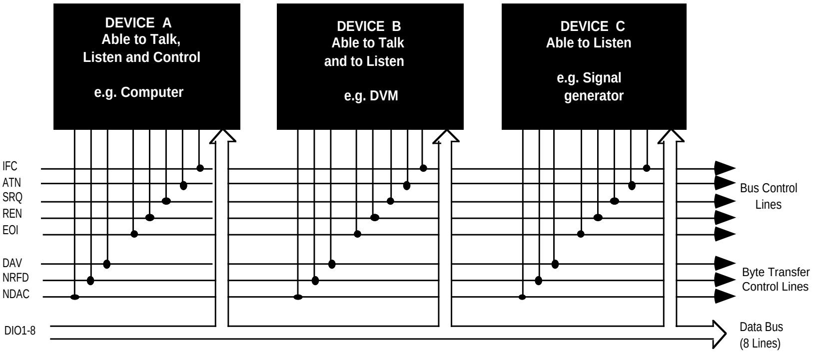


Figure A-1 IEEE 488 Bus

Two types of messages are transferred over the bus:

> Interface messages - for bus management
>
> Device-dependent messages - for device control and data transfer

Devices connected to the bus may act as talkers, listeners, controllers, or combinations of the three functions, depending upon their internal capability. The system controller is a controller that becomes active at power turn-on. It is the Bus manager and the initial controller-in-charge.

A controller **can send interface messages to manage the other devices, address devices to talk or listen and command specific actions within devices.**

A talker **sends device dependent messages, i.e., data, status.**

A listener **accepts interface messages, bus commands and device-dependent messages, i.e., setup commands, data.**

Bus systems can be as simple as two devices; one a talker always sending data to a second device which listens to the data. Larger systems can have one or more controllers and many devices (the IEEE 488 driver specifications limit the total number of units on one bus

A

system to 15). Only one controller can be the controller-in-charge at any given time. Control originates with the system controller and is passed back to other controller(s) as required. Control can be passes back to the system controller or to another controller after the completion of the task. The system controller has the capability of taking control back at any time and resetting all addressed devices to their unaddressed state.

Each bus device is identified by a five-bit binary address. There are 31 possible primary addresses 0 through 30. Address 31 is reserved as the 'untalk' or 'unlisten' command. Some devices contain subfunctions, or the devices themselves may be addressed by a secondary five-bit binary address immediately following the primary address, i.e. 1703. This secondary address capability expands the bus address range to 961 addresses. Most bus addresses are set at the time the system is configured by rocker switches which are typically located on each devices' rear panel. Devices that are SCPI 1991 compatible, can have their bus address set by a GPIB SYSTEM configuration command.

Information is transmitted on the data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer proceeds as fast as the devices respond (up to 1 Mbs), but no faster than that allowed by the slowest addressed device. This permits several devices to receive the same message byte at the same time. Although several devices can be addressed to listen simultaneously, only one device at a time can be addresses as a talker. When a talk address is put on the data lines, all other talkers are normally unaddressed.

ATN (attention) is one of the five control lines and is set true by the controller-in-charge while it is sending interface messages or device addresses. The messages are transmitted on the seven least significant data lines and are listed in the MSG columns in Table A-1. When a device is addressed as a talker, it is allowed to send device-dependent messages (e.g., data) when the controller-in-charge sets the ATN line false. The data messages are typically a series of ASCII characters ending in a CR, LF, or CR LF sequence. The data messages often consist of eight-bit binary characters and end on a predetermined count or when the talker asserts the EOI line simultaneously with the last data byte. The controller-in-charge must be programmed to correctly respond to each device's message termination sequence to avoid hanging-up the system or leaving characters that will be output when the device is addressed as a talker again.

IFC (interface clear) is sent by the system controller and places the interface system in a known quiescent state with all devices unaddressed.

REN (remote enable) is sent by the system controller and is used with other interface messages or device addresses to select either local or remote control of each device.

SRQ (service request) is sent by any device on the bus that wants service, such as counter that has just completed a time-interval measurement.

ASCII -- IEEE 488 BUS MESSAGES (COMMANDS AND ADDRESS) HEX CODES

LSD \ MSD	0		1		2		3		4		5		6		7	
	ASCII	MSG	ASCII	MSG	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16	`	▲	p	▲
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	Q	17	a		q	
2	STX		DC2		"	02	2	18	B	02	R	18	b		r	
3	ETX		DC3		#	03	3	19	C	03	S	19	c		s	
4	EOT	SDC	DC4	DCL	$	04	4	20	D	04	T	20	d		t	
5	ENQ	PPC	NAK	PPU	%	05	5	21	E	05	U	21	e		u	
6	ACK		SYN		&	06	6	22	F	06	V	22	f		v	
7	BEL		ETB		'	07	7	23	G	07	W	23	g	MEANING DEFINED BY PCG CODE	w	MEANING DEFINED BY PCG CODE
8	BS	GET	CAN	SPE	(	08	8	24	H	08	X	24	h		x	
9	HT	TCT	EM	SPD	)	09	9	25	I	09	Y	25	i		y	
A	LF		SUB		*	10	:	26	J	10	Z	26	j		z	
B	VT		ESC		+	11	;	27	K	11	[	27	k		(	
C	FF		FS		,	12	<	28	L	12	\	28	l		|	
D	CR		GS		-	13	=	29	M	13	]	29	m		)	
E	SO		RS		.	14	>	30	N	14	^	30	n		~	
F	SI		US		/	15	?	UNL	O	15	_	UNT	o	▼	DEL	▼

ADDRESSED COMMAND GROUP (MSD 0) — UNIVERSAL COMMAND GROUP (MSD 1) — LISTEN ADDRESS GROUP (MSD 2–3) — TALK ADDRESS GROUP (MSD 4–5) — SECONDARY COMMAND GROUP (MSD 6–7)

PRIMARY COMMAND GROUP (PCG) (MSD 0–5)

Notes:

1. Device Address messages shown in decimal
2. Message codes are:

   DCL -- Devices Clear
   GET -- Device Trigger
   GTL -- Go to Local
   LLO -- Local Lockout
   PPC -- Parallel Poll Configure
   PPU -- Parallel Poll Unconfigure
   SDC -- Selected Device Clear
   SPD -- Serial Poll Disable
   SPE -- Serial Poll Enable
3. ATN off, Bus data is ASCII; ATN on, Bus data is an IEEE MSG.

