ICS ELECTRONICS

MODEL 4863
GPIB ⟺ Parallel Interface
MODEL 2363
Serial ⟺ Parallel Interface

4863/2363

MODEL 4863
GPIB↔Parallel Interface
Model 2363
Serial↔Parallel 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 120143
October 2006 Edition Rev 9

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 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 4863 and 2363 have 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.

CE Certificate of Conformance reproduced in Figures 1-7 and 1-8.

TRADEMARKS

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

HP is a trademark of Hewlett-Packard Corporation, Palo Alto, CA
ICS is a trademark of ICS Electronics div of Systems West, Inc., Pleasanton, CA

REVISIONS

Contents

This page left intentionally blank

1

General Information

1.1 INTRODUCTION

This section provides specifications for ICS's Models 4863 and 2363 Parallel Digital Interface Modules. All specifications and functional descriptions apply to both units unless stated otherwise.

1.2 DESCRIPTION

The Model 4863 GPIB <-> Parallel Interface is an IEEE 488.2/GPIB/HP-IB compatible device that provides digital signals to drive external devices and to input digital data onto the GPIB bus from external devices. The Model 2363 Serial <-> Parallel Interface provides the same digital I/O capability and GPIB functionality but with RS-232 and RS-422/RS-485 serial interfaces.

Both modules provide 48 programmable digital lines with TTL levels for controlling devices or for data transfer. The digital interface is user configurable into gated inputs and/or latched outputs in eight bit increments. As inputs, each data line has a pullup resistor for sensing contact closures or for interfacing with CMOS signals. As outputs, each line can source 24 mA or sink up to 48 mA. The interfaces can also monitor 15 of the digital lines and report any changes to the Bus Controller. The 4863 transfers data either as the parameter in a command or as a transparent data string. Data can be formatted as decimal or hex characters or unformatted binary bytes. Applications include interfacing devices with parallel digital interfaces to the GPIB bus or to serial links, controlling discrete devices, reading the state of external switches or signals, and monitoring digital events.

1

Handshake lines are provided for transferring data to external devices and for inputting data. Other control lines include a Remote line for enabling external controls, and Trigger, Clear and Reset pulse outputs.

The 4863 and 2363 uses SCPI commands (Standard Commands for Programmable Instruments) to set their configuration parameters and address. The configuration settings can be retained in nonvolatile memory and recalled automatically at power turn-on. This means that the unit has to only be configured once and then left alone until used in another application. The SCPI commands can also be used to query the current parameter settings. Refer to the appendix for additional information about SCPI commands.

At power turn-on, the modules perform a self test of their internal logic. At the end of the self test, they momentarily display their GPIB or serial address by blinking the front panel LEDs before displaying their normal Power-on - Ready condition.

The 4863 is a member of ICS's Minibox product family and is packaged in a small metal case that is less than 1U (1.6 inches) in height. The front panel contains the power switch and LEDs which indicate the unit's status. The rear panel contains the GPIB connector, the Relay Contact/Digital Input connector and a DC power jack. The 4863 accepts a wide range of DC voltages and is shipped with an AC adapter for the local power lines.

The Model 2363 is similar to the Model 4863 except that the GPIB connector on the rear panel has been replaced with a connector that contains the RS-232 and RS-485 serial signals. Because of the nature of the serial interface, the 2363 only transfers formatted data as a command parameter and does not have the 4863's transparent or binary data transfer capability. The 2363 are network capable and up to sixteen 2363s can operate on an single RS-485 network. All of the 4863's functional descriptions apply to the 2363 unless otherwise stated.

1.3 MODEL 4863/2363 SPECIFICATIONS

1.3.1 4863/2363 Model Numbers and Options

The following specifications apply to all 4863 and 23633 models. Options for your unit may be found by comparing the list below to those listed on the serial label on your unit.

4863 - X General Model Number

Option Codes

-1 Special Crystal
-3 Custom Paint
-5 Custom Front Panel
-6 Special Settings
-7 Special Program
-8 Hardware Modification

-A Ship with Australian 230 Vac Adapter
-B Ship with British 230 Vac Adapter
-E Ship with European 230 Vac Adapter
-J Ship with Japanese 100 Vac Adapter

Base model number

4863 for GPIB Interface Module
2363 for Serial Interface Module

Shipped with US 115 Vac Adapter unless otherwise specified.

