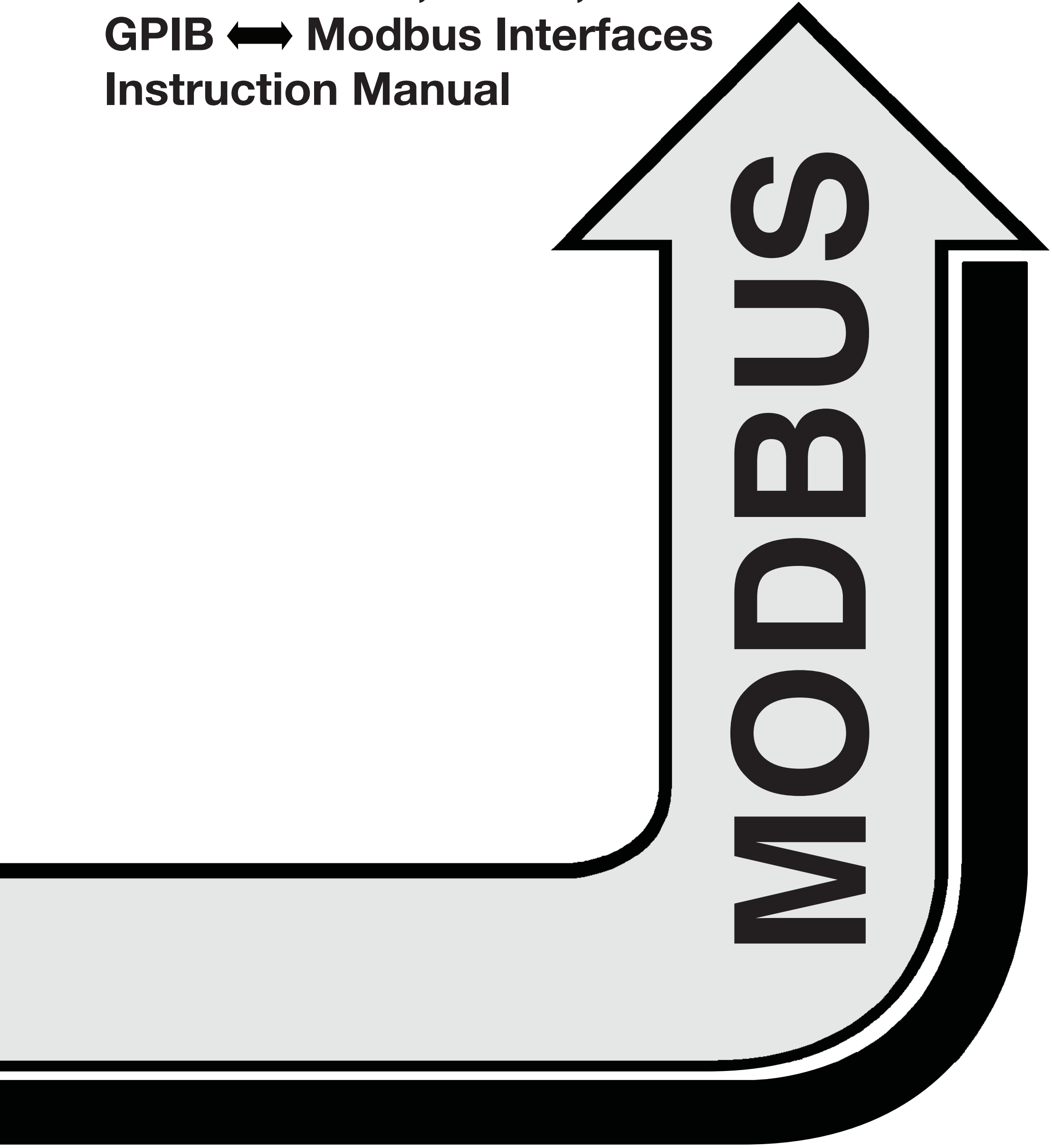
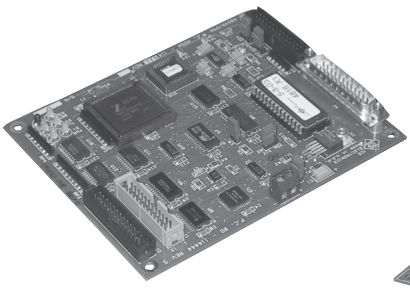
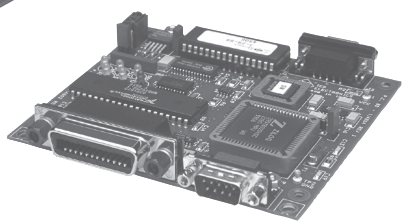
ICS
ELECTRONICS
a division of Systems West Inc.

MODEL 4809A, 4819A, 4829A and 4899A
GPIB ⟷ Modbus Interfaces
Instruction Manual



MODEL 4809A, 4819A, 4829A, 4899A
GPIB ⟷ Modbus Interface
Instruction Manual





ICS
ELECTRONICS
division of Systems West Inc.

7034 Commerce Circle, Pleasanton, CA 94588
Phone 925.416.1000, Fax 925.416.0105
Web Site http://www.icselect.com

Publication Number 120148
April 2017 Edition Rev 19

LIMITED WARRANTY

Within 12 months of delivery, 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 Corporation 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 4899A 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 Models 4809A, 4819A and 4829A boards 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:

VEE is a trademark of Agilent, Palo Alto, CA
LabView is a Trademark of National istruments, Austin, TX
ICS and GPIB AnyWhere are trademarks of ICS Electronics, Pleasanton, CA

Contents

This page left intentionally blank.

General Information

1.1 INTRODUCTION

This section provides the specifications for ICS's Model 4809A, 4819A 4829A and 4899A GPIB<->Modbus Interfaces and their accessory items. Models 4809A, 4819A and 4829A are PC board assemblies designed for mounting inside another piece of equipment. The Model 4899A is an enclosed Minibox™ product designed for bench top use with other equipment. All three products are functionally equivalent. The manual describes the units' functions by referring to the 4899A but, wherever the text refers to the Model 4899A, it also applies to the other models unless otherwise stated. Any model differences are noted in parenthesis or in separate paragraphs.

1.2 DESCRIPTION

The Model 4809A/4819A/4829A/4899A GPIB<->Modbus Interfaces are IEEE-488.2/GPIB to Serial Interfaces with special firmware to control Modbus RTU Slave devices from the GPIB or HP-IB bus. They let the user send simple commands with ASCII values over the GPIB bus to control and query slave Modbus devices. Each interface converts 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 4899A is next addressed to talk. The 4809A and 4899A handle both RS-232 and RS-485 Modbus devices. The 4819A handles only RS-232 devices while the 4829A handles only RS-422/RS-485 devices.

The units contain a number of advanced features that increase their flexibility and simplifies their use in system applications. All have IEEE-488.2

1

compatible interfaces with an expanded Status Reporting Structure that complies 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 4899A 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 4899A accepts a wide range of DC voltages and is shipped with an adapter for the local power lines. The 4899A can be used as a stand-alone box or rack mounted inside a standard 19-inch wide equipment rack.

The Model 4809A is a small, low-profile PC assembly designed for mounting inside the host chassis. It has two flat-ribbon connectors for its GPIB signals. One connector can be connected to a panel-mounted GPIB connector on a pin-to-pin basis. The second connector mates to one of ICS's GPIB Connector/Address Switch Assemblies which mount a GPIB connector and an GPIB address switch on the rear panel of the host chassis. ICS's GPIB Connector/Address Switch Assemblies are available in two mounting styles as described in Appendix 3. The RS-232 and RS-422 serial signals are available on a flat-ribbon header and on a DB-25P connector. The 4809A has the same status LEDs as the 4899A and runs on 5 to 12 volt power.

The Model 4819A and 4829A are small PC assemblies with GPIB and RS-232 serial interfaces. Both are designed to be mounted to the rear panel of the host chassis so the GPIB and external serial connectors protrude through the rear panel. Both interfaces can be used to control the internal Modbus device. The 4819A ORs the external serial input and the serial Modbus packets generated by GPIB bus commands into a single-ended RS-232 signals to drive the internal Modbus device. Modbus device responses are available at either interface. The 4819A has the same diagnostic LEDs as the 4899A and runs on 5 to 12 volt power. The 4829A provides differential RS-422/RS-485 signals to drive multiple Modbus slave devices. The 4829A also includes the RS-232 to RS-485 conversion function so external RS-232 signals can be used to control the internal RS-422/RS-485 Modbus device.

1.3 MODEL 4809A/4819A/4829A/4899A 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.

4899A - 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

4899A - Minibox Product

4809A - Board Product
114922 - 4809A board only version

4819A - Board Product
115122 - 4819A board only version

4829A - Board Product
115782 - 4829A board only version

Note options -3, -5, -9, and the AC adapters do not apply to the Model 4809A or 4819A and to their board only versions.

The 48x9A version interfaces have additional floating point Modbus commands not available in the original 48x9 interfaces. The 48x9A series units are fully compatible with the original 48x9 interfaces.

1

1.4 IEEE 488 INTERFACE

1.4.1 488.1 Capabilities

The 488 Bus interface meets the IEEE STD 488.1-19A87 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

The 4899A conforms to IEEE STD 488.2-19A87. 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

The extended SCPI parser complies with the SCPI Standard Version 1994.0.

1.5 SERIAL MODBUS INTERFACE

The 4899A and 4809A's serial Modbus interface provides RS-232 single-ended and RS-485 (RS-422) differential signals. Signals are selected by internal jumpers. The 4899A has a DB-25S connector on its rear panel, the 4809A has a 26-pin flat-ribbon header and a DB-25P connector. Signal pinouts conform to EIA RS-530 specification and are listed in Table 2-2.

The 4819A provides RS-232 single-ended signals on a DE-9S connector. DCE/DTE jumpers let user exchange TX and RX signal pin numbers. Signal pinouts are listed in Table 2-3.

The 4829A provides RS-422/RS-485 differential signal pairs on a DE-9S connector. On-board jumpers let user connect the TX and RX signal pairs together for 2-wire networks. The RX pair includes a termination network for biasing 2-wire networks. Signal pinouts are listed in Table 2-4.

1.5.1 Modbus RTU Messages

Messages conform to the Modbus RTU format and include the device address, command, register number, data and CRC formatted as binary bytes. Supported Modbus commands are: 01, 02, 03, 04, 05, 06, 07, 08, and 16 for integer values and commands 03 and 16 for floating point 32-bit values.

Integer range	16 bits or 0 to 65,536
Floating point	IEEE-754

1.5.2 Baud Rates:

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 and 115200 baud.

Baud Rates:	4899A - Any rate from 50 to 115,200 baud.
	4809A - Any rate from 50 to 38,400 baud.
	4819A - Any rate from 50 to 57,600 baud.
	4829A - Any rate from 50 to 57,600 baud.

1.5.3 Data Character Formats:

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

1.5.4 RS-232 Specifications

Single-ended RS-232C drivers and receivers that are designed to operate with up to 50 feet of cable.

Transmit	+9 Vdc = Logic "0" or On
Levels	-9 Vdc = Logic "1" or Off
Receive	±1.5 Vdc minimum, ±25 Vdc Maximum
Signals	
4899A	AA, AB, BA, BB, CA, CB, CD and CF
4809A	
4819A	AA, AB, BA and BB
	CC held high, CA jumpered back to CB

1.5.5 RS-422/RS-485 Specifications

The 4809A, 4829A and 4899A have differential RS-485 line drivers and receivers that provide RS-422 and RS-485 compatible signals. 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 for 4-wire systems or it can be tristated when not transmitting for 2-wire systems.

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

1.6 PROGRAMMABLE FUNCTIONS

All units 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. The 4899A and 4809A are factory set for RS-232 signals. The 4829A is factory set for 4-wire operation.

TABLE 1-1 FACTORY CONFIGURATION

Command	Functions	Factory Setting
:ADDRess	Sets GPIB bus address	4
:EXT	Enables External GPIB Address Switch	OFF (4809A 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

Notes: # indicates a parameter that can be blocked by the LOCK command

1.7 INDICATORS

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 the RQS bit is set.
ERR	-	On when ESR bits 2 thru 6 are set on.

When the 4899A 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 4899A displays its current GPIB address by blinking the front panel LEDs for one-half second. The GPIB address is the sum of the binary bit weights. 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 selftest errors and their possible causes.

During normal operation, the ERR LED indicates that there was a problem with the GPIB command or with the Modbus communication. If the ERR LED illuminates at power turn-on, it indicates that the unit has not been calibrated since the memory was last initialized. Refer to paragraph 5.3.1 for directions on calibrating the unit.

1.8 PHYSICAL

1.8.1 4899A Minibox

Size	7.45" L x 5.57" W x 1.52" H (18.92 cm L x 14.15 cm W x 3.86 cm H) (See Figure 1-1)
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-25S female connector with lock studs

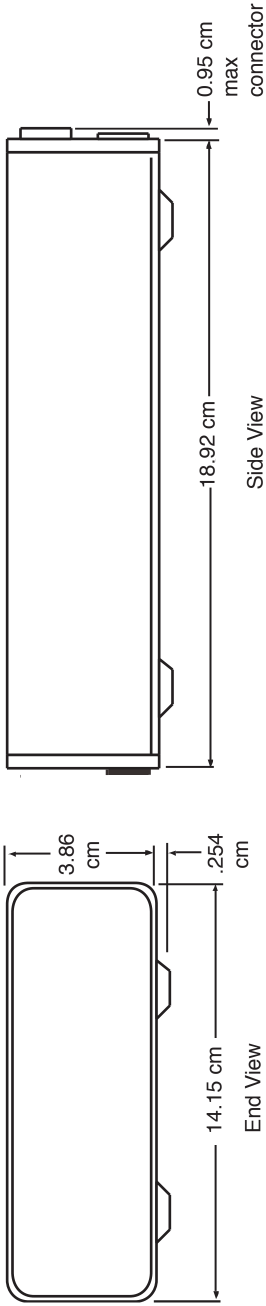


Figure 1-1 4899A Outline Dimensions

1.8.2 4809A Board Assembly

Size	5.50" L x 4.50" W x 0.5" H (139.7 mm L x 114.3 mm W x 12.7 mm H) (See Figure 1-2)
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 @ 300 MA (regulated) 5.5 to 12 Vdc (unregulated)
Connectors	GPIB - 24-pin 3M 2524 male header GPIB/Address Sw - 26-pin 3M 2526 male header Serial - 26-pin 3M 2526 male header and DB-25P male connector.