EOI (end or identify) is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller-in-charge sets both the ATN and EOI lines true, each device configured to respond to a parallel poll indicates its current status on the DIO line assigned to it.

Bus Commands are transmitted when ATN is asserted. The commands are listed in the message columns in Table 1 (on the left hand page) which shows the relationship between the commands and ASCII data characters. ASCII data characters have the same code values as bus commands but are transmitted with ATN off. The following chart lists the standard command and address mnemonics.

Address Commands

MLA	My listen address (controller to self)
MTA	My talk address (controller to self)
LAD	Device listen address
TAD	Device talk address
SAD	Secondary Device address (device optional address)
UNL	Unlisten
UNT	Listen

Universal Commands (to all devices)

LLO	Local Lockout
DCL	Device Clear
PPU	Parallel Poll Unconfigure
SPE	Serial Poll Enable
SPD	Serial Poll Disable

Addressed Commands (to addressed listeners only)

SDC	Selected Device Clear
GTL	Go to Local
GET	Device Trigger
PPC	Parallel Poll Configure
TCT	Take Control

A

Devices on the bus are normally interconnected by cables with dual male/female connectors at each end to allow easy cable stacking. The 24 conductor cable pinouts are shown in Figure A-2. Signal levels are 0 and 3.3 Vdc with 0 being the logic true level. Cable connectors are modified Amphenol 24 pin Blue ribbon style connectors (57-30240) with metric jack screws.

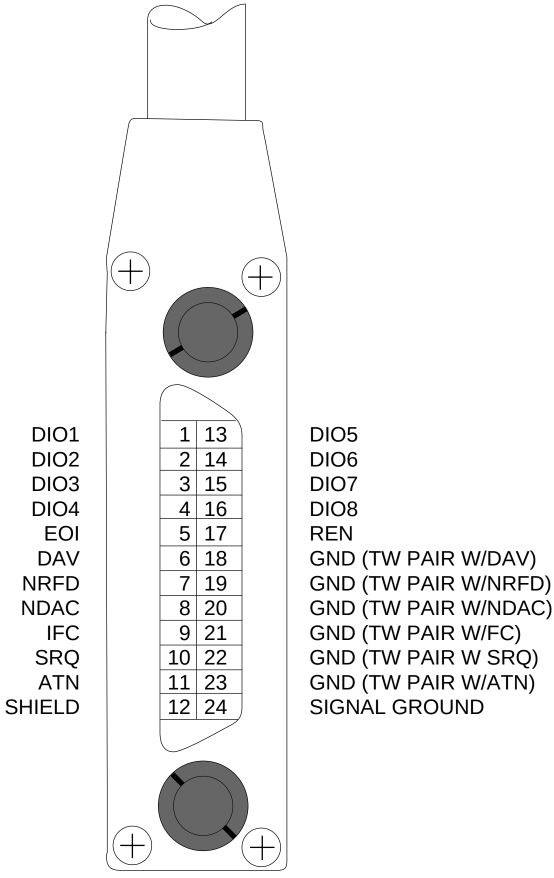


Figure A-2 GPIB Signal-Pin Assignments

A

A1.2 IEEE 488.2 STANDARD

A1.2.1 IEEE 488.2 Message Formats

The IEEE 488.2 Standard was established in 1987 to standardize message protocols, status reporting and define a set of common commands for use on the IEEE 488 bus. IEEE 488.2 devices are supposed to receive messages in a more flexible manner than they send. A message sent from GPIB controller to GPIB device is called: PROGRAM MESSAGE. A message sent from device to controller is called: RESPONSE MESSAGE. As part of the protocol standardization the following rules were generated:

(;) Semicolons are used to separate messages.
(:) Colons are used to separate command words.
(,) Commas are used to separate data fields.
<nl> Line feed and/or EOI on last character terminates a 'program message'. Line feed (ASCII 10) **and** EOI terminates a RESPONSE MESSAGE.
(*) Asterisk defines a 488.2 common command.
(?) Ends a query where a reply is expected.

A1.2.2 IEEE 488.2 Reporting Structure

With IEEE 488.2, status reporting was enhanced from the simple serial poll response byte in IEEE 488.1 to the multiple register concept shown in Figure A-3. The IEEE 488.2 Standard standardized the bit assignments in the Status Byte Register, added eight more bits of information in the Event Status Register and introduced the concept of summary bits reporting to the Status Byte Register. The Status and Event registers have enabling registers that can control the generation of their summary reporting bits and ultimately SRQ generation. Each 488.2 device must implement a Status Byte Register, a Standard Event Status Register and an Output Message Queue as a minimum status reporting structure. A device may include any number of additional condition registers, event registers and enabling registers providing they follow the model shown in Figure A-3.