1

1.3.2 OEM Board Part Numbers and Configurations

OEM Board versions of the 4863 and 2363 are available in the configurations shown in Table 1-1. On OEM Boards, the GPIB connector is a vertical header that mates with one of ICS's GPIB Connector/Address Switch Assemblies which may be used to mount an Address Switch and a GPIB Connector on the user's rear panel. (See Appendix A) An LED header is provided to remote the LEDs and the power switch is removed. A separate header is provided for the optional serial interface. The Digital I/O connector remains the same. Refer to Figure 1-6 for the OEM Board layout.

Most of the options listed in paragraph 1.3.1 can be applied to the board product by adding the dash number after the part number.
e.g. 114504-11-7 specifies a 114504-11 Board with a custom program.

TABLE 1-1 OEM BOARD CONFIGURATIONS

Part Number	Number of Digital Signals	Interfaces GPIB	RS-232	RS-485
114504-11	48	Yes	No	No
114504-12	48	No	Yes	Yes
114504-13	48	Yes	Yes	Yes

1.4 IEEE 488 INTERFACE

1.4.1 488.1 Capabilities

1

The 4863's 488 Bus interface meets the IEEE STD 488.1-1987 standard and has the following capabilities:

SH1, AH1, T6, L4, SR1, PP0, DC1, RL0, DT1, C0 and E2 drivers.

1.4.2 GPIB Addressing

The 4863 has three, user selectable address modes. In the Single address mode, the 4863 responds to a single primary address for all commands. In the Dual mode, the 4863 responds to two consecutive primary addresses. The lower address is used for commands and setting the device's address, the upper address is used for transparent data transfer. In the Secondary address mode, the 4863 responds to a single primary address and secondary addresses 00 and 01. The lower address is used for commands, the upper address is used for transparent data transfer. Valid primary addresses are 0 to 29 for the Dual address mode, 0 to 30 for the other modes.

1.4.3 Serial Poll Responses and SRQ Generation

The 4863's Serial Poll Responses and SRQ generation uses an IEEE-488.2 type reporting structure that has been enhanced by the addition of Questionable and Conditional Registers for reporting the digital input signal status and monitoring signal changes. The enhancement conforms to the SCPI 1995.0 Specification.

1.4.4 488.2 Common Commands

The 4863 conforms to IEEE STD 488.2-1987 and responds to the following 488.2 commands:

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

1

1.4.5 SCPI Parser

The 4863 and 2363 include an extended SCPI parser that complies with the SCPI Standard Version 1994.0.

1.4.6 Buffers

The 4863 and 2363 use the same input and output buffers. Buffer size is 1024 bytes.

1.5 SERIAL INTERFACES (2363)

Model 2363s have their RS-232 and RS-485 signals on an RS-530 compatible 25-pin female connector on the rear panel. The 2363 automatically responds to the serial port that received the last message.

1.5.1 Baud Rate and Character Format

Baud Rate	Any rate from 300 to 115,200 baud. Parser selects closest rate to specified rate when non-standard rate entered. Standard rates are: 300, 600, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 57600, 76800, 92160 and 115200 baud.
Format	7 or 8 data bits; odd, even or no parity;
Stop Bits	1 or 2 stop bits

1.5.2 RS-232 Interface

The RS-232 Interface has the following characteristics:

Signals	BA, BB and AB
Mode	Full Duplex with or without echo
Flow control	none
Control-E	Sets echo on
Control-F	Sets echo off

1.5.3 RS-485 Interface

The RS-485 Interface may be used in 2-wire, RS-422 or RS-485 point-to-point or network applications and has the following characteristics:

Signals	Tx/Rx+ and Tx/Rx- signal pairs
Mode	Half Duplex
Termination	220 ohm load resistor with 1 kohm pullup and pulldown resistors
Prompts	CR-LF-prompt for terminal control LF-prompt for computer control
Protocols	None, Addressed and Packet Protocol
Addresses	16

1

1.5.4 Buffers

Input Buffer 1024 bytes

1.5.5 RS-232 Transmission Protocol

1.5.5.1 Echos and Prompts

The 2363 returns a prompt when ready for the next command. If Echo is enabled, the prompt character is lined along the left edge of the screen. The CNTL-E and CNTL-F commands turn echo on and off and change the prompt sequence and message terminators as follows:

Command	Echo	Message	Terminator	Prompt
CNTL-E	On	Response	CR LF	CR LF >
CNTL-F	Off	Response	LF	> LF

The user should not attempt to send a new command until it has completed the current command. Echo mode changes are immediate.