1

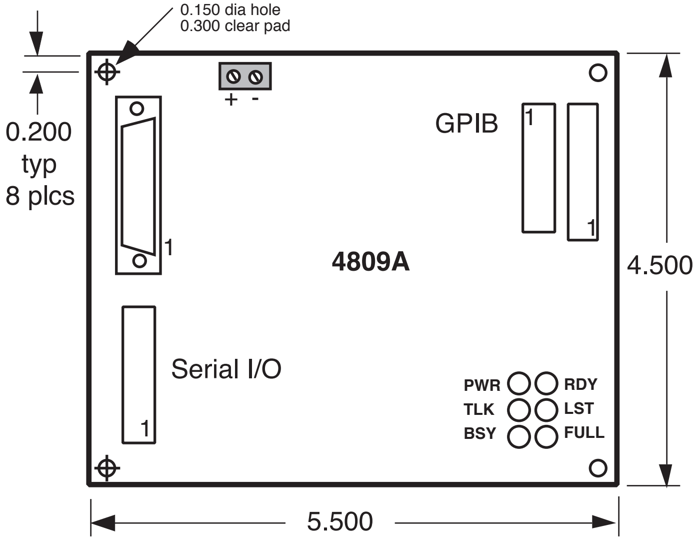


Figure 1-2 4809A Outline Dimensions

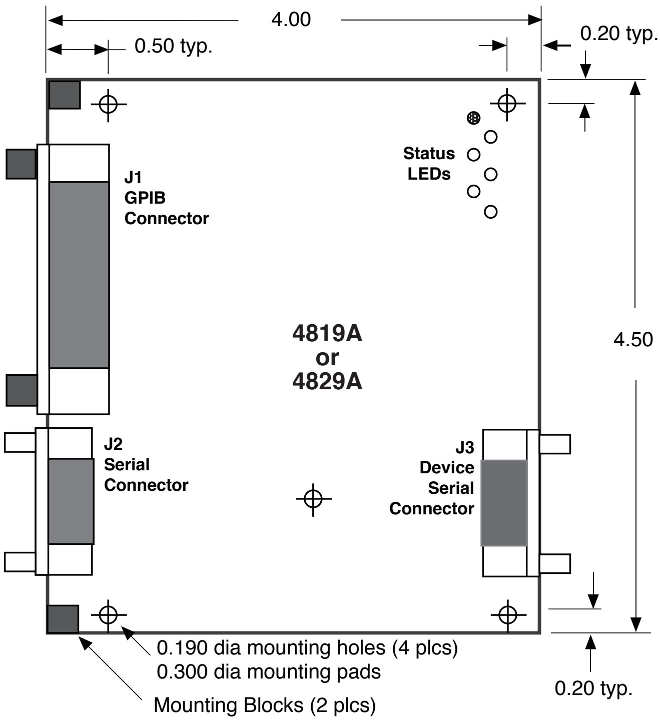


Figure 1-3 4819A/4829A Outline Dimensions

1.8.3 4819A and 4829A PHYSICAL

Size	4.00" L x 4.50" W x 0.7" H (10.16 cm L x 11.43 cm W x 1.78 cm H) (See Figure 1-3)	
Weight	8 oz (0.22 kg)	
Temperature	Operating Storage	-10 °C to +55 °C -20 °C to +70 °C
Humidity	0-90% RH without condensation	
Shock/Vibration	Normal handling only	
Construction	Flame-retardant printed circuit board. Connector shells grounded to chassis supports and mounting brackets.	
Power	5 ± 0.25 Vdc @ 300 MA (regulated) 5.5 to 12 Vdc (unregulated)	
Connectors	GPIB - 24-pin IEEE-488 connector with metric studs External RS-232 Interface - DE-9S male connector with lock studs Internal Serial Interface - DE-9P female connector with lock studs	

1.9 4899A 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

Signature

Date Sept.14, 1999

Gerald K. Mercola
(Full Name)

President
Title

Figure 1-4 4899A Certificate of Conformance

1

1.10 ACCESSORIES

1.10.1 4899A Included Accessories

120117	4809A/4819A/4829A/4899A Instruction Manual
A/R	AC Wall Adapter, with applicable plug
123021	Support CD-ROM

1.10.2 4809A/4819A/4829A Included Accessories

120117	4809A/4819A/4829A/4899A Instruction Manual
123021	Support CD-ROM

1.10.3 Optional Accessories

120117 4809A/4819A/4829A/4899A Instruction Manual

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 for 4809A. (Dash number is cable length from 10 to 90 CM long. 90 CM standard)

114256-L Serial Flat Ribbon Extension Cable from 4809A 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 4809A, 4819A, 4829A or 4899A GPIB<->Modbus Interface
(1) AC Power Adapter (Model 4899A only)
(1) 4809A/4819A/4829A/4899A Instruction Manual
(1) Support CD-ROM

Board only orders 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 INSTALLATION GUIDES

2.3.1 4899A Installation Guide

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

2

1. If the 4899A is to be used with RS-422 or RS-485 signals, change the jumper settings as directed in Section 2.11.

2. See Section 2.10 to select and/or design the serial cable.
CAUTION - Check 'standard' serial cable wiring carefully as may standard cables may not make proper connections to the Modbus RTU Controller.

3. If the unit is to go into a rack mounting kit, disconnect all cables from the unit. Follow the instructions in Section 2.12 to install the unit it in the rack mounting kit.

4. Connect the AC adapter to the 4899A and to the AC power. Turn the unit on and verify that it passes its selftest and that it indicates the correct GPIB address. Use the *IDN? query to verify GPIB communication.

5. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured. If the 4899A needs to be reconfigured, follow the instructions in Sections 2.4 - 2.8 to change its configuration. Save the new settings.

6. Use a GPIB keyboard or similar interactive GPIB control program to query the internal modbus device's model number or some other known value following the examples in section 3.8. A typical message would be 'R? 0,1' to query its model number. Read the response to confirm that it is the expected response. Try out other Modbus commands that apply to the Modbus device.

2.3.2 4809A Installation Guide

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

1. If the 4809A is to be used with RS-422 or RS-485 signals, change the jumper settings as directed in Section 2.11.

2. Review Sections 2.8 and 2.10 to select and/or design the GPIB and serial interface cables.

3. Select a convenient location to mount the 4809A. Do not mount it directly over a heat producing surface. Provide a 0.1 inch (2.5 mm) clearance underneath the 4809A or use an insulator if the 4809A is being mounted on a metal surface. Use the dimensions in Figure 1-3 to establish the mounting hole locations.

4. Use a twisted pair of #24 wires to connect the 4809A's power terminals to the host's power supply. Connect the 4809A directly to the power supply to avoid noise problems. Set jumper W1 to REG for unregulated 5.5-12 volt power, to P1 for regulated 5 volt power.

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. Use the *IDN? query to verify GPIB communication.

6. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured. If the 4809A needs to be reconfigured, follow the instructions in Sections 2.4 - 2.8 to change its configuration. Save the new settings.

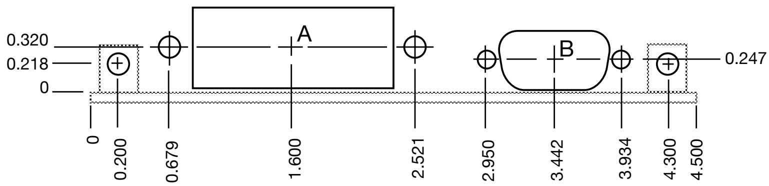
7. Use a GPIB keyboard or similar interactive GPIB control program to query the internal modbus device's model number or some other known value following the examples in section 3.8. A typical message would be 'R? 0,1' to query its model number. Read the response to confirm that it is the expected response. Try out other Modbus commands that apply to the Modbus device.

2.3.3 4819A/4829A Installation Guide

The following steps should be used as a guide to the 4819A or 4829A installation.

1. Review Section 2.10 to select and/or design the serial interface cables.

2. Select a spot on the rear panel to mount the board. The 4819A/4829A mounts perpendicular to the rear panel and is held in place with mounting blocks. Use the cutouts and mounting dimensions shown in Figure 2-1 for the connector cutouts. Cutout 'A' is for the IEEE-488/GPIB connector. Cutout 'B' is for the DE-9 serial connector. Do not mount the board directly over a heat emitting surface. Provide a 0.1 inch (2.5 mm) clearance underneath the board or use an insulator if the board is being mounted on a metal surface.



Notes: 1. Grayed lines are PC board and mounting blocks
2. Mounting holes are 0.150 dia thru
3. Cutout A is 1.575 in x 0.625 in with 0.150 dia holes
4. Cutout B is 0.875 in x 0.450 in 'D' shell with 10° slope to bottom and with 0.125 dia holes

Download a mounting template from http://www.icselect.com/pdfs/4819Alayout.pdf.

Figure 2-1 4819A and 4829A Rear Panel Cutouts

3. To minimize EMI/RFI and to maximize electrical immunity, use the mounting blocks to fasten the board to the rear panel of the host chassis. The connector shells should fit against the rear panel. Use the recommended cutouts in Figure 2-1 to overlap the connectors. De burr the panel after machining and remove any insulating finish that would prevent the board from making good chassis connections.

4. Use a twisted pair of #24 wires to connect external DC power to the screw terminals at P1. Connect the board directly to the power

supply to avoid noise problems. Set jumper W2 to REG for un-regulated 5.5 -12 volt power, to P1 for regulated 5 volt power.

5. Plug the cable from the internal modbus device into J3 and connect the board to a GPIB controller. Turn the unit on and verify that it passes its selftest and blinks its correct GPIB address. Use the *IDN? query to verify GPIB communication with the board.

6. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured before it can send data to the serial device. If the board needs to be reconfigured, follow the instructions in Sections 2.4 - 2.8 to change its settings. Save the new values.

7. Use a GPIB keyboard or similar interactive GPIB control program to query the internal modbus device's model number or some other known value following the examples in section 3.8. A typical message would be 'R? 0,1' to query its model number. Read the response to confirm that it is the expected response. Try out other Modbus commands that apply to the Modbus device.

2

2.4 CONFIGURATION DIRECTIONS

When shipped, the 4809A, 4819A, 4829A or 4899A are set to the factory configuration listed in Table 1-1. The configurable parameters are stored in Flash (EEPROM memory on older boards) and can be queried and changed by the user.

Configuring a 4899A is a three step process. First design the link between the unit and the serial device. This will provide you with the serial configuration values. Secondly, review the SCPI commands and programming guide in Section 3 to select the configuration settings. Then follow the instructions in 2.6 or 2.7 to configure the unit.

If you are using a PC as a bus controller, the easiest way to configure the unit is to use ICS's Minibox configuration program, *niconf_w.exe*, which guides the user through a menu-driven configuration procedure. *niconf_w.exe* runs with any ICS, National Instruments, MCC or other GPIB Controller card that has a GPIB-32.DLL and uses the NI command set. Follow the instructions in Section 2.6 when using the *niconf_w.exe* configuration program.

Because these units have only a few configurable parameters, they can be easily configured by entering the SCPI commands directly into an interactive command line utility supplied by your GPIB Controller Card manufacturer. You can also use ICS's GPIBkybd program with any Adlink, ICS, Keithley, MCC or National Instrument GPIB Controller card. Follow the instructions in Section 2.7 when using one of these interactive command line programs.

2.5 THE SUPPORT CD-ROM

The Support CD-ROM contains Configuration Programs, Utility Programs, and Example Programs for ICS's interface products plus backup Manuals Application Notes and reference materials.

When you load the CD into the CD-ROM drive, it should automatically run and display the Support CD Selection Window. Click Configure GPIB Interfaces to run the *niconf_w.exe* program or Install Keyboard Utilities to install the GPIBkybd program. You can always download *niconf_w.exe* from the Support CD and save it on your hard drive and later run it directly from our hard drive.

CAUTION
Do not click on Install 488.2V3 Driver as that may damage your existing GPIB Driver Installation.

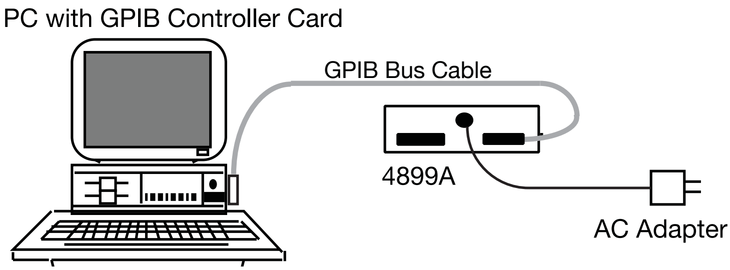
Use the Browse CD or View Readme File to examine the CD contents.

2.6 RUNNING ICS'S CONFIGURATION PROGRAM

The configuration program supports the standard configurable items. Special settings such as the user's IDN message will have to be entered with a live keyboard program (such as ICS's GPIBkybd program) or as part of the user's program (See section 3.8.6).

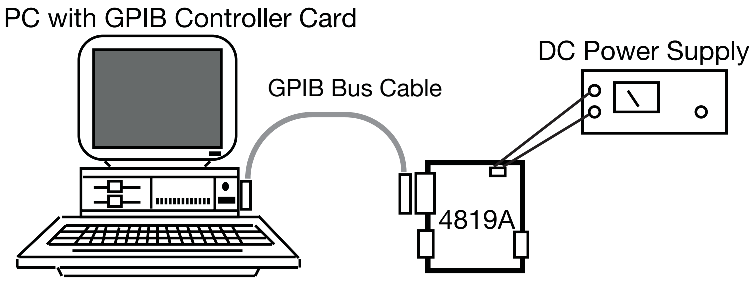
2

1. Connect the 4899A to the GPIB controller card in the PC or to a USB-GPIB Controller as shown in Figure 2-2(a). Connect a DC power supply to the unit.



(a) 4899A Configuration Setup

Connect the 4809A, 4819A or 4829A to the GPIB controller card in the PC or to a USB-GPIB Controller as shown in Figure 2-2(b). Connect a DC power supply to the power terminal strip. Set jumper W2 to the P1 position. Set the power supply to 5 ± 0.2 Vdc.



(b) 4809A , 4819A or 4829A Configuration Setup

Figure 2-2 Configuration Setups

2. Apply power to the unit. After 0.5 seconds, 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

4899A <return> 'selects Model 4899A

The program may ask that you turn the unit off and back on. Press the Continue 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 and click the SET button. 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

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 Table 3-3 to understand the command choices and their affect on the unit’s operation. Setting changes are made by entering a value in the box above the Enter/Accept button and then clicking the button. The program will make the change and display the new setting. Settings are accepted by clicking the Enter/Accept button which advances you to the next parameter and does not change the current setting.

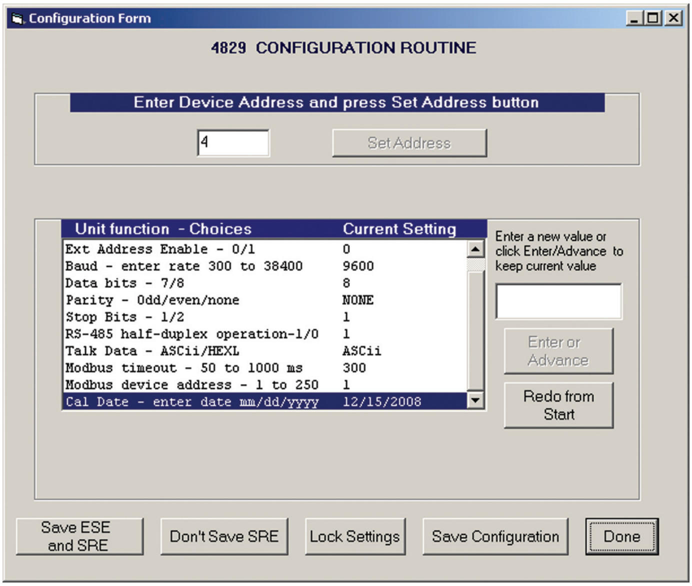


Figure 2-3 Configuration Program shown prior to saving the new settings

7. Saving the New Settings

After the last parameter has been accepted, 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 WITH AN INTERACTIVE PROGRAM

2.7.1 General Configuration Instructions

The 4899A can be easily configured from any GPIB bus controller with an interactive program such as ICS's GPIBkybd program by using the following procedure.