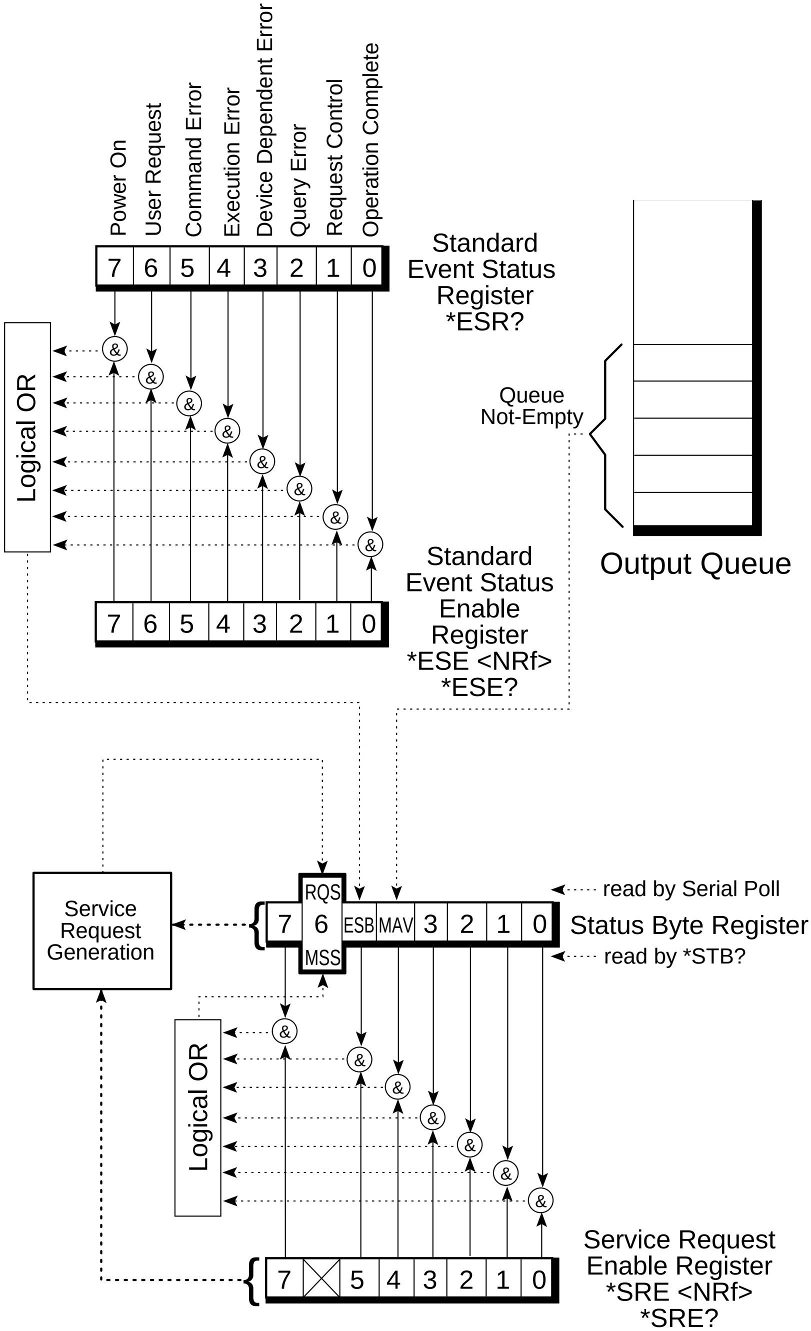


Figure A-3 488.2 Required Status Reporting Capabilities

A

TABLE A-2 IEEE 488.2 COMMON COMMANDS

Required common commands are:

***CLS**	Clear Status Command
***ESE**	Standard Event Status Enable Command
***ESE?**	Standard Event Status Enable Query
***ESR?**	Standard Event Status Register Query
***IDN?**	Identification Query
***OPC**	Operation Complete Command
***OPC?**	Operation Complete Query
***RST**	Reset Command
***SRE**	Service Request Enable Command
***SRE?**	Service Request Enable Query
***STB?**	Status Byte Query
***TST?**	Self-Test Query
***WAI**	Wait-to-Continue Command

Devices that support parallel polls must support the following three commands:

***IST?**	Individual Status Query?
***PRE**	Parallel Poll Register Enable Command
***PRE?**	Parallel Poll Register Enable Query

Devices that support Device Trigger must support the following commands:

***TRG**	Trigger Command

Controllers must support the following command:

***PCB**	Pass Control Back Command

Devices that save and restore settings support the following commands:

***RCL**	Recall configuration
***SAV**	Save configuration

Devices that save and restore enable register settings support the following commands:

***PSC**	Saves enable register values and enables/disables recall
***PSC?**	PSC value query

A

A1.2.3 IEEE 488.2 Common Commands

The IEEE 488.2 Standard also mandated a list of required and optional Common Commands that all 488.2 devices could support. All of the Common Commands start with an asterisk. Commands that end with a question mark are queries. Query responses can be an ASCII number or an ASCII string. Other numerical formats are legal as long as the device supports the required ASCII format. Table A-2 lists the IEEE 488.2 Common Commands.

A1.2.4 IEEE 488.2 Differences From IEEE 488.1

The user who is familiar with the older 488.1 devices should take the following differences into account when programming a 488.2 device.

A 488.2 device outputs the Status Byte Register contents plus the RQS bit in response to a serial poll. The RQS bit is reset by the serial poll. The same 488.2 device outputs the Status Byte Register contents plus the MSS bit in response to a *STB? query. The MSS bit is cleared when the condition is cleared.

488.2 restricts the Device Clear to only clearing the device's buffers and pending operations. It does not clear the Status Reporting Structure or the output lines. Use *CLS to clear the Status Structure and *RST or *RCL to reset the outputs.

488.2 commands are really special data messages and are executed by the device's parser. Always allow sufficient time for the parser to execute the commands before sending the device a 488.1 command. i.e. a Device Clear sent too soon will erase any pending commands and reset the parser.