1.5.5.2 Service Request Message (SRM)

The 2363 emulates the IEEE-488.1 SRQ line by transmitting an asynchronous Service Request Message (SRM) when it is not busy processing a message and when an enabled Status Byte bit becomes set. The SRM format is:

SRM n <LF> **w**here n is the decimal value of the Status Byte Register

e.g. **SRM 96** indicates bits 5 and 6 are on.

1.5.6 RS-485 Basic Protocol

When the Network mode is set to Off, the RS-485 interface operates as does the RS-232 interface with Echo OFF. All responses messages are terminated with a linefeed character (LF). A '> LF' prompt is sent after each command or after the response message. The transmitter is enabled until the prompt is sent. SRM messages are sent when enabled.

1.5.7 RS-485 Network Address Protocol

When Network Address is enabled, the first two characters of each received message are checked for an address character that matches the address set in the unit's Flash memory. If a valid address is detected, then the unit responds normally to the message. The Address character sequence is an STX character (02) followed by the address character. The address character is the ASCII number (0-9 :;<=>?) with a hex value of 30 - 3F. An example is the IDN query sent to a 2363 at address 4.

i.e. **STX 4 * I D N ? (LF)**

Prompts are supported when network addressing is enabled. The transmitter is enabled until the prompt is sent. SRM messages are inhibited when Network Address is selected

1.5.8 RS-485 Packet Protocol

When Network Packet protocol is enabled, the 2363 always responds to valid packets sent to it so the user has confirmation the unit received the message. All message packets include an address character, the command or query and a checksum. The 2363 responds to packets containing an address character that matches the address set in its Flash memory. If the packet checksum and command is valid, the command is executed and an ACK or response packet is returned. If the checksum or command is invalid, a NAK response is returned. ACK/NAK command response packets contain the current ESR Register value.

The general packet format is:

STX Addr Message Characters....ETX Checksum

An acknowledgment packet is:

ACK Addr ESR_Register_Value ETX Checksum

ACK Addr Query_response ETX Checksum

An NAK packet is:

NAK Addr ESR_Register_Value ETX Checksum

Where:

ASCII characters 0 to 1F hex are reserved for message control.
ASCII characters 20-7F are for address and data.
Address character is 30-3F hex.
STX is ASCII character 02 hex.
ETX is ASCII character 03 hex.
ACK is ASCII character 06 hex.
NAK is ASCII character 15 hex.
Checksum is the exclusive OR sum of all of the characters from the STX character though the ETX character.

1.6 DIGITAL SIGNAL SPECIFICATIONS

The 4863/2363's Digital I/O signals have the following specifications:

1.6.1 Data Lines

Number	48 with internal 33 Kohm pullups to + 5 Vdc	
Input	High	>=2.4 Vdc or open circuit
Levels	Low	< 0.5 Vdc at 200 µA
Output	High	> 3.0 Vdc with 3 mA source
Levels		> 2.0 Vdc with 24 mA source
	Low	< 0.55 Vdc with 48 mA sink

1.6.2 Data Input

Data may be read after receipt of an External Data Ready signal (EDR) if Talk handshaking is enabled or anytime if Talk handshaking is disabled. Inhibit responds < 0.1 µsec after EDR edge. Figure 1-1 shows data loading and GPIB bus output times for 6 HEX characters. Active EDR edge and Inhibit signal polarities are selected by configuration commands. The EDR F/F is reset when the data is read, or by the SENSE:RESET:EDR command or by a Device Clear. Standard units only use EDR #1 input. Both EDR inputs have 33 Kohm pullups to + 5 Vdc. Times are listed in Table 1-2.

EDR	High	>=2.4 Vdc or open circuit
Input	Low	< 0.5 Vdc at 200 µA
Inhibit	High	> 2.4 Vdc with 4 mA source
Output	Low	< 0.55 Vdc with 16 mA sink

1.6.3 Output Data and Data Strobe

Data may be placed in the output latches by a port command or by string commands. A strobe pulse is automatically generated after data is placed in the output latches by a data string or in response to a STRobe command as shown in Figure 1-2. Times are listed in Table 1-2. Strobe signal levels same as the output levels in paragraph 1.6.1.