1. Connect the unit to the GPIB controller as shown in Figure 2-2. Run the program and if required 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 LEDs.

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

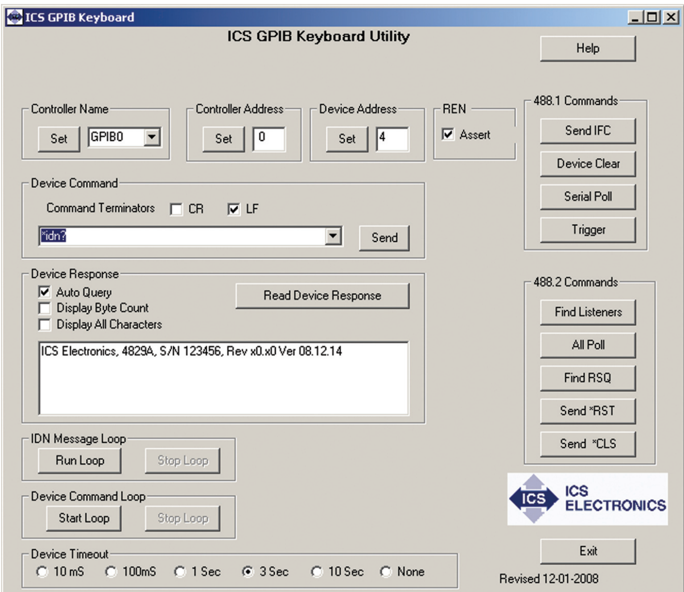


Figure 2-4 GPIBkybd Program with 4829A's IDN response

2

3. Verify communication to the unit by sending it the *IDN? query and reading back the unit's IDN message. GPIBkybd users can use the pulldown arrow on the right side of the Device Command window and select *IDN?.

4. Use the SCPI Command tree in Table 3-2 to put together the SCPI command for the parameter you want to change. (See paragraph 3.6 if you need help using SCPI commands.) Enter the commands into your interactive program (GPIBkybd Device Command Window) and press "Send". If you are querying a value press the "Query" if the program has that choice. If not, you will have to manually "Read" the response. ICS's GPIBkybd program has an auto query function so it will automatically read back a response if you just press "Send". The following example shows how to change and query the baud rate.

SYST:COMM:SER:BAUD 19200	‘sets the baud rate
SYST:COMM:SER:BAUD?	‘queries the baud rate
	‘should reply 19200

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

SYST:COMM:GPIB:ADDR 20	‘ send address 20
	‘Wait 0.1 seconds

   Change the device's GPIB address in the interactive program to the new address, in this case 20. Query the device to be sure the address changed.

SYST:COMM:GPIB:ADDR?	‘should read back 20

6. Use the ***SAV 0** command (That is asterisk S A V space 0) to save the new address values in the unit's nonvolatile memory. The ***SAV 0** command stores the current I/O configuration settings as the power-on values.

   ***SAV 0**

2.8 GPIB CONNECTIONS

2.8.1 4819A, 4829A and 4899A GPIB Connections

The 4819A, 4829A and 4899A have standard 24-pin IEEE-488 connector with metric lock studs. Their IEEE-488 connector mates with all standard IEEE 488/GPIB bus cables. Signal-pin assignments for the standard IEEE-488 connector are shown in Figure A-2 in the Appendix.**2.9**

2.8.2 4809A GPIB Connections

The 4809A has two male flat-ribbon male connectors that can be used to connect the 4809A to the GPIB bus. Connector J1 is a 24-pin connector that is designed for pin-to-pin connection to the GPIB. Connector J2 is a 26-pin connector that contains the address switch input signals and the GPIB bus signals. The 4809A only requires that one of the connectors be used to connect it to the GPIB bus. The unused connector should be left open.

2.8.2.1 GPIB Connector J1

The GPIB Signal-pin assignments for J1 are identical to a standard IEEE-488 connector as shown in Figure A-2 in the Appendix. The J1 connector layout is shown in Figure 2-5. Use a flat ribbon cable like ICS's 114439-L cable with a 24-pin plug on one end and a GPIB connector on the other end to connect J1 to the GPIB bus. Cut an opening for the GPIB connector on the rear panel (See cutout A and holes C in Figure 2-1). Mount the GPIB connector on the rear panel of the chassis with the metric lockstuds. Fabricate a flat ribbon cable as shown in Figure 2-6 or purchase it from ICS Electronics as P/N 114439-L where L is the cable length in cm.

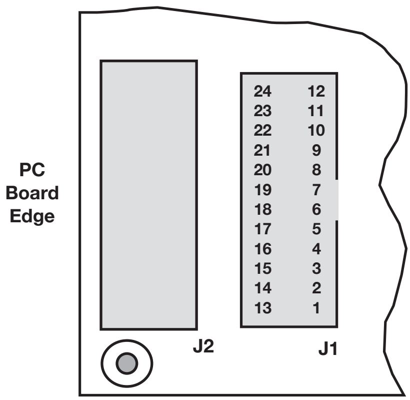


Figure 2-5
4809A J1 GPIB Connector-Pin Layout

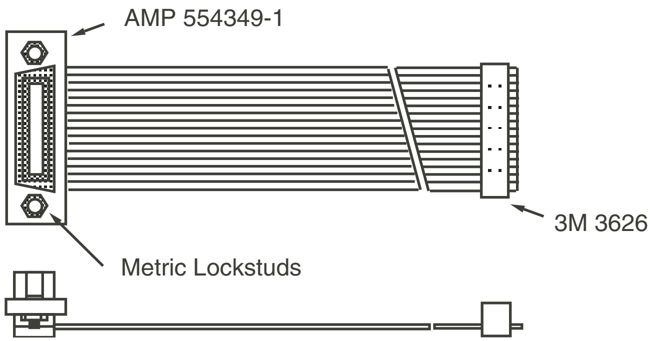


Figure 2-6 114439 Flat Ribbon Cable Assembly

2.8.1.2 GPIB/Address Switch Connector J2

Connector J2 is a 26-pin connector that contains the GPIB external address switch signals as well as the GPIB bus signals. J2 is used when you want to use an external GPIB address switch with the 4809A. The J2 connector layout is shown in Figure 2-8 and the signal-pin assignments are listed in Table 2-1. The external address switch inputs are low true signals with pullup resistors on the 4809A. At power turn-on, the 4809A 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 4809A. The assemblies are available in two layout styles. Refer to Appendix A3 for dimensions, installation instructions and silkscreen. Switch layout and rocker functions are shown in Figure 2.7. Note that rockers 7 and 8 are not used by the 4809A.

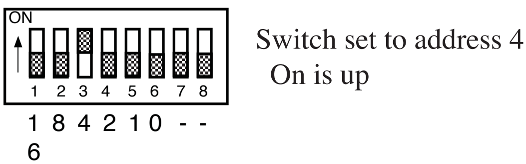


Figure 2-7 Address Switch Rocker Assignments

TABLE 2-1 4809A 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	not used
ADSW1	7	VIO 1	1 (LSB)
ADSW3	8	GRY 1	4
ADSW2	9	WHT 1	2
NRFD	10	BLK 1	GPIB Signals
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 Signals

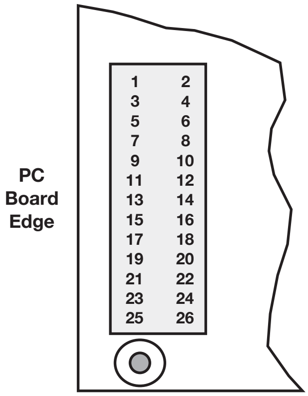


Figure 2-8 4809A GPIB J2Connector Pin Layouts

2

2.9.1 4809A Serial Connections

The 4809A's serial port is a DTE (Data Terminal Equipment) interface on a 26-pin male ribbon connector (J3) and a DB-25P connector (J4). The 26-pin male ribbon connector is for compatibility with the earlier 4809's. The DB-25P connector allows standard DB-25S connectors to be used with the 4809A. RS-232 and RS-422 (RS-485) signal selection is made by setting jumpers on the 4809A. Refer to section 2.13 for jumper setting instructions. Table 2-2 shows the 4809A's signal-pin assignments and signal directions. Figure 2-9 shows the pin layout for connector J3.

2.9.1.1 4809A LED Outputs

The 4809A's TALK and LSTN LED driver outputs are provided on pins 17 and 21 on connector J3 and J4 to drive remote LEDs. Each signal is low true and can sink 3 mA. Use a resistor in series with 5 Vdc to limit the current to LED. Connect the LED cathodes to the driver signals. The recommended resistor value is 1500 ohms for 5 Vdc. Connect the 5 volt return line to pin 7.

2.9.2 4819A Serial Connections

The 4819A has two right-angle 9-pin connectors for its RS-232 interfaces. Connector J2 is the external serial connector that protrudes through the rear panel. Connector J3 is the internal serial connector for connecting the 4819A to the serial device. Signal pin assignments for both connectors are shown in Table 2-3.

2.9.2.1 4819A Rear Panel Connector J2

Rear panel connector J2 is a DE-9P connector with DTE type signal assignments similar to the signals in a PC COM port. Signal TxD on pin 2 is the output signal and RxD on pin 3 is the input signal. RTS and DTR are pulled high to the 'ON' state. DTR is internally jumpered to CTS.

TABLE 2-2 4809A/4899A Serial Connector Signal-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			4809A Talk LED Dvr-	→
18				
19		RS(B)	Request-to-send (B)	→
20	CD	TR(A)	Data Terminal Rdy (A)	→
21			4809A Listen LED Dvr -	→
22				
23		TR(B)	Data Terminal Rdy (B)	→
24				
25				
26	Note: Pin 26 is only available on the 4809A			

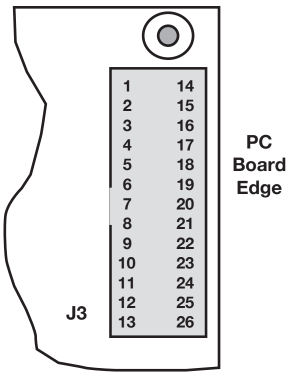


Figure 2-9 4809A J3 Serial Connector Pin Layout

TABLE 2-3 4819A SIGNAL ASSIGNMENTS

Rear Panel Connector J2			Internal Connector J3	
Pin #	Signals DE-9P	Signal Direction	Signals DE-9S	Pin #
1	DCD		+V DCD	1
2	TxD	←	TxD *	2
3	RxD	→	RxD *	3
4	DTR +V		DTR	4
5	GND	–	GND	5
6	DSR		DSR	6
7	RTS +V		RTS	7
8	CTS		CTS	8
9	nc		– nc	9

Notes: 1. +V is + 9 Vdc through a 3.3 kohm resistor.
2. * Standard setting. TxD and RxD can be switched by setting jumper W4 to the DTE position.

2.9.2.2 4819A Internal Connector J3

Internal connector J3 is a DE-9S connector with DCE type signal assignments. Signal RxD on pin 3 is the output to the internal serial device. Signal TxD on pin 2 is driven by the internal serial device. The RxD and TxD pins can be exchanged by rotating jumpers W4 90 degrees. DCD and DSR are pulled up to an 'ON' state and RTS is jumpered back to CTS.

2.9.3 4829A Serial Connections

The 4829A has two right-angle 9-pin connectors for its serial interfaces. Connector J2 is the external RS-232 serial connector that protrudes through the rear panel. Connector J3 is the internal RS-485 serial connector for connecting the 4829A to the serial device. Signal pin assignments for both connectors are shown in Table 2-4.

2.9.3.1 4829A Rear Panel Connector J2

Rear panel connector J2 is a DE-9P connector with DTE type signal assignments similar to the signals in a PC COM port. Signal TxD on pin 2 is the output signal and RxD on pin 3 is the input signal. RTS and DTR are pulled high to the 'ON' state. DTR is internally jumpered to CTS.

TABLE 2-4 4829A SIGNAL ASSIGNMENTS

Rear Panel Connector J2			Internal Connector J3	
Pin #	Signals DE-9P	Signal Direction	Signals DE-9S	Pin #
1	DCD		RD(B)	1
2	TxD	←	RD(A)	2
3	RxD	→	SD(A)	3
4	DTR +V		SD(B)	4
5	GND	–	GND	5
6	DSR		nc	6
7	RTS +V		nc	7
8	CTS		nc	8
9	nc	–	nc	9

Notes: 1. +V is + 9 Vdc through a 3.3 kohm resistor.
2. * Standard setting. TxD and RxD can be switched by setting jumper W4 to the DTE position.

2.9.3.2 4829A Internal Connector J3

Internal connector J3 is a DE-9S connector with SD and RD differential signal pairs. The SD and RD signal pairs are a four-wire, full-duplex interface. The SD and RD signal pairs can be jumpered together on the 4829A board for half-duplex operation by installing jumpers W4 and W5.

2.9.4 4899A Serial Connections

The 4899A's serial port is a DTE (Data Terminal Equipment) interface on a DB-25S female connector. The connector has both RS-232 and RS-422/RS-485 signals in accordance with EIA-STD - RS-530. RS-232 and RS-422 (RS-485) signal selection is made by setting jumpers inside the 4899A. Refer to section 2.10 for jumper setting instructions. Table 2-2 shows the 4899A's signal-pin assignments and signal directions.

2.10 Modbus Device Connections

2.10.1 RS-232 Connections to a Modbus Device

The minimum RS-232 connection uses just three lines to connect the unit to a Modbus slave device. The lines are transmit data (TxD), receive data (RxD), and Ground. The following examples show how the 4899A, 4809A and 4819A are wired to some typical RS-232 Temperature Controllers. Refer to Table 5-2 for troubleshooting Modbus Device communication problems.

Figure 2-10 shows the 4899A or 4809A RS-232 connection to a Watlow F4 Temperature Controller. Both interfaces transmit on pin 2 and receive on pin 3.

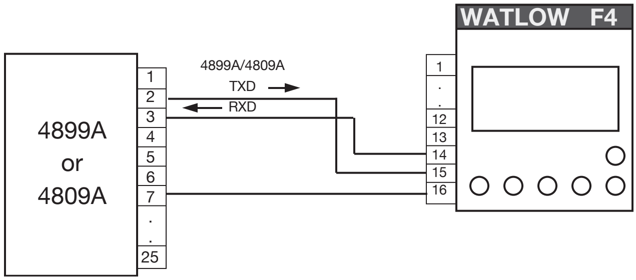


Figure 2-10 4899A/4809A RS-232 Connections to a Watlow F4

Figure 2-11 shows the 4819A's RS-232 connection to a Watlow F4 Temperature Controller. The 4819A's transmit signal is on pin 3 and its receive input signal is on pin 2 with the DCE/DTE jumpers W4 set in the DCE position. Switch jumper W4 to the DTE direction to swap the signal pin assignments.