Enable Register values are only saved and restored if the *PSC command is 0. A *PSC command of 1 causes zeros to be loaded into the enable registers when the unit is next reset or powered on.

A1.3 SCPI COMMANDS

A1.3.1 Introduction

SCPI (Standard Commands for Programmable Instruments) builds on the programming syntax of 488.2 to give the programmer the capability handling a wide variety of instrument functions in a common manner. This gives all instruments a common "look and feel".

SCPI commands use common command words defined in the SCPI specification. Control of any instrument capability that is described in SCPI shall be implemented exactly as specified. Guidelines are included for adding new defined commands in the future as new instruments are introduced without causing programming problems.

SCPI is designed to be laid on top of the hardware - independent portion of the IEEE 488.2 and operates with any language or graphic instrument program generators. The obvious benefits of SCPI for the ATE programmer is in reducing the learning time on how to program multiple SCPI instruments since they all use a common command language and syntax.

A second benefit of SCPI is that its English like structure and words are self documenting, eliminating the needs for comments explaining cryptic instrument commands. A third benefit is the reduction in programming effort to replace one manufacturer's instrument with one from another manufacturer, where both instruments have the same capabilities.

This consistent programming environment is achieved by the use of defined program messages, instrument responses and data formats for all SCPI devices, regardless of the manufacturer.

A1.3.2 Command Structure and Examples

SCPI commands are based on a hierarchical structure that eliminates the need for most multi-word mnemonics. Each key word in the command steps the device parser out along the decision branch - similar to a squirrel hopping from the tree trunk out on the branches to the leaves. Subsequent keywords are considered to be at the same branch level until a new complete command is sent to the device. SCPI commands may be abbreviated as shown by the capital letters in Figure A-4 or the whole key word may be used when entering a command. Figure A-4 shows some single SCPI commands for setting up and querying a serial interface.

SYSTem:COMMunicate:SERial:BAUD 9600 <nl>
Sets the baud rate to 9600 baud

SYST:COMM:SER:BAUD? <nl>
Queries the current baud setting

SYST:COMM:SER:BITS 8 <nl>
Sets character format to 8 data bits

Figure A-4 SCPI Command Examples

Multiple SCPI commands may be concatenated together as a compound command using semi colons as command separators. The first command is always referenced to the root node. Subsequent commands are referenced to the same tree level as the previous command. Starting the subsequent command with a colon puts it back at the root node. IEEE 488.2 common commands and queries can be freely mixed with SCPI messages in the same program message without affecting the above rules. Figure A-5 shows some compound command examples.

SYST:COMM:SER:BAUD 9600; BAUD? <nl>

SYST:COMM:SER:BAUD 9600; :SYST:COMM:SER: BITS 8 <nl>

SYST:COMM:SER:BAUD 9600; BAUD?; *ESR?; BIT 6; BIT?; PACE XON; PACE?; *ESR? <nl>

Figure A-5 Compound Command Examples

A typical response would be: **9600; 0; 8; XON; 32 <nl>**

The response includes five items because the command contains 5 queries. The first item is **9600** which is the baud rate, the second item is **ESR=0** which means no errors (so far). The third item is **8** (bit/word) which is the current setting. The BIT 6 command was not accepted because only 7 or 8 are valid for this command. The fourth item **XON** means that XON is active. The last item is **32** (ESR register bit 5) which means execution error - caused by the BIT 6 command.

A1.3.3 Variables and Channel Lists

SCPI variables are separated by a space from the last keyword in the SCPI command. The variables can be numeric values, boolean values or ASCII strings. Numeric values are typically decimal numbers unless otherwise stated. When setting or querying register values, the decimal variable represents the sum of the binary bit weights for the bits with a logic '1' value. e.g. a decimal value of 23 represents 16 + 4 + 2 + 1 or 0001 0111 in binary. Boolean values can be either 0 or 1 or else OFF or ON. ASCII strings can be any legal ASCII character between 0 and 255 decimal except for 10 which is the Linefeed character.

Channel lists are used as a way of listing multiple values. Channel lists are enclosed in parenthesis and start with the ASCII '@' character. The values are separated with commas. The length of the channel list is determined by the unit. A range of values can be indicated by the two end values separated by a colon. e.g.

(@1,2,3,4)	lists sequential values
(@ 1:4)	shows a range of sequential values
(@ 1,5,7,34)	lists random values

Figure A-6 Channel List Examples

A1.3.4 Error Reporting

SCPI provides a means of reporting errors by responses to the **SYST:ERR?** query. If the SCPI error queue is empty, the unit responds with 0, "No error" message. The error queue is cleared at power turn-on, by a ***CLS** command or by reading all current error messages. The error messages and numbers are defined by the SCPI specification and are the same for all SCPI devices.

A1.3.5 Additional Information

For more information about SCPI refer to the SCPI Standard or to the SCPI section in any SCPI compatible instrument manual.

A2 SERIAL DATA COMMUNICATIONS BACKGROUND

A2.1 INTRODUCTION TO SERIAL COMMUNICATION

Serial data communication is the most common means of transmitting data from one point to another. In serial communication systems, the data word or character is sent bit by bit over some kind of transmission path. The receiving device recognizes each bit as they are received and reassembles them back into the original data word. Serial data communication systems are characterized by four primary factors:

1. Data speed or baud rate

2. Data format

3. Transmission medium

4. Clocking method

Serial data speed is referred to as Baud Rate. A baud is defined as a signaling bit, which includes data bits as well as start/stop framing, parity or any other bits that make up the data format. Typical computer baud rates and their uses are:

110 - for old mechanical teletypes

300, 1200 - for low speed devices an older modems

9600 to 38400 baud for high speed devices and newer modems

Data format refers to the method or pattern the transmitter uses to send the data word or character as a series of bits so that the receiver will know how to recognize the pattern and reassemble the bits back into the original data word. The most common method and the one used in the 2303, is called asynchronous transmission because each character is sent one at a time with an undetermined amount of time between characters. Each asynchronous character has a low going start bit, a number of data bits, an optional parity bit and 1 or 2 high stop bits. The transmitter automatically extends the stop bit when it has no more characters to transmit. The receiver uses the start bit to resynchronize its clock with the data at the start of each character as shown in Figure A-7.