If data is outputted with a port command, data polarity is set by a port command. If data is outputted with a string command, data groupings, data polarity and strobe polarity are set by the configuration commands.

1

1.6.4 Monitored Digital Inputs

The first fifteen Digital I/O lines (CH1-15) are sampled at an approximate 1 kHz rate and the values placed in the Questionable Register in the 4863/2363's IEEE-488.2 Status Reporting Structure. Changes may be used to generate a Service Request. The digital input lines are reported at the following bits in the Questionable Register:

CH#	-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

1.6.5 Trigger Output

Trigger pulse generated by 488.2 *TRG or 488.1 GET commands. Pulse polarity is defined by configuration command. Signal levels same as output levels in paragraph 1.6.1.

1.6.6 Reset Output

Reset output is pulsed by 488.2 *RST or *RCL command and when the unit is powered on or reset. Pulse polarity is defined by a configuration command. Signal levels same as output levels in paragraph 1.6.1.

1.6.7 Remote Output

Remote output level is asserted when the unit is in the GPIB Remote state. Signal polarity is defined by a configuration command. Signal levels same as output levels in paragraph 1.6.1. The 4863 will deassert the Remote output in response to a true Status B input if the unit is not in the GPIB Local Lockout State.

1.6.8 External Reset Input

The -ExtRst is a low true input with a 10 Kohm pullup that is used to reset the 4863/2363. The 4863/2363 is held in the reset state while the -ExtRst input is low. Logic levels are same as the input levels in paragraph 1.6.1.

1.6.9 Status Inputs

Two external device status inputs with 33 Kohm pullups to + 5 Vdc. Inputs sampled at an approximate 1 kHz rate and the states placed in the Operational Register in the 4863/2363's IEEE-488.2 Status Reporting Structure. Status logic polarity is defined by a configuration command. Input logic levels are same as the input levels in paragraph 1.6.1.

Status A may be used for output data handshaking. If output handshaking is enabled, data is only output and the Strobe line pulsed when the Status A input is true. A false input inhibits data transfer.

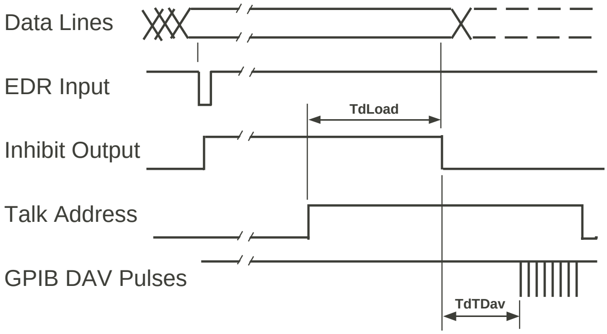
Status B may be used as a request for the 4863 to go to the local state. The 4863 will deassert the Remote output in response to a true Status B input if the unit is not in the GPIB Local Lockout State.

1.6.10 LED Driver Outputs

Five, low true driver outputs for driving remote LEDs. Maximum sink current is 15 mA at 0.5 Vdc.

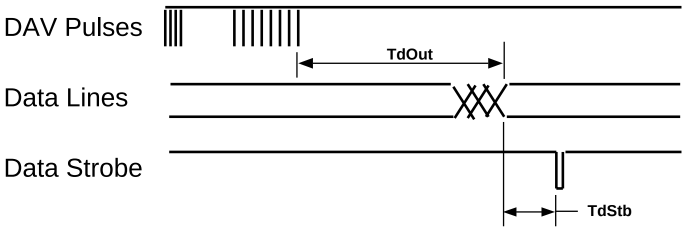
1.6.11 Timing Diagrams

Figure 1-1 shows input data handshake timing from setting the EDR input until the data is placed on the GPIB bus. Figure 1-2 shows output data from command terminator until data is present on the output lines. Figure 1-3 shows how the Digital I/O lines are configured after a power turn-on or when the board is reset. The time values are listed in Table 1-2.