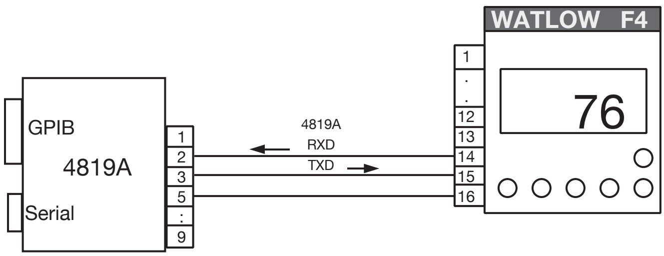


Figure 2-11 4819A RS-232 Connections to a Watlow F4

Figure 2-12 shows the 4819A's RS-232 connection to a Watlow series 96 Temperature Controller.

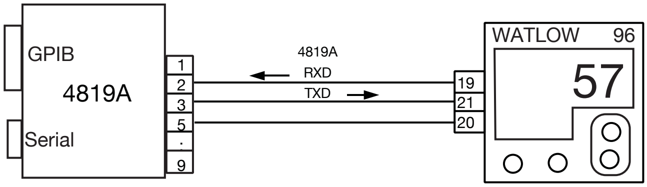


Figure 2-12 4819A RS-232 Connections to a Watlow 96

2.10.2 RS-485 Connections to a Modbus Device

The 4809A, 4829A and 4899A's serial interface provides a transmit (SD) and a receive (RD) pair of RS-422/RS-485 signals. Because most RS-485 Modbus networks are two wire, half-duplex networks, the SD and RD signal pairs have to be jumpered together in the cable connector. The 4809A, 4829A and 4899A have to be configured for RS-485 operation when used on a two wire RS-485 network. Use the SYST:COMM:SER:RS485 ON command to configure the units. The ON setting 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 termination networks to bias the lines in the 'mark' state when neither unit is transmitting. The termination networks are made up of a pullup, a pulldown and a load resistor. The termination networks prevent each receiver from inputting noise when nothing is being transmitted. Use one termination network for short cables of 200 feet or less. For longer cables, use a termination network at each end of the cable. Set the bias voltages to approximately 2 Vdc and 2.5 Vdc. Use resistors with an approximate value of 500 ohms/volt.

Figure 2-13 shows a method of making the RS-485 connection to a 4899A or 4809A. Figure 2-13 uses the 4899A/4809A's internal pullup resistor available at pin 20 to pullup the SD(B) and RD(B) lines. A 1 kohm resistor from SD(A) and RD(A) to ground is the pulldown resistor. The 220 ohm load resistor completes the circuit. The 4829A contains an internal termination network on the RD signal pair and does not need external resistors.

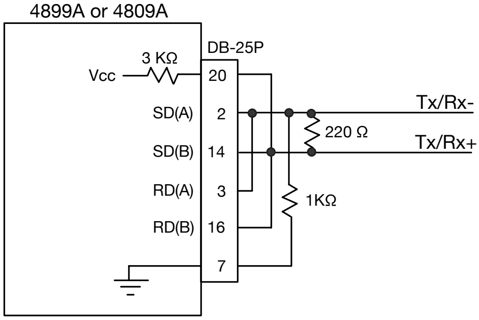


Figure 2-13 4899A/4809A Pullup/Pulldown Resistor Connections

Figure 2-14 shows an example of a single 4899A or 4809A driving two Modbus devices over a RS-485 network. In Figure 2-12, the termination network uses 5 Vdc and ground provided by the Watlow F4 Temperature Controllers.

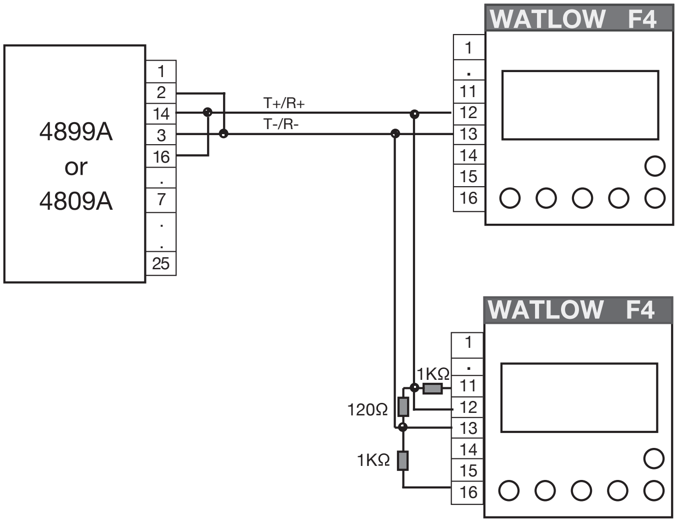


Figure 2-14 RS-485 Network connections to Watlow F4 Controllers

Figure 2-15 shows an example of a single 4899A or 4809A driving a Watlow EZ Zone Controller with a simple point-to-point RS-485 network. The EZ Zone controller does not have easy terminals to attach a termination network so the network connections shown in Figure 11 are reused here.

The 4899A/4809A's internal pullup resistor at pin 20 is used to pullup the SD(B) and RD(B) lines and apply a positive signal to the Tx/Rx- line. A 1 kohm resistor from SD(A) and RD(A) to ground is the pulldown resistor for the Tx/Rx+ line. The 220 ohm load resistor completes the circuit.

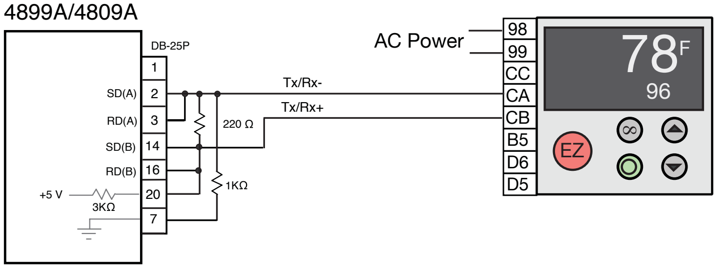


Figure 2-15 RS-485 Network connections to a Watlow EZ Zone Controller

Figure 16 shows an 4829A driving a single Watlow EZ Zone Controller with it RS-485 output. The 4829A has an internal termination network with a 220 ohm load resistor so it does not need external resistors. If the internal W4 and W5 jumppers are installed (the gray links in Figure 2-16), then no external jumpers are required and the EXZ Zone Controler can be connected to either (A)-(B) signal pair.

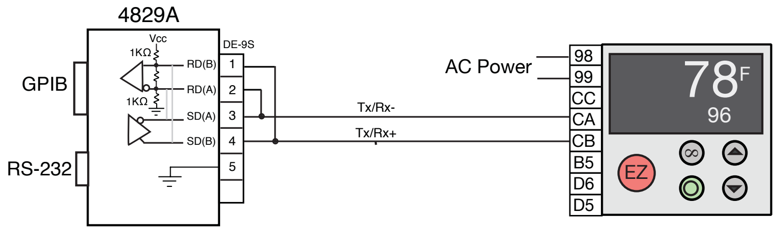


Figure 2-16 4829A connection to an EZ Zone Controller

2.11 JUMPER SETTINGS

2.11.1 4809A Jumper Settings

The 4809A has five jumpers as shown in Figure 2-17. Table 2-5 lists the jumper functions and their factory settings. For RS-232 serial signals, leave the jumpers in the '232' position. For RS-422 or RS-485 signals, set jumpers W4, and W5 to the '422' position.

2

TABLE 2-5 4809A JUMPER SETTINGS

Jumper	Function	Factory Setting
W1	Selects input power P1 for direct regulated 5 Vdc power REG for unregulated DC power	REG
W2	Set for EPROM size. Not user changeable.	512
W3	Write Enable. Installed enables saving setup parameters in EEPROM. Remove to block writes to configuration tables	Installed
W4 W5	RS-232/RS-422 signal selection Jumpers 232 for RS-232 signals 422 for RS-422 signals	232* 232*

Notes: * Set RS485 to 1 for half-duplex operation or to 0 for full-duplex operation.

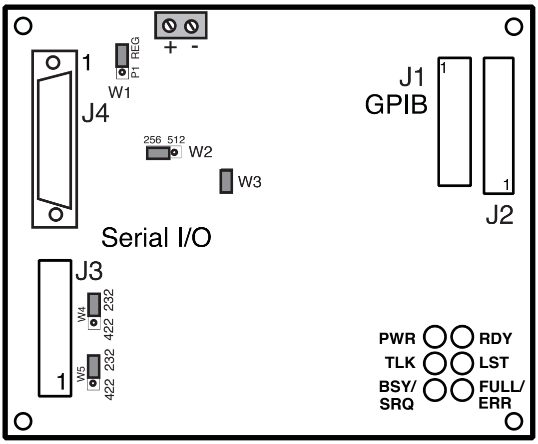


Figure 2-17 4809A W4 and W5 Jumpers set to RS-422 Position

2.11.2 4819A Jumper Settings

The 4819A has five jumpers as shown in Figure 2-18. Table 2-6 lists the 4819A jumper functions and their factory settings.

TABLE 2-6 4819A JUMPER SETTINGS

Jumper	Functions	Factory Setting
W1	Write Enable - Must be in place to write to or to save data in the 4819A's flash memory.	Installed
W2	Power Selection - Selects input power. P1 position selects the terminal block for a 5 Vdc regulated power source. REG selects the regulator output for 5.5 to 15 Vdc power.	REG
W3	Default - Install and power cycle the unit to reset the unit to its factory default settings. Leave out for normal operation.	Omitted
W4	TxD/RxD Signal Swapping Jumper Pair - W4 can be used to swap the TxD/RxD pins on connector J3. Factory setting is the DCE position with the pinouts listed in Table 2-2.	DCE
W5	Jumpers GPIB logic ground to chassis ground.	Omitted

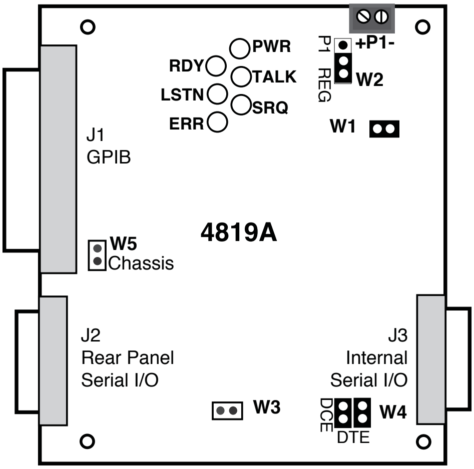


Figure 2-18 4819A Jumper Locations

2.11.3 4829A Jumper Settings

The 4829A has seven jumpers as shown in Figure 2-19. Table 2-7 lists the 4829A jumper functions and their factory settings.

TABLE 2-7 4829A JUMPER SETTINGS

Jumper	Functions	Factory Setting
W1	Write Enable - Must be in place to write to or to save data in the 4819A's flash memory.	Installed
W2	Power Selection - Selects input power. P1 position selects the terminal block for a 5 Vdc regulated power source. REG selects the regulator output for 5.5 to 15 Vdc power.	REG
W3	Default - Install and power cycle the unit to reset the unit to its factory default settings. Leave out for normal operation.	Omitted
W4 W5	Connects SD pair to RD signal pair. Install for half-duplex operation, omit for full-duplex operation.	Open*
W6	Enables RD transceiver 100% of time (Full-duplex) or only when transmitting (Half-duplex operation)	Full*
W7	Jumpers GPIB logic ground to chassis ground.	Omitted

Notes: * Set the RS485 setting to match your jumper setting.

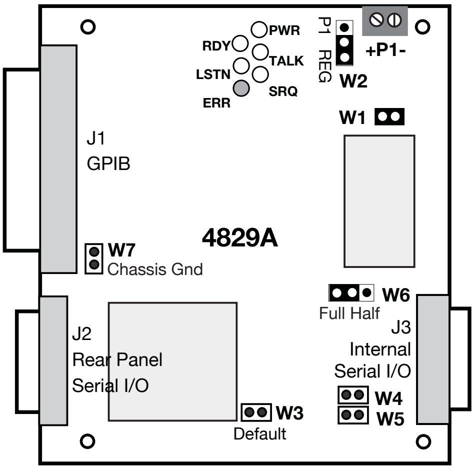


Figure 2-19 4829A Jumper Locations

2.11.4 4899A Jumper Settings

The 4899A has internal jumpers for selecting RS-232 and RS-422/RS-485 signals. Figure 2-19 on the next page shows the jumper locations on the 4899A's PC board. Table 2-8 lists the jumper functions.

The jumpers are accessed by removing the two screws on the rear panel and sliding the unit out from the rear of its case. Jumpers W4, W5 and W6 select the type of signals for the 4899A's serial interface. For RS-232 serial signals, leave the jumpers in the '232' position as shown in Figure 2-15. For RS-422 or RS-485 signals, set jumpers W4, W5 and W6 to the '422' position. Do not change the other jumper positions.

Carefully slide the unit back into its case, fitting the PC board into the tracks on each side of the case. Be sure the switch and LEDs are correctly aligned before pushing the switch through the front panel.

Replace the two rear panel screws.

TABLE 2-8 4899A JUMPER SETTINGS

Jumper	Function	Factory Setting
W1	Selects interrupt source CT/CV for 4899A and 4895, PRL for 4892	CT/CV
W2	Set for EPROM size. Not user changeable	512
W3	Write Enable. Installed enables saving setup parameters in EEPROM. Remove to block writes to configuration tables	Installed
W4 W5 W6	RS-232/RS-422 signal selection Jumpers 232 for RS-232 signals 422 for RS-422 signals	232* 232* 232*
W7 W8	LED signal jumpers CV/PRL for 4899A, CT for 4895	CV CV
W9	Default Jumper. Restores factory default settings when installed. Same action as the CAL:DEFAULT command.	Open

Notes: * Set RS485 to 1 for half-duplex operation or to 0 for full-duplex operation.