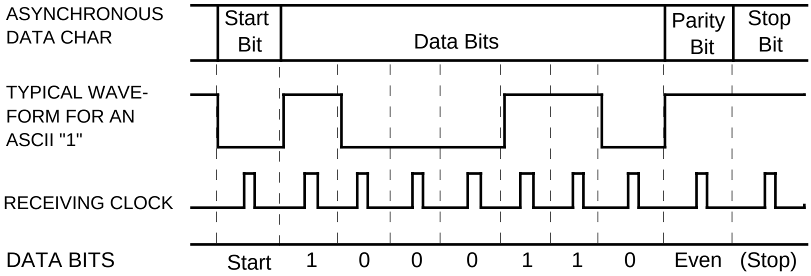


Figure A-7 Asynchronous Data Character Waveforms

Synchronous character do not have start/stop bits and are sent without spaces between characters. Voids between data characters are filled by predetermined sync characters which are discarded by the receiver.

The data portion of the serial character usually contains 5 to 8 bits and is transmitted least significant bit first. Today most of the computers and terminals use the 7 bit ASCII code to represent numbers and characters. Figure A-7 shows how the ASCII "1" is transmitted. Compare the binary code in Figure A-7 against the hex code for an ASCII '1' (HEX 31) and they will be the same. Binary data is usually sent in binary form as single 8 bit characters or in hex form as a pair of the ASCII characters, 0 through 9 and A through F. Each Hex character represents 4 binary bits so two Hex characters are needed for each 8 bit binary byte.

Parity bits are added after the data field if the user wants to detect transmission errors. When parity bits are used, the transmitter counts the number of high bits in the data field and makes the parity bit a 1 or 0 so the final count will be either even or odd. The receiver then validates the received characters by counting 1's in the data and parity bit fields. The 2303 detects parity errors along with data overrun and framing errors, generates a Bus SRQ message for each data error and indicates the error by setting the bit 3 in the Standard Event Status Register.

Although serial data can be transmitted over any medium, most of today's computer systems use metallic cable. To ensure compatibility, the manufacturers have adopted interface standards so that they are electrically compatible. The more popular standards are:

RS-232	Most popular standard for office machines and computer systems.
RS-422 and RS-485	New high speed standard with noise improvements over RS-232 for longer distances.

Devices employing the same interface standard can usually be connected together but the user **must** verify each devices signal requirements before plugging them together.

When data must be transmitted over long distances, it is typically sent over the phone company's direct dial network (DDN) as shown in Figure A-8. Modems are used to convert the serial data bits into tones that will pass through the telephone system's 300 to 3000 Hz voice band. For low baud rates, up to 1200 Hz, the modems convert the bits into two tones (frequency switched keying) that the receiving modem recognizes and converts back to data bits. These low speed modems are referred to by the telephone company's designations, i.e.: Type 103 (300 baud) and Type 212 (1200 baud). Higher data rates require more complex modulation techniques and the modems are referred to by their CCITT specification i.e., V22.

With asynchronous characters, the receiver normally uses the start bit to synchronize its internal clock. However, some devices, such as the higher speed modems, require the data bits to be synchronized with their clock. These units are referred to as synchronous modems (not the same as synchronous data characters) and they will supply the clock signals to both the transmitting and receiving device.

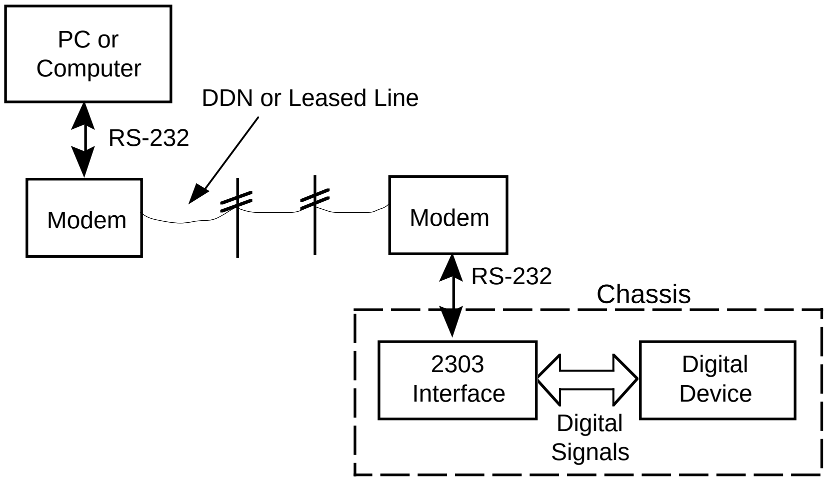


Figure A-8   Long Distance Communication using Modems

Another aspect of timing is the control of data transmission to avoid data overrun. The two methods used are control signals and X-on/X-off characters.

For the control signal method, extra wires are provided in the cable for handshake signals that enable or inhibit data flow. The more common control signal pairs are:

Request-to-send / Clear-to-send
Data-terminal-ready / Data-set-ready

All signals must be high to enable data transmission. Dropping any line normally means the receiving device's buffer is full or it is busy with the last message.