TdLoad is time to load data after being addressed to talk
Data must remain stable until Inhibit goes false
TdTDav is the time to output first byte on GPIB Bus

Figure 1-1 Input Data Timing



TdOut is delay from command terminator to data output
TdStb is the delay from last byte output to the data strobe

Figure 1-2 Output Data Timing

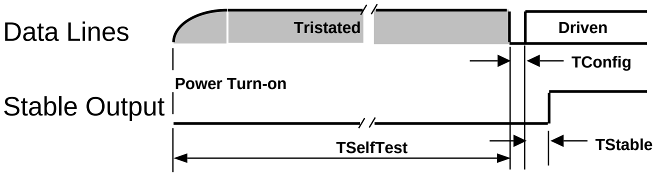


Figure 1-3 Data I/O Lines at Power Turn-on

1.6.12 Timing Chart

The times in Table 1-2 are for a 4863 with a 20 MHz clock.

TABLE 1-2 4863 TIMING

Symbol	Command/Notes	Time
TdLoad	Loading time for 6 hex characters	4.0 ms
TdTDAV	Delay to first Talk DAV Pulse	0.7 ms
TdOut	Terminator to data stable	
	"SOURCE:DATA hhhh"	5.0 ms
	":DATA hhhh"	3.6 ms
	"PO hhhh"	1.5 ms
	"hhhh" (dual address mode)	0.1 ms
	"bb" (binary format, dual address)	70 µs
TdStb	Strobe delay after data stable	100 µsec
Strobe	Data Strobe Pulse Width	5 µsec
TdSTB	Delay from comd terminator to pulse	
	"SOUR:DATA:STR"	7.5 ms
	":DATA:STR"	6.3 ms
	"SP"	0.8 to 1.1 ms
TdStb2	Delay to next data strobe	
	"SOURCE:DATA hhhh,hhhh"	140 µs
	":DATA hhhh,hhhh"	140 µs
	"PO hhhh,hhhh"	140 µs
	"hhhh,hhhh" (dual address mode)	160 µs
	"bbbb" (binary format, dual address)	19 µs
TdRST	Delay from comd terminator to pulse	7 ms
Reset	Reset Pulse Width	65 µsec
TdTRG	Delay from comd terminator to pulse	
	"*TRG"	1.8 ms
	GET	50 to 200 µs
Trigger	Trigger Pulse Width	5 µsec
TSelfTest	Pon to Digital I/O lines configured	350 ms
TConf	Byte configuration glitch	5 µs max
TStable	Last byte configured to Stable high	5 µs min

NOTE: Multiply above values by 1.35 for the Model 2363.

1

1.7 CONFIGURABLE FUNCTIONS

Table 1-3 lists the 4863's configuration parameters and their factory settings. Table 1-4 lists the 2363's factory settings. The configuration parameters are saved in nonvolatile Flash memory.

TABLE 1-3 4863 CONFIGURATION PARAMETERS

Command	Function	Factory Setting
:ADDRESS	Sets 4863's GPIB bus address	4
:EXTernal	Enables external address switch	OFF #
:MODE	Sets GPIB address mode	SINGle
:INPut	Sets number of Talk bytes	6 #
:POLarity	Sets Input polarity	1 #
:HANDshake	Enables Input Handshaking	ON #
:TALK	Selects Input Format	HEX #
:TRANslation	Sets input conversion table for BCD HEX characters	see note #
:OUTput	Sets number of Listen bytes	0 #
:POLarity	Sets Output polarity	1 #
:LISTen	Sets Output Format	HEX #
:HANDshake	Enables Output Handshaking	OFF #
:CLEar	Sets Clear pulse active level	0 #
:EDR	Sets EDR input active level	0 #
:INHibit	Sets Inhibit output active level	1 #
:REMote	Sets Remote output active level	0 #
:RESet	Sets Reset pulse output active level	0 #
:STRobe	Sets Data Strobe pulse active level	0 #
:TRIGger	Sets Trigger pulse active level	0 #
:ASTATus	Sets Status A input active level	1 #
:BSTATus	Sets Status B input active level	0 #
:LOCK	Blocks # items from changes	Off
:IDN	Sets user's IDN message	ICS IDN msg #
:QUES Reg	Enables digital signal changes	0 *
:PTRANSISTION	Enables positive going inputs	All 1s *
:NTRANSISTION	Enables negative going inputs	All 0s *
*ESE	Enables events to generate an SRQ	0
*SRE	Enables Summary bits to generate an SRQ	0

* Indicates parameters that cannot be saved in the unit's nonvolatile memory.
# Indicates parameters that may be blocked by the LOCK command.
Std Conversion Table is 0123456789ABCDEF.