2

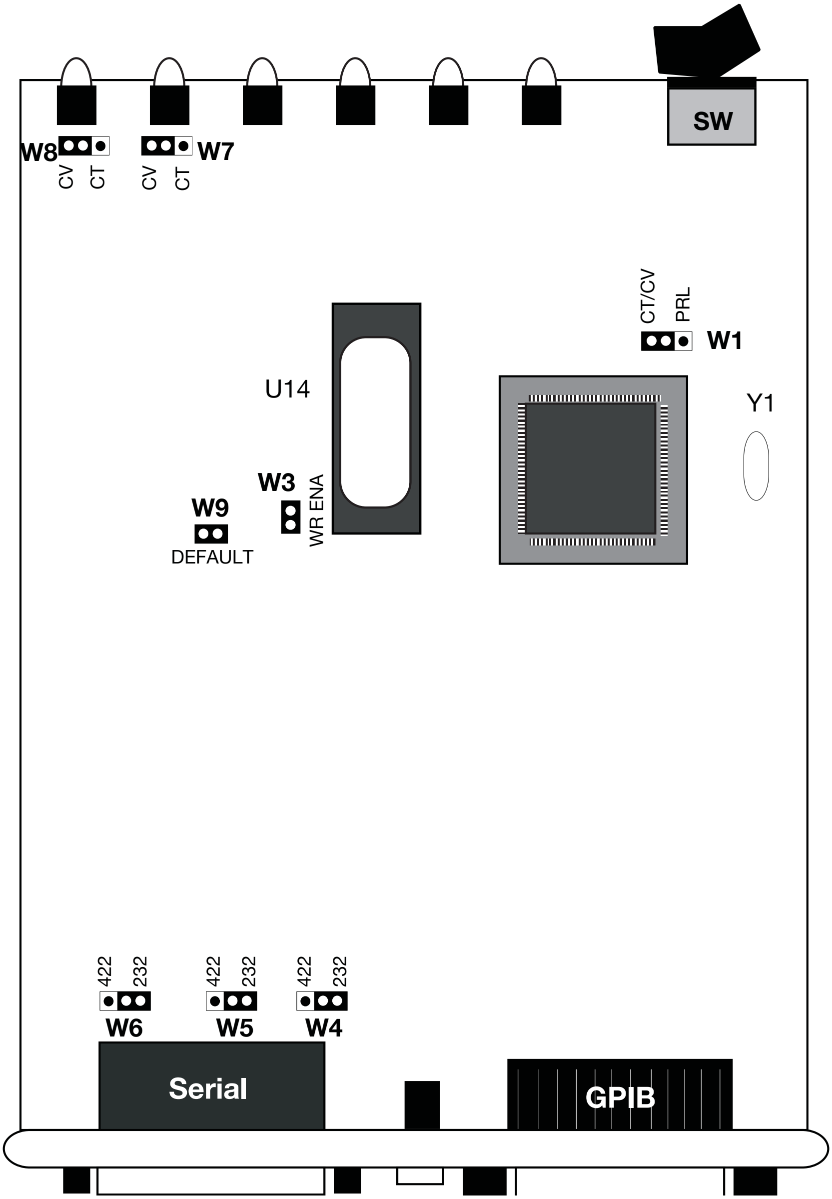


Figure 2-20 4899A Jumper Locations

2.12 4899A RACK MOUNTING INSTRUCTIONS

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

1. Hold the 4899A 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. A line 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.

2

This page left intentionally blank

3

Operation

3.1 INTRODUCTION

This section describes how the 4809A, 4819A, 4829A and 4899A interfaces operate and how they control Modbus RTU controllers from the GPIB bus. This section also describes the SCPI commands used to configure and control the units. Wherever the text refers to the Model 4899A, it applies equally to the Models 4809A, 4819A, and 4829A unless otherwise noted.

3.2 OPERATION

3.2.1 Basic Operation

The 4899A is a GPIB-to-Serial Interface with special firmware that converts GPIB commands into Modbus RTU packets to communicate with Modus devices. The 4899A responds to three types of commands:

1. IEEE-488.2 Common Commands in Table 3-2. These include the *IDN? and *ESR? queries and the *SAV 0 save command.
2. SCPI Commands in Table 3-3 are for setting internal values such as the GPIB address, serial parameters like baud rate and configuring the Status Reporting Structure.
3. Modbus Commands in Table 3-5 for communicating with Modbus device(s), setting the 4899A's serial timeout and reading the Modbus Error Register.

Any commands that end in a '?' are a query and the 4899A responds by outputting the response on the GPIB bus the next time it is addressed as a talker. Any 4899A parameter settings will be lost when the unit is power cycled unless the 4899A's configuration is saved with the *SAV 0 command.

Communication with the Modbus device is serial and requires that the user set the 4899A and the Modbus device to the same serial settings. Each Modbus device has its own address so that it can respond to serial packets sent to its address. **The Modbus device address is different and independent from the 4899A's GPIB bus address setting.** Although the typical Temperature Chamber has only one Modbus Controller, the 4899A can drive multiple Modbus devices when using a RS-485 network. The 'C' command is used to set the Modbus device address in the 4899A. The 4899A remembers the Modbus device address until changed by a subsequent 'C' command or the 4899A is powered off or reset.

3

Modbus devices are register based devices and they are controlled by writing values to registers that control different functions i.e. temperature setpoint, alarm settings etc. Data is taken from Modbus devices by reading registers associated with those parameters i.e. temperature, humidity, etc. ICS has created a set of simple Modbus commands for reading, writing and communicating with Modbus devices. When these Modbus commands are sent to the 4899A over the GPIB bus, the 4899A communicates with the selected Modbus device. Modbus commands should not be mixed or concatenated with IEEE-488.2 or SCPI commands.

If the 4899A's message packet is successfully received by the Modbus device, the Modbus device will generate a response packet that either confirms receipt of the message or that contains the requested data. The 4899A receives the response packet and validates the packet. If the response packet is a valid response 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 4899A is addressed to talk. If the message is an acknowledgment message, there is no further action.

The 4899A 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. It is better to set the timeout period to a larger than needed value to avoid unnecessary timeout errors.

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

3.2.2 4809A Differences

The 4809A is a 4.5 by 5.5 inch board that mounts inside a chassis or temperature chamber. It is functionally identical to the 4899A and has both RS-232 and RS-422/RS-485 serial interfaces. The 4809A's baud rate is limited to 38,400 baud.

3.2.3 4819A Differences

The 4819A mounts on the rear panel and has both a GPIB and an external serial connector which extends the Modbus serial path to the rear panel of the host chassis. The user can control an RS-232 Modbus device from the serial connector or from the 4819A's GPIB interface. However, both interfaces cannot be used at the same time. Once the GPIB interface is activated, the serial interface cannot be used to control the Modbus device until the 4819A has been powered off and back on.

3.2.4 4829A Differences

Like the 4819A, the 4829A mounts on the rear panel of the host chassis and has both a GPIB and an external RS-232 serial connector which extends the Modbus serial path to the rear panel of the host chassis. The user can control one to several RS-485 Modbus device(s) from the serial connector or from the 4829A's GPIB interface. However, both interfaces cannot be used at the same time. Once the GPIB interface is activated, the serial interface cannot be used to control the Modbus device until the GPIB connection is removed.

3.3 ADDRESSING THE UNITS

3.3.1 Internal GPIB Address

The 4809A, 4819A and 4899A 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. Refer to Sections 2.6 and 2.7 for address setting instructions.

3.3.2 4809A Address External Address Switch

The 4809A's GPIB address can also be set by connecting the 4809A 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 4809A's external address switch is enabled with the SCPI SYST:COMM:GPIB:ADDR:EXT ON command. When enabled, the 4809A 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 4899A 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 4899A. 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 4899A 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.

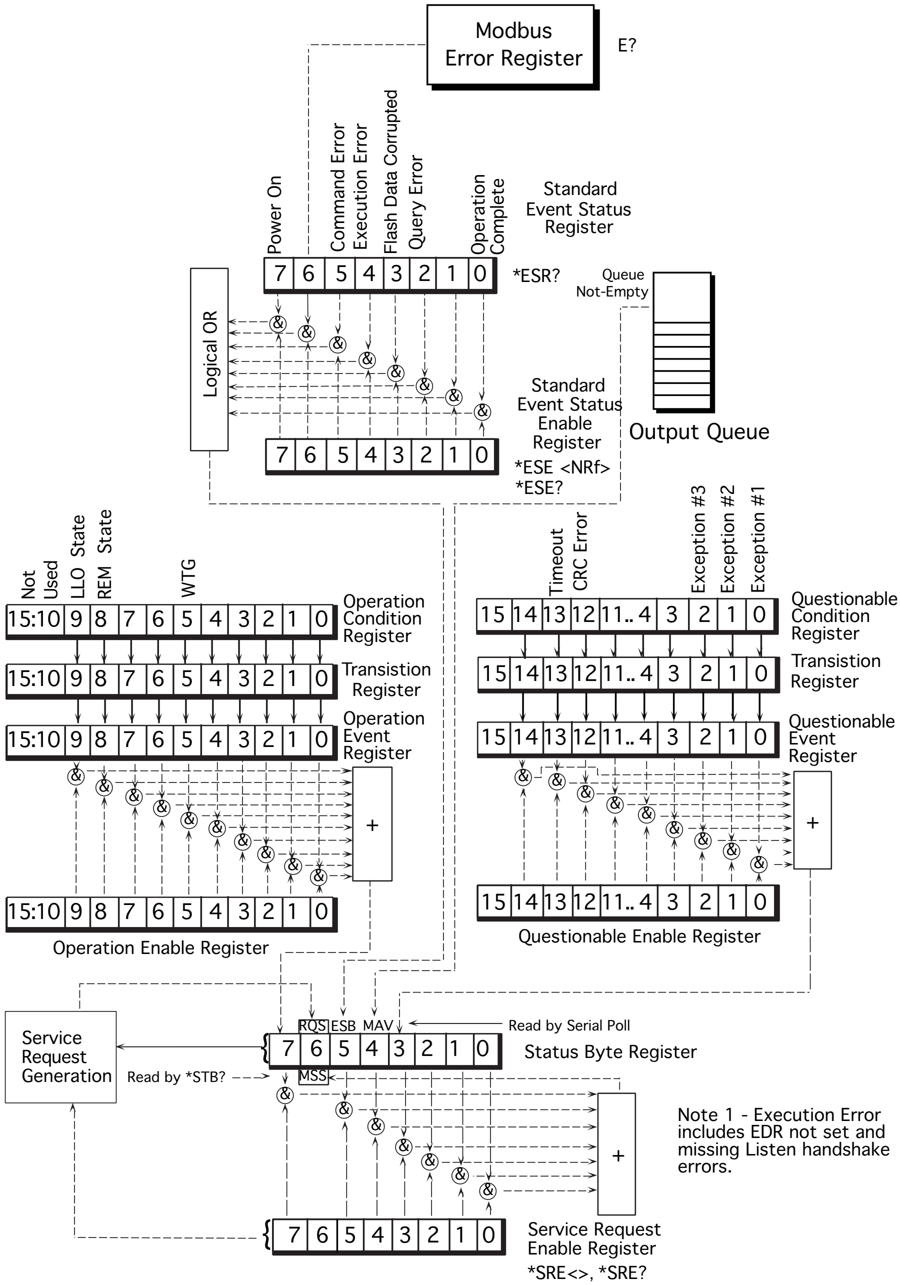
3.4.1 Event Registers

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.

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



Note 1 - Modbus Register codes are listed in Table 3-5 on page 3-22

Figure 3-1 48x9A Status Reporting Structure

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

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. ESR bits 2 through 6 light the ERR LED when on. The Power-on event occurs at power turn-on and can be used to signal a power off-on occurrence. In the 4899A, 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 unit.

3

TABLE 3-1 ESR BIT DEFINITIONS

Bit	Bit Weight	Event	Description
7	128	PON	The Power-on event occurs at power turn-on and can be used to signal a power off-on occurrence.
6	64	Modbus Error	Modbus Error detected. Reading the Modbus Errror Register clears this bit. See the E? query in Table 3-5.
5	32	Cmd	Command Error
4.	16	Exc	Execution Error includes EDR not set and missing listen handshake.
3	8	Flash	Flash data corrupted.
2	4	Query	Query error, data not read or read attempt with no data.
1	2	not used	
0	1	OPC	Operation Complete has no meaning in the 48x9As.

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 ESR error bits 2 through 5
***ESE 124**	‘enables ESR error bits 2 through 6
***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 3-5 for the Modbus Error Register values. The following commands will generate a Service Request when a Modbus error occurs:

***ESE 64**	‘enables ESR bits 6
***SRE 32**	‘enables StatusByte bit 5
***ESR?**	‘reads ESR Register bits
E?	‘reads Modbus Error Register

3

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 3-1. 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 4899A 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

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. In the 4899A, 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 4899A 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?	‘quires the Operation Condition Register

3.4.6 Output Queue

The Output Queue is used by the 4899A 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 4899A 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 4899A generates a service request (SRQ) whenever any of the enabled bits in the Status Byte Register become true and the 4899A 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

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 negative transition
***ESE 192;*SRE 32;*PSC 0**	'saves ESE and SRE bits 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. The Enable and Transition Register values can be set or changes at any time by program commands.

3.4.9 488.2 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

The IEEE 488.2 Standard mandated a list of common commands that are common to all IEEE 488.2 compatible devices. The 4899A, 4819A and the 4809A respond to all of the mandated common commands and to some optional commands defined in IEEE-488.2. Table 3-2 lists the IEEE-488.2 commands that apply to this unit, and describes the affect they have on the 4809A, 4819A or the 4899A and its status reporting structure.

3.6 SCPI CONFORMANCE INFORMATION

TABLE 3-2 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>. <value> is an integer between 0 and 255, 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 which sets bit 4 to a logical 1.
***ESE?**	Event Status Enable Query	4899A 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	4899A 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.

3

TABLE 3-2 IEEE-488.2 COMMON COMMANDS (CONTINUED)

COMMAND	NAME	DESCRIPTION
***IDN?**	Identification Query	4899A returns its identification code as four fields separated by commas. These fields are: manufacturer, model, six-digit serial number and hardware-firmware version and date e.g. **ICS Electronics, 4899A, S/N 603001, Rev. 00.00, 06.03.02.** The IEEE-488.2 specification states that the word 'model' may not appear in the IDN message.
***OPC**	Operation Complete Command	Causes the 4899A to generate the operation complete message in the Standard Event Status Register when all pending selected 4899A operations have been finished.
***OPC?**	Operation Complete Query	Places an ASCII character 1 into the 4899A's Output Queue when all pending selected 4899A operations have been finished.
***PSC<value>**	Power-On Status Clear	Controls the automatic power-on clearing of the SRE and ESE registers. *PSC 0 allows devices to restore the saved SRE and ESE values and to assert SRQ upon power turn-on. *PSC 1 enables the power-on clear and disallows a SRQ at power turn-on. The PSC commands saves the 488.2 SRE and ESE registers and the SCPI transition and enable register values.
***PSC?**	Power-On Status Clear Query	Querys the PSC flag value. A returned value of 0 indicates the registers will retain their saved values, a returned value of 1 indicates the registers will be cleared.
***RCL <value>**	Recall	Restores the state of 4899A from a copy stored in its Flash by *SAV command. *RCL 0 recalls saved configuration, updates output levels and re-initializes the UART. Allow the 4899A 100 ms and the 2303 150 ms to complete the *RCL command.

TABLE 3-2 IEEE-488.2 COMMON COMMANDS (CONT'D)

COMMAND	NAME	DESCRIPTION
***RST**	Reset	4899A restores its power-up state except that the state of IEEE-488 interface is unchanged, including: instrument address, Status Byte and ESR Register. Disables the trigger function and pulses the Reset output signal. Allow the 4899A 100 ms and the 2303 150 ms to complete the *RST command.
***SAV <value>**	Save	Saves the 4899A's current configuration in the Flash. *SAV 0 saves the current setting as the new power on setting. <value>=0
***SRE <value>**	Service Request Enable	Sets the "Service Request Enable Register" to <value>. The value of bit six is ignored because it is not used by the Service Request Enable Register. <value> is an integer between 0 and 255, 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	4899A 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	4899A 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.
***TRG**	Device Trigger	Pulses the Trigger Output line.
***TST?**	Self-Test Query	Queries the results of the last self test. A zero response indicates no failures. Other responses are not returned as the unit will be running in a blink LED loop and will be unable to respond to the query.
***WAI**	Wait-to-continue	Prevents the 4899A from executing any further commands or queries until the No-Operation-Pending flag is TRUE.