Another method of controlling the data flow is to imbed X-on/X-off characters in the data message. At turn on, both devices are initially in the X-on state. When one device becomes full, it sends the other an X-off character to inhibit future data transmission. X-on is then sent to restart the data transmission when there is room in the receiving device's buffer for additional data.

A2

The 2303's Serial Interface normally uses asynchronous 8 bit data characters with no parity and single start and stop bits. The 2303 will also work with 7 bit data characters. The unused data bits are outputted on the 488 Bus as fixed zeros. The user can also add a parity bit and the second stop bit if required for his system.

A2

A2.2 RS-232 STANDARD

In 1963, the Electronic Industry Association (EIA) established a standard to govern the interface between data terminal equipment and data communication equipment employing serial binary interchange. The latest revision of this standard (RS-232) has been in effect since 1969 and is known as RS-232C. It specifies:

- Mechanical characteristics of the interface
- Electrical characteristics of the interface
- A number of interchange circuits with descriptions of their functions
- The relationship of interchange circuits to standard interface types

The specification does not mean that two devices that are RS-232 compatible can be connected together with a standard cable and be expected to work.

Mechanically, RS-232 interfaces use a 25 pin male connector (DS-25P) with the data terminals and a 25 pin female connector (DS-25S) with the data communications units (modems).

A2

Electrically, RS-232 signals are bipolar and are referenced to a common ground (AB) on pin 7. Transmitted signals must be between $^{+}5$ and $^{+}15$V or -5 and -15V into 3000 to 7000 ohm loads. Maximum open circuit transmitter outputs is ±25V. Logic levels are:

	+5 to +15V	-5 to -15V
Data	0	1
Control	1 (On)	0 (Off)

Functionally, the specification established two types of devices, DCE and DTE, that would mate together by a pin-to-pin cable. The Data Communication Equipment (DCE) was designated as the device that connected to the communication line. An example of a DCE is a modem. The Data Terminal Equipment (DTE) was designated as the device that connected to the DCE. Examples of a DTE are a PC computer or a terminal. DTE devices can be mated to DTE devices by a special 'null-modem' cable that crosses the transmit signals of one device with the receive signals on the other device.

In Europe, the Comite Constultatif International Telephonique it Telegraphiqe (CCITT) has established standards that correspond to RS-232C. While these standards, CCITT V.24 and CCITT V.28, are very similar to RS-232C, they are not identical. The Model 4984 conforms to both RS-232 and CCITT V.24 standards, but does not contain or use all of the circuits allowed for in both standards.

A2.3 RS-422 AND RS-485 STANDARDS

In 1978, the EIA adopted the RS-422 standard to overcome the noise and distance problems associated with the single-ended RS-232 signals. The RS-422 standard specified a differential signal that used two lines per signal.

The RS-422 differential signals have the advantage of higher speed (up to 2Mbs) and longer distance capability (up to 1200M) over the single-ended RS-232 signals. The RS-422 differential signals require a differential receiver and are not referenced to Signal Ground. Differential transmitted signals applied to the interconnecting cable are +2 to +6V or -2 to -6V. Receivers are specified to have a ±0.2V sensitivity, 4Kohm minimum input impedance and be capable of withstanding a maximum input of ±10V. Cable terminators and transmitter wave shaping may be required to minimize cross talk. Logic levels are:

	+2 to +6V	-2 to -6V
Data A/B	0	1
Control A/B	1 (On)	0 (Off)

The differential transmitter output terminal that is positive with respect to the other terminal for the Control On Signal is designated the A terminal. The negative terminal is designated the B terminal. All voltage measurements are made by connecting a voltmeter between the A and B terminals.

A2

RS-485 signals are similar to RS-422 signals except their transmitters are capable of driving up to 32 receivers and their protocol addresses individual devices.

A2.4 RS-530 PINOUTS

In 1987, the EIA released a new standard, EIA-530, for high speed signals on a 25 pin connector. This new standard combined the older RS-232 single-ended signals and the newer RS-422/RS-485 differential signals on one connector. The advantage of the RS-530 specification is that it established a pin out standard for RS-422/RS-485 signals on a 25 pin connector and at the same time provided for the presence of both signals on the same connector.

The 2303 serial interface conforms to the EIA-530 Standard and uses internal jumpers to select the active signal levels on its serial interface. The 2303 is designed so that it may receive either single ended RS-232 or differential RS-422/RS-485 signals.

A2.5 SERIAL INTERFACE PROBLEMS

Most of the problems that arise when connecting serial devices can be avoided if the user will compare the signals on both devices' interfaces before plugging them together. The obvious things to look for are:

1. Verify transmit and receive data direction and pin numbers. DTE devices mate directly with DCE devices while DTE and DTE connections need to be crossed.

2. Check needed control lines. Some devices need signal inputs, others can function with open inputs. All inputs need a valid signal level. If in doubt add jumpers to a known 'on' signal such as the devices's DTR or DSR output signal.