TABLE 1-4 2363 CONFIGURATION PARAMETERS

Command	Function	Factory Setting
:BAUD	Sets 2363's baud rate	9600
:PARity	Sets 2363's serial parity	NONE
:BITS	Sets 2363's data bits	8
: :SBITs	Sets 2363's stop bits	1
: :NETwork	Sets RS-485 protocol	OFF #
:ADDRess	Sets 2363's address for network use	4 #
:INPut	Sets number of Talk bytes	6 #
:POLarity	Sets Input polarity	1 #
:HANDshake	Enables Input Handshaking	ON #
:TALK	Selects Input Format	HEX #
:TRANslation	Sets input conversion table for BCD HEX characters	see note #
:OUTput	Sets number of Listen bytes	0 #
:POLarity	Sets Output polarity	1 #
:LISTen	Sets Output Format	HEX #
:HANDshake	Enables Output Handshaking	OFF #
:CLEar	Sets Clear pulse active level	0 #
:EDR	Sets EDR input active level	0 #
:INHibit	Sets Inhibit output active level	1 #
:REMote	Sets Remote output active level	0 #
:RESet	Sets Reset pulse output active level	0 #
:STRobe	Sets Data Strobe pulse active level	0 #
:TRIGger	Sets Trigger pulse active level	0 #
:ASTATus	Sets Status A input active level	1 #
:BSTATus	Sets Status B input active level	0 #
:LOCK	Blocks # items from changes	Off
:IDN	Sets user's IDN message	ICS IDN msg #
:QUES Reg	Enables digital signal changes	0 *
:PTRANSISTION	Enables positive going inputs	All 1s *
:NTRANSISTION	Enables negative going inputs	All 0s *
*ESE	Enables events to generate an SRQ	0
*SRE	Enables Summary bits to generate an SRQ	0

* Indicates parameters that cannot be saved in the unit's nonvolatile memory.
\# Indicates parameters that may be blocked by the LOCK command.
Std Conversion Table is 0123456789ABCDEF

1

1.8 INDICATORS

The 4863/2363 has six front panel LEDs that display the following conditions:

4863 /	**2363**		
PWR	PWR	-	Indicates power on.
RDY	RDY	-	Indicates unit has passed self test.
TALK	TX	-	Indicates unit has recognized its talk address or is sending serial data
LSTN	RX	-	Indicates unit has recognized its listen address or is receiving serial data.
SRQ	SRV	-	Indicates unit is generating an SRQ or has sent a Service Request Message.
ERR	ERR	-	Indicates unit has detected an command error and the command was not executed. At power turn-on, the ERR LED indicates the unit is uncalibrated. See the CAL:DATE command.

When the 4863 or 2363 is turned on, it performs an internal self test which takes about 0.3 seconds, during which time the PWR indicator is on. At the end of the self test, the unit displays its current GPIB or network address by blinking it address on the front panel LEDs for a half a second. If the 4863 is set for dual secondary addresses, the address is blinked twice. The LED bit weights are:

RDY	TALK/TX	LSTN/RX	SRQ/SRV	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 Table 5-3 for a description of the self test errors and their possible causes. When all of the LEDs **except the ERR LED** blink on for 1 second, the unit is writing configuration parameters to the Flash memory. When all of the LEDs blink on for 1 second, the unit is initializing the Flash configuration table to factory defaults. The LED remote header (J7 on the OEM board versions) provides low true signals that sink up to 15 mA for operating a second set of LEDs.

1.9 PHYSICAL

1

Size		
4863/2363 Minibox	-	7.45" L x 7.29" W x 1.52" H (18.92 cm L x 18.52 cm W x 3.86 cm H) (See Figure 1-5)
114504 Board	-	7.0" L x 7.0" W x 1.26" H (17.78 cm L x 17.78 cm W x 3.20 cm H) (See Figure 1-6)
Weight	-	3 lbs (1.4 kg) including adapter (4863)
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 (4863)
Power	-	9 to 32 Vdc @ 3 VA (unloaded outputs)
Connectors	-	IEEE 488 Interface Amphenol 57-20240 with metric studs
	-	Serial - DB-25S connector with lock studs
	-	Digital Interface DC-62S female connector with locking studs. Mates to DHS-62P or any metal shell DC-62P type connector.