The 4899A 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 1994.0 and provide an industry standard, self-documenting form of code that makes it easy for the programmer to maintain the application program.

Table 3-3 shows the 4899A's SCPI command tree. The command tree uses portions of the SCPI SYSTEM, STATUS, FORMAT, INITIATE, ABORT and CALIBRATE subsystems. The 4899A and 4809A 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.

STATus:QUEStionable?	is the same as
STAT:QUES:EVEN?	or also as
stat:ques?	

Table 3-4 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 4899A'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.

3.8 PROGRAMMING GUIDELINES

TABLE 3-3 SCPI COMMAND TREE

Keyword	Parameter Form	Notes & Short Form Commands
SYSTem		**System Address**
:COMMunicate		
:GPIB		
:ADDRess	0 - 30 [4]	
:EXTernal	0l1 or OFFlON [0]	4809A only
:SERial		
:BAUD	<numeric value> [9600]	
:PARity	EVEN l ODD l [NONE]	
:BITS	7 l [8]	
:SBITs	[1] l 2	
:UPdate	no value-command only	
:RS485	0l1 or OFFlON [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 3-3 SCPI COMMAND TREE CONT'D

Keyword	Parameter Form	Notes & Short Form Commands
FORMat		**Format Strings**
[:DATA]		
:TALK	ASCii I HEXL	
	[ASCII]	FT
CALibrate		**Calibrate**
:IDN	string	
:DATe	mm/dd/yy	
:DEFault		
:LOCK	1(On)I 0(Off) [0]	

3

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 4809A'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 4809A'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.

TABLE 3-4 SCPI COMMANDS AND QUERIES

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 4809A's primary address.
:EXTernal	OFF	On enables the 4809A's external 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 4899A. Values = 0|1 or OFF|ON.
:SERial		Controls Serial Interface settings
:BAUD	9600	Sets serial baud rate. Vallues for the 4809A are 300 to 38400 baud. Values for the 4899A and 4819A 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 re-program the serial controller's COM port after this command.
RS485	OFF	Tristates 4899A and 4809A transmitter when not transmitting for two wire networks. Values are ON and OFF.

3

TABLE 3-4 SCPI COMMANDS AND QUERIES CONTINUED

Keyword	Default Value	Description
:ERRor?	0, "No error"	Requests next entry in 4809A's error/event queue. Error messages are: 0, "no error" -100, "Command error" -200, "Execution error" -400, "Query error"
:VERSion?	1994.0	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 4809A detects a positive changes
FORMat		Starts string format branch.
:DATA		Optional digital data identifier

3

TABLE 3-4 SCPI COMMANDS AND QUERIES CONTINUED

Keyword	Default Value	Description
: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. 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, 4809A, S/N 708001, Rev 00.14, Ver 07.08.28.**
: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.
: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 1-1 lists the locked parameters.

3

3.7 MODBUS COMMANDS

The following commands are used to Control Modbus slave devices. These Modbus Commands should not be mixed or concatenated with IEEE-488.2 or SCPI commands. Commands marked with an asterisk are new in the 48x9A series interfaces and are not available in the older 48x9 interfaces.

3

TABLE 3-5 MODBUS COMMANDS

Syntax	Meaning
C addr	Modbus Address Command. Sets Modbus slave device address for subsequent commands. Value for *addr* is 1 to 255. Default setting is 1.
RC[?] reg, ncoil *	Read Coil Status Command (code 0x01). Reads the status of coils in a remote device. User specifies starting coil address in register *reg* and number of coils to be read *ncoil*. The [?] is an optional symbol for smart programs.(4) Values for *reg* are 0 to 65535. Values for *ncoil* are 1 to 2000. Responses are returned as a packed binary value with 1-bit per coil, 8 coils per byte. 1 = ON.
RI[?] reg, ninp *	Read Discrete Inputs Command (code 0x02). Reads discrete inputs. User specifies starting address in register *reg* and number of inputs to be read *ninp*. The [?] is an optional symbol for smart programs.(4) Values for *reg* are 0 to 65535. Values for *ninp* are 1 to 2000. Responses are returned as a packed binary value with 8 inputs per byte. 1 = ON.
R[?] reg, num	Read Register Command (code 0x03). 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 for smart programs.(4) Values for *reg* are 0 to 65535. Values for *num* are 1 to 64. Responses are returned as signed 16-bit decimal or HEX values , +32767 to -32767, 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.

TABLE 3-5 MODBUS COMMANDS CONT'D

Syntax	Meaning
RR[?] reg, num *	Read Input Register Command (code 0x04). 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 for smart programs.(4) Values for *reg* are 0 to 65535. Values for *num* are 1 to 64. Responses are returned as signed 16-bit decimal or HEX values, +32767 to -32767, separated by commas. Output format selected with the Format command. See the R? query above.
RE[?] *	Read Exception Status Query (code 0x07). Reads eight exception status outputs from a remote device. The [?] is an optional symbol for smart programs.(4) Responses are returned as a packed binary value with the eight status bits in one byte.
RF? reg	Read Floating Point Value Command (code 0x03). Reads two sequential registers as an IEEE-754 32-bit floating point value in low byte to high byte order. The specified register, *reg*, contains the lower two bytes and the next higher register contains the upper two bytes.
WC reg, b *	Write Coil Command* (code 0x05). Writes a ON/OFF value, *b* to a single Modbus device register, *reg*. Values for reg are 0 to 65535. Values for *b* are 0/OFF or 1/ON/255. An example is: WC 1000, ON
W reg, w	Write Register Command (code 0x06). Writes a 16-bit value, *w* to a single Modbus device register, *reg*. Values for reg are 0 to 65535. Values for *w* are +32767 to -32767. An example is: W 100, 55 writes the decimal 55 to register 100.
WB reg, num, w(0),.w(n)	Write Block Command (code 0x10). Writes multiple 16-bit words, *w(i)* to multiple registers or 32-bit values to two adjacent registers. Starting register, *reg*. Values for *reg* are 0 to 65535. Number, *num* specifies how many words are to be written. Values for *num* are 1 to 64. Values for *w* are +32767 to -32767. *w(i)* values are separated by commas.

3

TABLE 3-5 MODBUS COMMANDS CONT'D

Syntax	Meaning
WF reg, num	Write Floating Point Value Command (0x16). Writes an IEEE 754 single precision 32-bit value to two registers in low word to high word order. *reg* specifies the low word. *reg* +1 is the high word. *num* is determined by the parameter being controlled and can range from 2^{127} to 2^{-127}
L[?] w	Loopback Command (code 0x08). 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 for smart programs.(4) Value for *w* is 0 to 65535.
D time	Timeout Command. Sets timeout value of Modbus response message in milliseconds. Timeout is the total time for the message to be received by the 48x9A. Value for *time* is 1 to 65,535 milliseconds. The default value 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 the response message. 2nn Partial or corrupted message received. where 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. Integer range is 0 to 65535.
2. Response parameter format set by SCPI FORMat command. Default is ASCii
3. Do not combine the Modbus commands in Table 3-5 with IEEE-488.2 commands or SCPI commands to avoid query errors or otherwise confusing the GPIB<->Modbus Interfaces.
4. The [?] is an optional symbol for smart programs like ICS's GPIBKybd program. These programs can recognize the command as a query and automatically read the response.

The following section provides information on how to program the 4809A, 4819A and 4899A to set its configuration and how to send commands to the Modbus slave device. New users should try these simple examples with a live keyboard program like ICS's GPIBkybd program to become familiar with controlling the GPIB<->Modbus Interface and Modbus device(s) from the GPIB bus. Although the comments refer to a 4899A, they apply to all units unless otherwise noted.

The Visual Basic TempCltr example program on the Support CD-ROM can be used as an example when writing Temperature Control programs.

Modbus commands should not be mixed on the same command line with IEEE-488.2 and SCPI commands to prevent query errors and confusing the GPIB-to-Modbus Interface.

3

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:

Send IFC	‘gets control, asserts REN
Send SCPI command	‘change a setting
Send SCPI query	'verify new setting
Send *ESR?	‘query the Event Status Register to be sure that there were no errors or visually check the ERR LED.

Repeat the change and verify steps for each parameter you are changing. When done save the new values.

Send “*SAV 0”	‘save the new configuration

The *SAV 0 command will cause the 4899A 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

Set the Modbus device address in the 4899A with the 'C' command so that it matches the address set in the desired Modbus device. The 4899A remembers the Modbus address until it is changed. It is only necessary to send the 4899A the 'C' command at the start of the program. If the 4899A is being used with only one Modbus device, the address can be set and saved as part of the 4899A's power on configuration. The 4899A and most Modbus devices default to a Modbus address of 1.

```
C n          ‘sets device address to value n
*SAV 0       ‘optional save new default address
```

3.8.3 Querying a Modbus Device

3

The second step is to send a query to the 4899A and read back the response from the Modbus device. The following commands set Modbus device #1 and read one 16-bit 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          ‘optional-sets Modbus Address to 1
R? 0,1       ‘reads Watlow model number
             ‘Watlow F4 response is 5270
```

A more realistic command might be to read a measured value. Register numbers and functions vary with different Modbus devices so consult your Modbus device manual for its register numbers and functions. With Watlow F4 series Controllers, register 100 is the measured temperature value.

```
R? 100,1     'reads temperature from a Watlow F4
```

For multiple devices, the device address and read command can be concatenated on one line by separating them with semicolons. i.e.

```
C 1; R? 100,1  ‘concatenated command example writes
               'Modbus address and queries register 100
```

3.8.4 Writing to the Modbus Device

The nature of the command depends upon the specific Modbus device. Simple writes to 16-bit Modbus registers are handled with the W command. In the following example, a value of 50 is written to register 300. i.e.

W 300, 50	'sets F4 temperature setpoint
C 7; W 300, 50	'concatenated command

Writes to sequential Modbus registers are possible with the WB command.

3.8.5 32-Bit Variables

3

Most Modbus devices have 16-bit wide registers for setting a parameter and for reading back data. The prior command examples showed how to read and write to 16-bit registers. Watlow's new Temperature Controllers like the Series SD and Series PD have 32-bit registers which are accessed as two 16-bit registers. The value is assumed to have three decimal places.

3.8.6 32-Bit Write

To write a set point of 1250 degrees (which is really 1250.000) to Registers 27 and 28, multiply the setpoint value (SP) by 1000 to get 1,250,000. Add 65536 to negative numbers. This produces the setpoint (SP) we want to send. To determine the most significant word (MSW) for Register 27, divide the SP by 2^16 or 65536. To determine the least significant word (LSW) for Register 28, subtract from the SP the result of multiplying the MSW by 2^16 or 65536. i.e.

SP = 1250 * 1000 = 1250000
MSW = 1250000 / 65536 = 19
LSW = 1250000 - (19 * 65536) = 4816

The 4899A can write each register separately with standard write commands or both registers can be written together with the Write Block command. Examples are:

W 27,19	'writes to register 27
W 28,4816	'writes to register 28
or	
WB 27,2,19,4816	'writes to registers 27 and 28

3.8.7 32-Bit Read

To read a 32-bit value, two successive 16-bit registers are read and the user's program then puts the values together to form the 32-bit result. An example is reading a process variable from Registers 20 (MSW) and 21 (LSW). The 4899A can be used to read each register individually or to read two successive registers. The commands are:

```
R? 20,1        'reads register 20
R? 21,1        'reads register 21
  or
R? 20,2        'reads register 20 and 21
```

Both sequences return two numbers to the user. The MSW is returned from Register 20, the LSW from Register 21. Multiply the MSW by 2^16 or 65536 and add it to the LSW. Divide the result by 1000 to scale it to three decimal places.

Reading = ((MSW * 65536) + LSW)/1000

3.8.8 Floating Point Variables

Some new Modbus devices like Watlow's EZ PM series controllers use two consecutive register to control a value or to read back a process variable. The two registers hold an IEEE-754 32-bit floating point word. The registers are read and written to in the low word-upper word order.

3.8.9 Floating Point Write

The WF command writes the num value in floating point format to two consecutive registers starting with the low word register.

```
WF 2160, 75        'writes to registers 2160 and 2161
```

3.8.10 Floating Point Read

The RF? query reads a 32-bit floating point value from two sequential register in low word-upper word order. The RF? does not require the number of register to read since it is fixed at two registers.

```
RF? 360        'reads registers 360 and 361
```

3.8.11 Setting Modbus Device Timeouts

The time that the 4899A waits to receive a response from the Modbus device is programmable by the D command. This is an internal 4899A timeout and not the same as the GPIB bus timeout. If the 4899A does not receive a response within the timeout period, it assumes that the Modbus device is not responding and sets the timeout error. Timeout errors can be determined by reading the 4899A's Modbus Error Register with the E? query. If the error code is 101 (Modbus timeout) then the timeout period should be lengthened. The command to change the timeout period is:

```
D 300                    'sets timeout period to 300 ms
```

The default time period of 100 milliseconds has proved to be satisfactory for some Watlow controllers but should be verified 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

3.8.12 Locking Setup Parameters

All of the 4899A'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 setup parameters
```

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

3.8.13 Generating SRQs from Modbus Errors

Figure 3-1 shows the 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.

3.8.14 Personalizing the Unit’s IDN Message

The 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:

CAL:LOCK OFF ‘unlocks all parameters

CAL:IDN Acme Test Co, 101, s/n 007, Rev 1 07/08/30
'enter a new IDN message
***SAV 0** ‘saves lock status

CAL:LOCK ON	‘relocks all parameters
***SAV 0**	‘saves lock status

3.8.15 Saving the Configuration

The *SAV 0 command saves 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 4809A, 4819A, 4829A and 4899A 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 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 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 or products that incorporate ICS's Model 4809A, 4819A or 4899A. 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 4809A, 4819A, 4829A and 4899A GPIB<->Modbus Controllers. The models are similar and have a similar block diagram. Wherever the text refers to the Model 4899A, it applies equally to the other models unless otherwise noted.

The Model 4819A and 4829A have some additional features and is covered separately.