3. Same baud rates. Different baud rates result in garbled data.

   i.e.,*!1-

A2

4. Same character formats. It may be obvious but often the character formats and parity settings are incorrect. A typical parity setting symptom is half good- half bad characters.

i.e.,'1', '2', '4'	good
'3' and '5'	bad

A2

A3 GPIB CONNECTOR/SWITCH BOARDASSEMBLIES

A3.1 BOARD DESCRIPTIONS

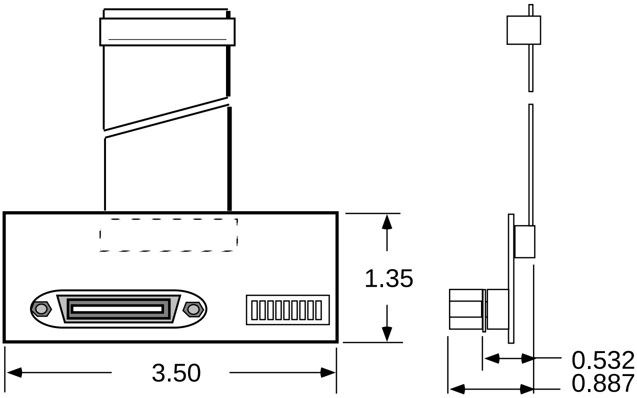
The GPIB Connector/Switch Board Assemblies are small printed circuit boards that provide a convenient way to mount an IEEE-488 Connector and an Address Switch on the rear of the host unit. They connect to the 4803 with a flat ribbon cable that plugs into the GPIB/ Address header (J2).

The GPIB Connector/Switch Board Assemblies are available in two layout styles. The Horizontal Connector/Switch Board Assembly has the Address Switch in line with the IEEE-488 connector as shown in Figure A-9(a). The Vertical Connector/Switch Board Assembly has the Address Switch located on top of the IEEE 488 connector as shown in Figure A-9(b).

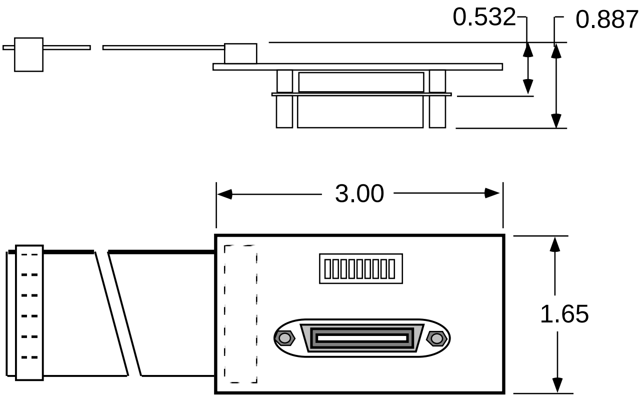
The Address Switch is an eight position rocker switch. For the 4803, the five left most switches set the GPIB address. The bit weights are shown in Figure A-10. Up is a logical 1, down is a logical 0.

The assemblies may be ordered with any length flat ribbon cable, from 10 to 90 cm long. The dash number specifies the cable length. Order as:

Type	**Part Number**
Horizontal Conn./Sw Assy with 90 cm long cable	113640-90
Vertical Conn./Sw Assy with 90 cm long cable	113642-90



(a) Horizontal/Connector Assembly



(b) Vertical Connector/Switch Assembly

Figure A-9 GPIB Connector/Switch Board Assemblies

A3.2 BOARD INSTALLATION

Both the Horizontal and the Vertical Connector/Switch Board Assemblies are designed to be mounted to the rear panel of the host equipment's rear panel by the included metric studs. The following are the suggested installation steps:

1. Select the appropriate cutout from Figure A-11

2. Locate a blank area or the host's rear panel. Leave enough room for the flat ribbon cable bend radius.

3. Machine the cutouts.

4. Install the Connector/Switch Assembly from the inside. Use the metric studs and two thin lock washers to hold the assembly to the panel.

5. Route the flat ribbon cable to the 4803 so it avoids any high RFI or electrical noise area. Plug the cable into J2.

6. Mark or silkscreen the switch functions onto the rear panel as shown in Figure A-10. Identify the 5 address rockers as shown and switch 6 for your application.

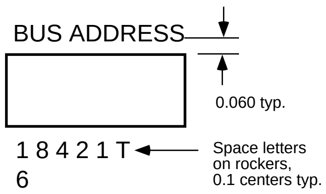
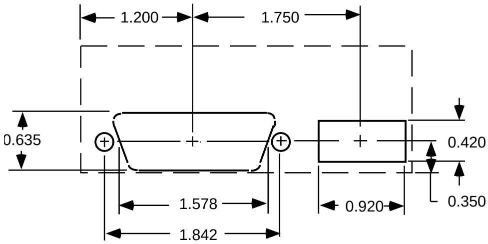


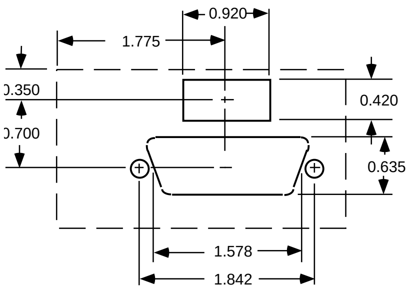
Figure A-10 Switch Silkscreen Detail

A3



Notes:
1. All dimensions are in inches
2. D cutout radius is 0.2 inches
3. Holes are 0.180 dia., 2 plcs
4. Allow 0.25 inches for cable bend

(a) Horizontal Connector/Switch Assembly Mounting Dimensions



Notes:
1. All dimensions are in inches
2. D cutout radius is 0.2 inches
3. Holes are 0.180 dia., 2 plcs
4. Allow 0.25 inches for cable bend radius

(b) Vertical Connector/Switch Assembly Mounting Dimensions

Figure A-11 GPIB Connector/Switch Board Cutouts

Index

Symbols

4809
Description 1-1
Differences 5-4
Factory Jumper Settings 2-20
GPIB Connector/Address Switch Connections 2-13
Installation 2-3
Jumper Settings 2-21
Model Number 1-3
RS-232 Jumper Settings 2-20
RS-422/RS-485 Jumper Settings 2-20
Serial Connections 2-15
4899
Block Diagram 5-2
Description 1-1
Factory Jumper Settings 2-18
Installation 2-2
Jumper Settings 2-19
Model Number 1-3
RS-232 Jumper Settings 2-18
RS-422/RS-485 Jumper Settings 2-19
Serial Connections 2-15