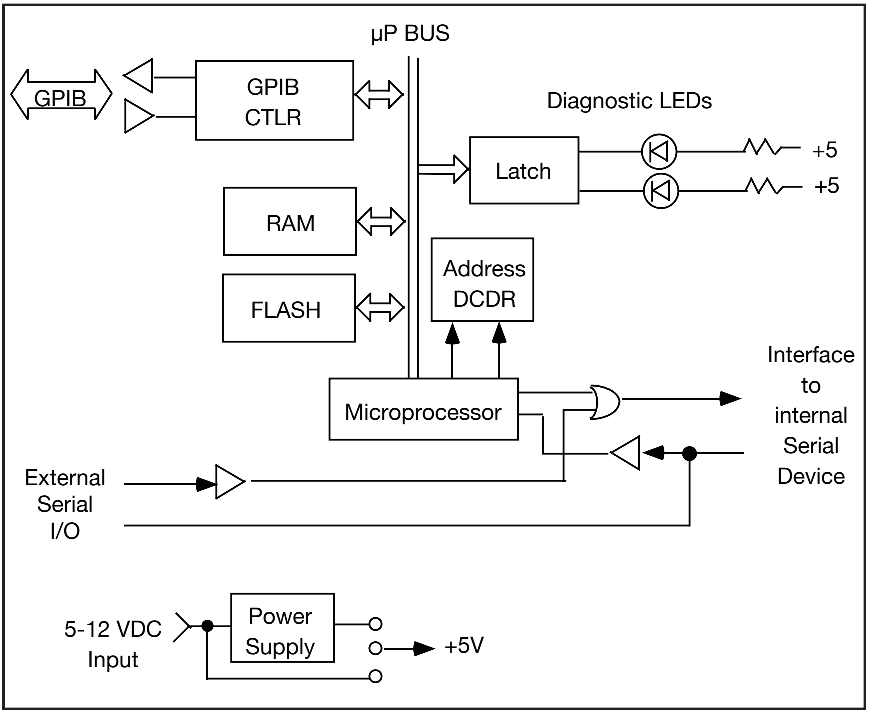
4

4.2 4899A AND 4809A BLOCK DIAGRAM DESCRIPTION

A block diagram of the 4899A is shown in Figure 4-1. The 4899A 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 4899A 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 4809A's block diagram is similar to the 4899A.

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.

If the message is a Modbus command then it is converted into a series of binary characters to make up the Modbus RTU message packet. The Modbus



4

Figure 4-1 4899A Block Diagram

message packet includes the slave device address, the command number, the registers and data (if any) that is being sent to the registers. A check-sum is added to make up the complete Modbus RTU packet. The Modbus packet 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 4899A to set bits in the Questionable Register and to place an error value in the Modbus Error Register. The 4899A contains a multilevel Status Byte Register and Event Register structure enables the 4899A to generate a Service Request and interrupt the GPIB bus controller when errors are detected.

Flash Memory contains all of the 4899A's program instructions, command tables, and power turn-on/self test routines. At power turn-on, the 4899A 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 4899A 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.

The 4899A's and 4809A's configuration settings, serial number and other parameters that are subject to change are saved in a nonvolatile Flash sector. 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 not stored in the Flash sector until the user sends the 4899A the *SAV 0 command.

In the 4899A, the RAM is a 8 bit wide memory that is primarily used for data storage, operating variables and configuration settings. The 4899A 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 4899A simply inhibits further Bus handshakes until there is room in the GPIB buffer for more data.

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

The 4809A has an on-card regulator that converts unregulated +5.5 to +12 volt DC power to +5 Vdc to run the 4809A's circuits. A jumper on the 4809A bypasses the regulator so that the 4809A can be run from regulated 5 volt DC power.

4.3 4819A BLOCK DIAGRAM DESCRIPTION

A block diagram of the 4819A is shown in Figure 4-2. The 4819A 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 4819A also has a serial I/O connector for communicating directly with the Modbus slave device. The 4819A is made up of seven major elements, most of which are interconnected to the microprocessor by a common data, address and control signal bus.

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 to see if it is an IEEE-488.2 Common Command or if it is a SCPI command. SCPI commands and IEEE-488.2 commands are parsed and used to set the 4819A's configuration, 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.

If the message is a Modbus command then it is converted into a Modbus packet and outputted through the serial OR gate to the Modbus device. Incoming serial data from the Modbus slave device is converted into TTL levels and applied to the input of the 4819A's UART. Received packets are checked as described for the 4899A. If the received message is a valid response, the response data is converted in to the correct format and placed in the GPIB buffer where can be output onto the GPIB bus when the unit is next address to talk. Messages that contain errors or Exception messages cause the 4899A to set bits in the Questionable Register and to place an error value in the Modbus Error Register.

The 4819A also provides a serial path from rear panel serial connector to the Modbus device. External serial data is ORed with the serial output of the GPIB logic to make the serial output to the Modbus device. Serial responses from the Modbus device are routed to both the rear panel serial

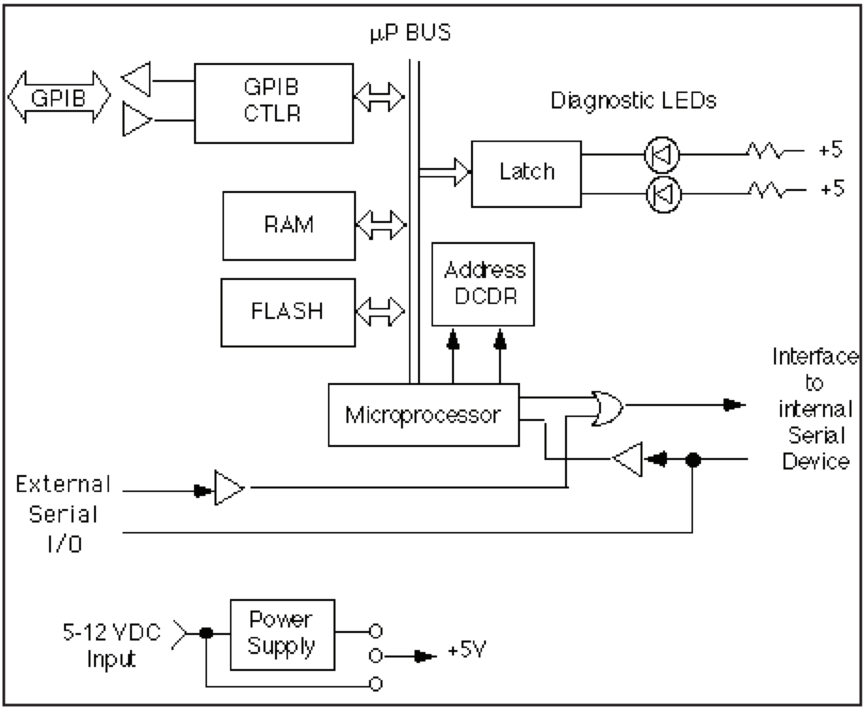


Figure 4-2 4819A Block Diagram

4

connector and to the GPIB logic. Because of the simple OR gate path, only one interface can be used to control the Modbus device at a time.

The Flash memory contains all of the 4819A's program instructions, command tables, and power turn-on/self test routines. At power turn-on, the 4819A 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 4819A displays its current GPIB address setting on its LEDs. The RDY LED comes on to indicate a successful completion of the self test routine.

The Flash also contains all of the 4819A's configuration settings, serial number and other parameters that are subject to change. At power on time, the microprocessor copies the configuration from Flash memory 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 Flash memory until saved with the *SAV 0 command.

In the 4819A, the RAM is a 8 bit wide memory that is primarily used for data storage, operating variables and configuration settings. The 4819A data buffers are several times larger than any anticipated message so no data loss ever occurs. GPIB bus data is never lost since the 4819A simply inhibits further Bus handshakes until there is room in the GPIB buffer for more data.

The 4819A has an on-card regulator that converts unregulated +5.5 to +12 volt DC power to +5 Vdc to run the 4819A's circuits. A jumper on the 4819A bypasses the regulator so that the 4819A can be run from regulated 5 volt DC power. A DC-DC converter in the RS-232 transmitter IC makes the ± 9 Vdc necessary to power the RS-232 drivers.

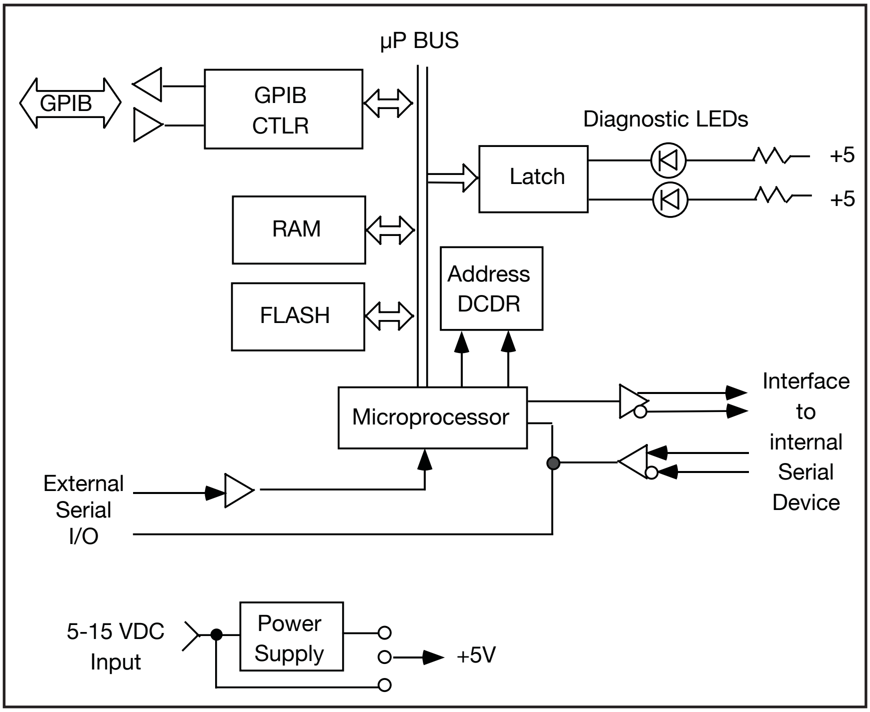


Figure 4-3 4829A Block Diagram

4.4 4829A BLOCK DIAGRAM DESCRIPTION

A block diagram of the 4829A is similar to the 4819A block diagram shown in Figure 4-2. Major difference is that signals from the External (RS-232)

Serial Interface is routed into the microprocessor. The incoming serial message is buffered and then retransmitted to the internal serial device. The serial signals to the internal device are differential RS-485 signals. Internal jumpers connect the transmit and receive signals together for two-wire, half duplex operation.

4

4

This page intentionally left blank

5

Troubleshooting and Repair

5.1 INTRODUCTION

This section describes the maintenance, troubleshooting and repair procedures for the Model 4809A, 4819A and 4899A GPIB <-> Modbus Interfaces. All comments and errors apply equally to all units unless otherwise stated.

5.2 MAINTENANCE

The 4809A, 4819A, 4829A and 4899A do not require periodic calibration and have no internal adjustments. However, if the interfaces are used in an application where the IEEE 488 bus cables are frequently changed or if the input signals appear erratic, the 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 GPIB 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 4809A, 4819A and 4899A 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 SELF TEST ERROR CODES

Blinking LED	Error	Possible Fault
All	Flash Variable Sector	Flash Variable Memory corrupted and unit reset variable memory 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	Flash Variable Sector	Loose Flash in socket. Check part for bent pins and press into socket. U14 in 4899A, U7 in 4809A or U2 in 4819A. Defective Flash that cannot be written to. Replace part. Defective decoder GAL. Test and/or replace GAL. U3 in 4899A, U16 in 4809A or U5 in 4819A.
TALK	Flash Program Memory	Flash loose in its socket or has a bent pin. Check Flash chip for a bent pin then press te Flash chip into its socket. U14 in 4899A, U7 in 4809A or U2 in 4819A. Flash Memory dropped a bit. Replace the defective Flash with a known good Flash. Defective decoder GAL. Test and/or replace GAL. U3 in 4899A, U16 in 4809A or U5 in 4819A.
LSTN	RAM	Defective RAM. Replace RAM with new IC. U18 or 22 in 4899A, U8 in 4809A or U3 in 4819A. Defective decoder GAL. Test and/or replace GAL. U3 in 4899A, U16 in 4809A or U5 in 4819A.

5

TABLE 5-1 SELF TEST ERROR CODES CONTINUED

Blinking LED	Error	Possible Fault
SRQ	GPIB	Defective GPIB controller chip. Replace GPIB Controller, U6 in 4899A, U3 in 4809A or U8 in 4819A.
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. U8 in 4819A
ERR + LSTN	CPU type	Wrong CPU type. Should be Zilog Z8S18020VSC (SL19A19A 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 and slave 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	Flash data lost	Use *CLS to clear the LED. Use CAL:DATE command to enter new date so ERR LED will not come on at next power-on time. Recall the factory defaults with the CAL:DEFAULT command. Use *SAV 0 to save the factory configura tion. Reload your configuration and use the *SAV 0 command to save your new 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 the GPIB address.

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.
	Modbus Error	If ESR bit 6 is on, query the Modbus Error Register with the E? command.
No communication with Modbus device	Wiring error	Check RS-232 or RS-485 cable wiring against figures in Section 2. Modbus TXD signal must go to 48x9A receive input. Use an RS-232 Tracker or DVM to find Modbus TXD signal. TXD will be -6 to -12 Vdc.
	Parameter setting mismatch	Check 4899A Serial settings vs the settings on the Modbus device. Use the SCPI commands in Table 3-2 to query or change the 4899A settings. Set the serial parameters to match. Save new 4899A settings with the *SAV 0 command. Power cycle the 4899A or use the SYST:COMM:SER:UP command to activate the new settings.

TABLE 5-2 TROUBLESHOOTING GUIDE CONTINUED

Symptom	Possible Fault	Action or Check
	Wrong Modbus address or address changed.	Query the 4899A's Modbus device address and compare it against Modbus device setting. Set Modbus device address with the C n command.
	Message not being received by the Modbus device.	Watch Access LED on Modbus device to verify serial message received by Modbus device.
	Command syntax	Modbus device manual for applicable commands, registers and values. Check Modbus Error register for Modbus problems
Faulty or missing Modbus responses	Modbus timeout setting	Check Modbus Error Register with the E? query for a time out error (Error 101) Use D command to lengthen timeout to 300 or 500 msec. Save setting with the *SAV 0 command.
No communication on RS-485 link.	RS-485 not enabled	4899A/4809A 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 FACTORY DEFAULT RECOVERY

If the boards configuration gets into an unknown setting, it can be restored to the factory default configuration with one of the following procedures. Use the CAL:DEFAULT command to re initialize the settings. If that does not work, use the hardware default jumper.

1. Power the board off.

2. Locate the default jumper and place a shorting jumper on it. See Section 2.11 for jumper locations.

3. Power the board on. After 5 seconds the LEDs should blink as the variables are reset. Wait until the LEDs stabilize and the board has finished its save procedure (takes approximately 10 seconds).

4. Remove the shorting jumper and operate the unit normally.

5.5 REPAIR

Repair of the 4809A, 4819A, 4829A and 4899A is done 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.

This page left intentionally blank

5

Appendix

Appendix		Page

A

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.

Two types of messages are transferred over the bus:

Interface messages - for bus management

Device-dependent messages - for device control and data transfer

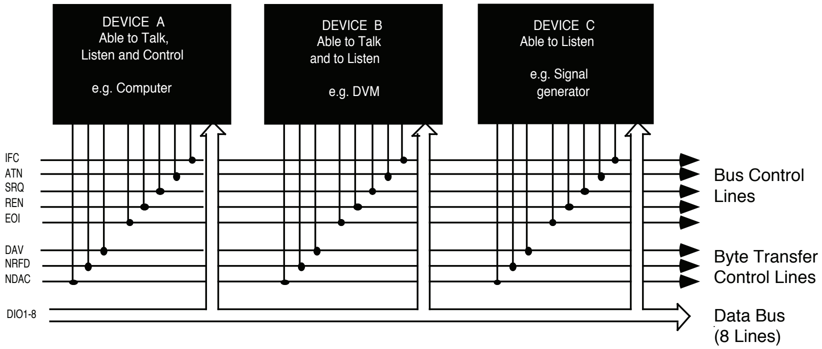


Figure A-1 IEEE 488 Bus

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 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 sub-functions, or

A

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.

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

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

TABLE A-1 IEEE 488 COMMAND AND ADDRESS MESSAGES

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	`	MEANING DEFINED BY PCG CODE	p	MEANING DEFINED BY PCG CODE
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		w	
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.

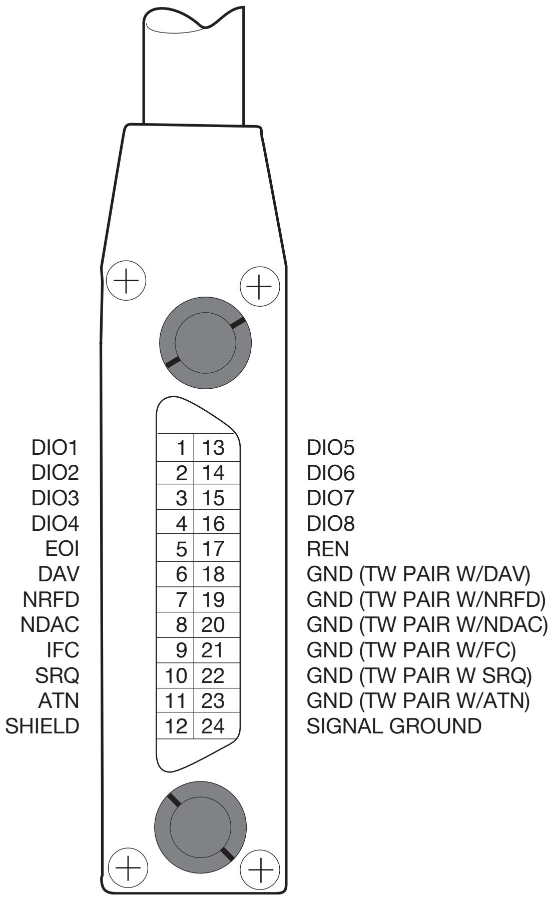


Figure A-2 GPIB Signal-Pin Assignments

A

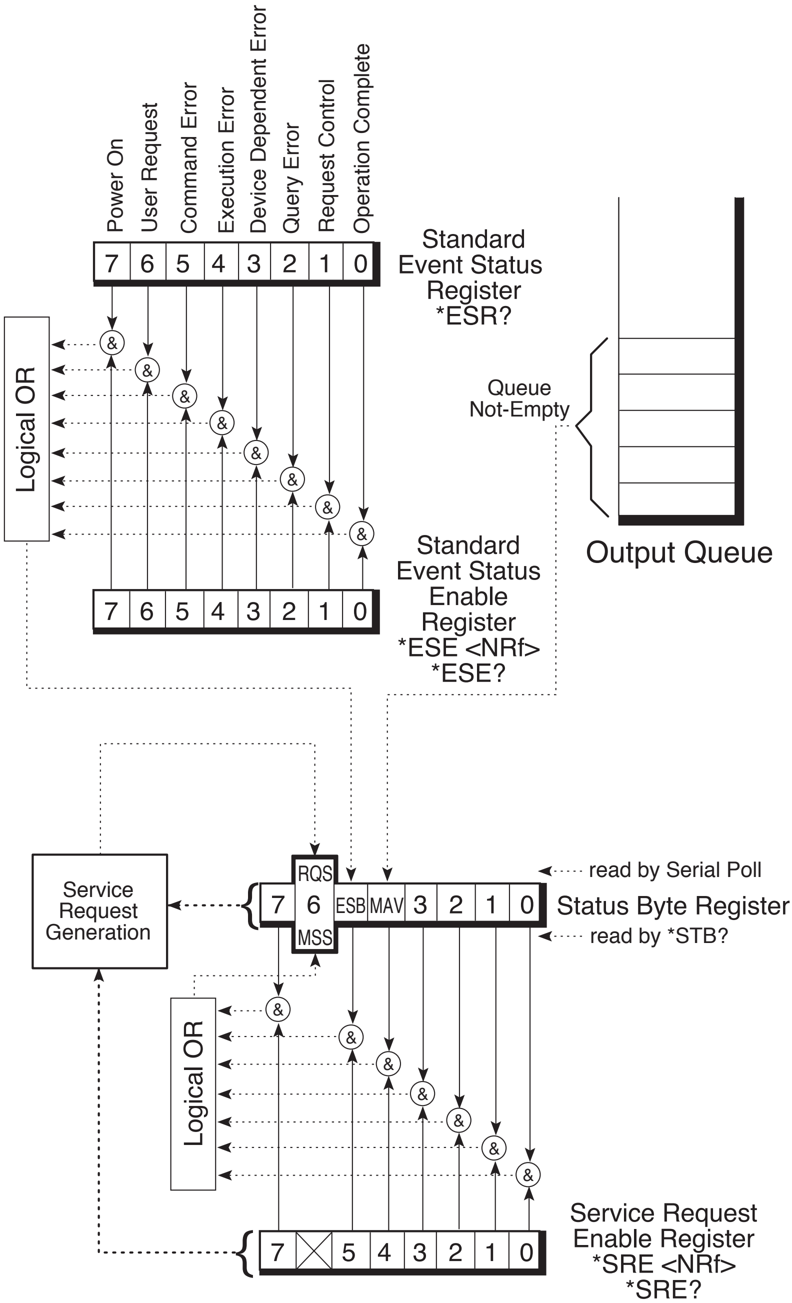


Figure A-3 488.2 Required Status Reporting Capabilities

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.

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

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

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.

A

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

A

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

A

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 COMMUNICATION 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.

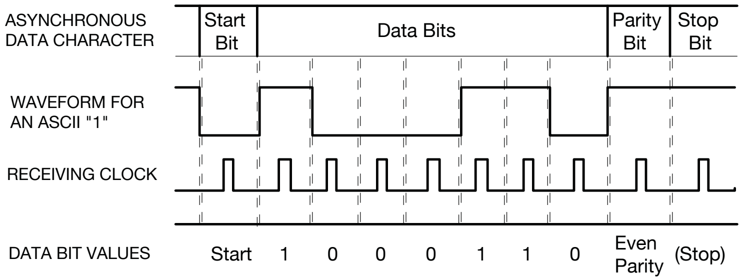


Figure A-7 Asynchronous Data Character Waveforms

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

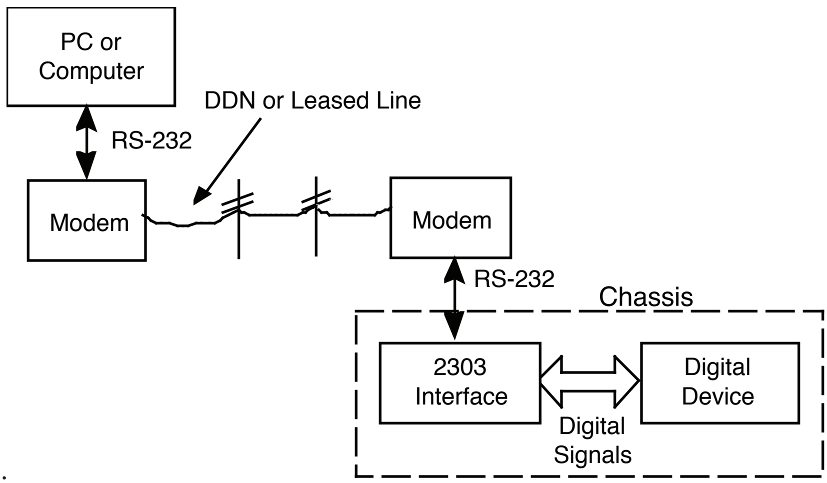


Figure A-8 Long Distance Communication using Modems

A2

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.

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.

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.

A2.2 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).

Electrically, RS-232 signals are bipolar and are referenced to a common ground (AB) on pin 7. Transmitted signals must be between +5 and +15V 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)

A2

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.

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-

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' 3 and 5 missing
   '3','5' 1, 2 and 4 missing

A2

A3 GPIB CONNECTOR/SWITCH BOARD ASSEMBLIES

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 ICS board products, 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. Rocker 6 sets address 0. Rockers 7 and 8 are normally not used.

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:

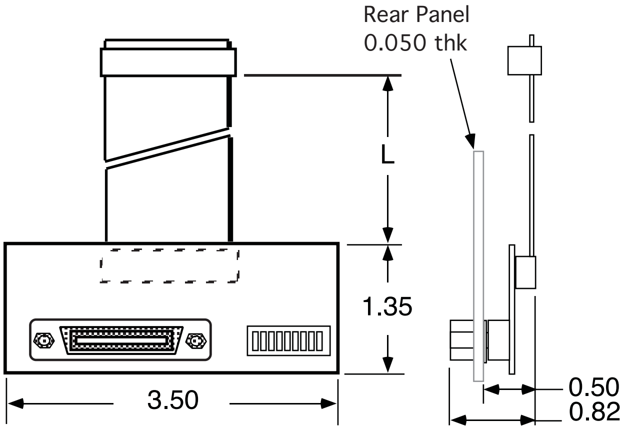
Type	Part Number
Horizontal Conn./Sw Assy with 90 cm long cable	113640-90
Vertical Conn./Sw Assy with 90 cm long cable	113642-90

A3

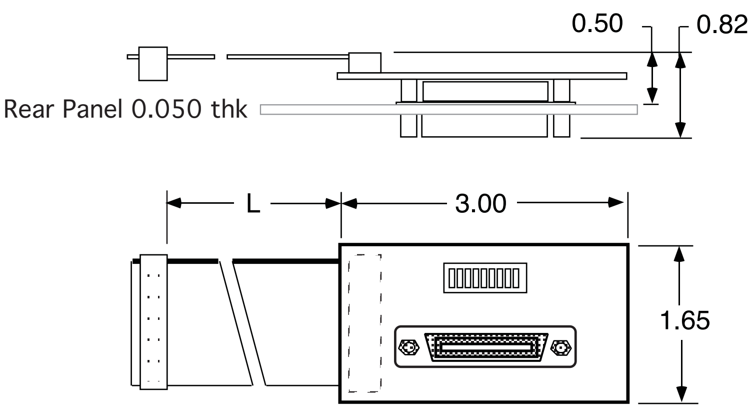
A3.2 Board Installation

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-12



(a) Horizontal/Connector Assembly



(b) Vertical Connector/Switch Assembly

Figure A-9 GPIB Connector/Switch Board Assemblies

2. Locate a blank area on the host chassis rear panel as shown in Figure A-11. Leave enough room inside for the flat ribbon cable bend radius.

3. Machine the cutouts shown in Figure A-12. If the rear panel is thicker than 0.050 inches, mill a 2.3 in x 0.8 in relief on the inside of the panel for the Amphenol 57-20240 connector.

A3

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 as shown in Figure A-9.

5. Route the flat ribbon cable to the board 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.

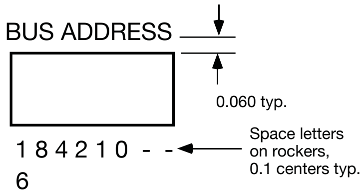
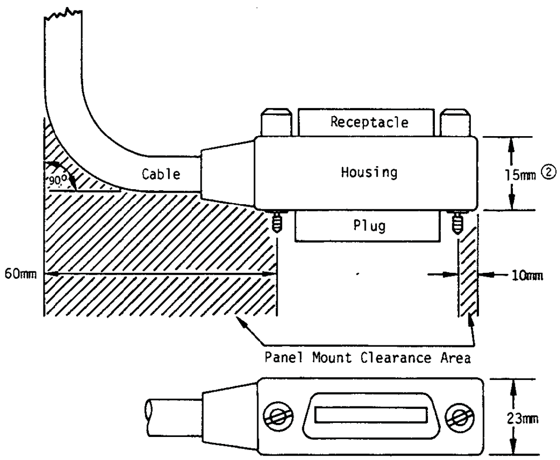
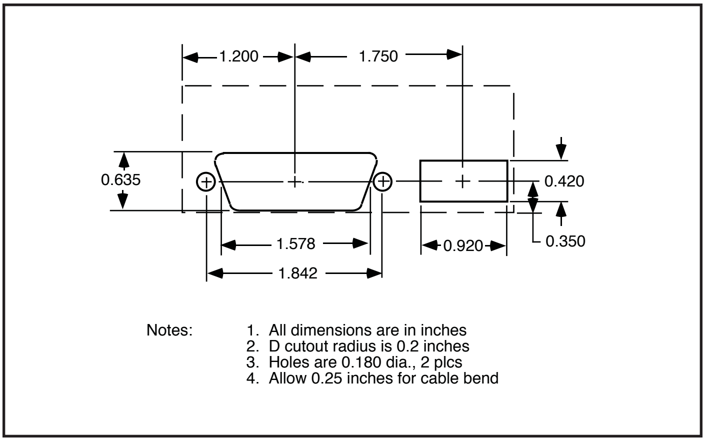


Figure A-10 Switch Silkscreen Detail

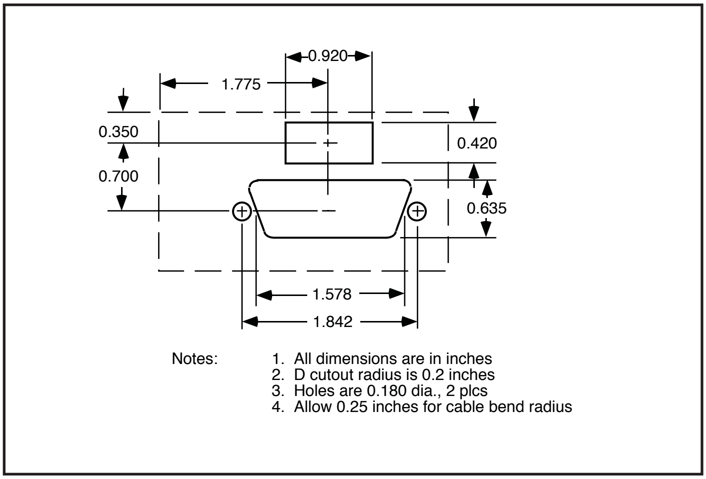


NOTES: (1) All measurements are typical.
(2) Length of lock screw is a function of this dimension.

Figure A-11 IEEE-488.1 Figure 21



(a) Horizontal Connector/Switch Assembly Mounting Dimensions



(b) Vertical Connector/Switch Assembly Mounting Dimensions

Notes: If the rear panel is thicker than 0.050 inches, mill a relief on the reverse side for the Amphenol 57-20240 connector. Suggested relief is 2.3 x 0.8 inches, centered on the connector cutout. Maximum depth is 0.090 inches.

Figure A-12 GPIB Connector/Switch Panel Cutouts

A3

This page intentionally left blank

A3

Index

Symbols

I

I

I

L

M

I

O

P

Q

R

I

This page left intentionally blank

I