A

Accessories, included 1-13
Address Switch
Connections 2-13
Addressing 3-2
4809 External Address Switch 3-3
Internal Address Setting 2-7
Internal Settings 3-2
Asynchronous data character A-18

B

Block diagram 5-2
Description 5-1

C

CE
Mark 1-11
Certifications/Approvals 1-11
Channel List. *See* SCPI: Channel list
Command and address messages, IEEE 488 A-6
Commands
SCPI A-13
SCPI, example A-14
Configuration
Choices 2-8
Commands 1-7
Methods 2-4

I

Non-PC bus controllers 2-9
Other units 2-8
Program disk, using 2-5
Saving the New Settings 2-8
Configuration Program
Downloaded Program 2-6
Installation 2-5
Running 2-6
Configuring
from other controllers 2-9
Conformance information
488.2 3-10
SCPI 3-13
Connector/Switch Board Assembly A-28
Cable length A-27
Cutouts A-30
Connectors
4809 1-11
4899 1-9

D

Digital Inputs
Monitoring changes 3-7
Dimensions
4899 1-10

E

Enable Registers
Saving Values 3-9
Event Registers 3-4, 3-6

F

Factory
Jumper Settings 1-7
Factory Configuration
4809 Jumpers 2-20
4899 Jumpers 2-18
Command Settings 1-7
Jumper Settings 1-7
Restoring 5-7
Functions
Programmable 1-7

G

GPIB
4809 Address Switch Connections 2-13
Connections 2-11
GPIB Address 2-7
GPIB Bus
Signal-Pin Assignments A-8
GPIB Connector/Switch Board Assembly
Description A-27
Horizontal A-27
Installation A-28
Vertical A-27

H

Horizontal GPIB Connector/Switch Board Assembly A-28

I

IEEE 488
Command and address messages A-6
Message formats (IEEE 488.2) A-12
IEEE 488 Bus Description A-2
IEEE 488.1 A-2–A-3
IEEE 488 Interface
488.1 Capabilities 1-4
488.2 Common commands 1-4, 3-11, 3-12
488.2 Conformance information 3-10
488.2 required status reporting capabilities A-10
488.2 status reporting structure 3-4
Address ranges 1-4

I

Bus Figure A-3
SCPI command structure and examples A-14
SCPI commands A-13
SCPI error reporting A-15
SCPI parser 1-4
IEEE 488.2
Differences from 488.1 3-10
IEEE 488.2 Interface
Buffers 1-4
IEEE 488.2 STANDARD A-9
Common Commands A-12
Differences from 488.1 A-12
Message Formats A-9
Reporting Structure A-9
Indicators, front panel 1-8. *See also* Indicators, front panel
Installation
4809 2-3
4899 2-2
Configuration
Choices 2-8
from non-PC bus controllers 2-9
Other units 2-8
Program disk 2-5
Saving the New Settings 2-8
Connector/Switch Boards A-28
GPIB address 2-7
Rack Mount Kit 2-21
Set-up 2-6
Shipment verification 2-1
Unpacking 2-1
Interface. *See* IEEE 488 Interface

J

Jumper Settings 2-19

L

Locking Setup Parameters 3-24
Long distance transmission using 4894's A-20

M

Maintenance 5-1
Modbus
Error Register 3-6
Generating SRQs 3-25
Querying 3-23
RS-232 Connections 2-15
RS-485 Connections 2-17
RTU Message Format 1-5
Setting Address 3-22
Setting Timeouts 3-24
Writing to a device 3-23
Model Numbers 1-3

N

Non-PC bus controllers 2-9

O

OEM
Copyright wavier 3-27
Documentation 3-27
Operation 3-1
SCPI conformance information 3-13
Theory of 5-1
Operation Registers 3-8
Outline dimensions,
4809 1-10
4899 1-10
Output Queue 3-8

P

Physical
4809 1-11
4899 1-9
Programmable functions 1-7
Programming
Generating SRQs 3-25
Guidelines 3-22
Modbus Device Address 3-22

Querying a Modbus Device 3-23
Saving the setup 3-26
User's IDN Message 3-26

Q

Questionable Event Register 3-7

R

Rack Mount Kit
Instructions 2-21
RAM 5-3
Repair 5-8
RS-232
Specifications 1-6
Standard A-22
RS-485/RS-422
Specifications 1-6
RS-530 pinouts A-25

S

SCPI
ABORt 3-18
Channel list A-15
Examples A-16
Command Reference Table 3-16
Command structure and example A-14
Commands
Example A-14
Commands and queries A-13
Compound commands examples A-15, A-16
Conformance information 3-13
Error reporting A-15
INITiate 3-18
STATus 3-17
SYSTem 3-16
Variables A-15
Self test
Error code list 5-3, 5-4
Errors 5-2
Serial
Communications Background A-17
Serial Interface 1-5
Serial Interface
4899/4809 2-15
Baud rates 1-5
Data character formats 1-5
Problems A-25
Shipment Verification 2-1
Signal-Pin assignments
GPIB bus A-8
Specifications 1-3
RS-232 1-6
RS-485/RS-422 1-6
SRQs
Generating from Modbus Errors 3-25
Standards
RS-232 A-22
RS-422/RS485 A-24
Status Byte Register 3-9
Status Reporting Structure, 488.2 3-4

T

Theory of Operation 5-1
Transmission
Long distance A-20
Troubleshooting 5-2
Guide 5-5, 5-6
Operational Failures 5-2
Self test errors 5-2

U

Unpacking 2-1
User IDN message
Example 3-26

V

Vertical GPIB Connector/Switch Board Assembly A-28