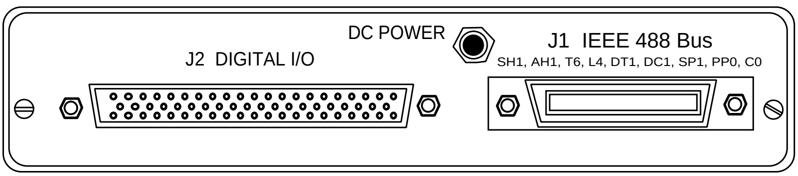


Figure 1-4 4863 Rear Panel

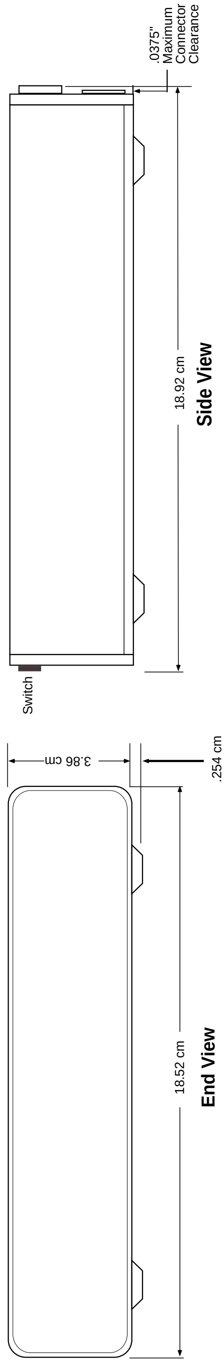


Figure 1-5 Outline Dimensions

1

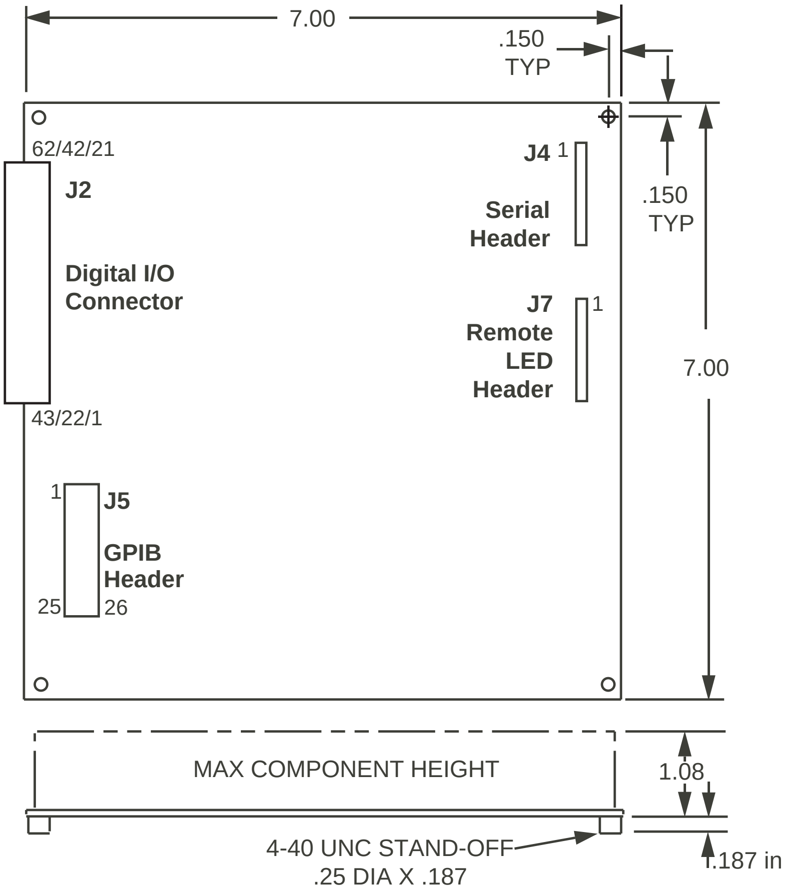


Figure 1-6 4863 OEM Board Outline Dimensions

1.10 CERTIFICATIONS OR APPROVALS

EMI/RFI 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 Compliances reproduced in Figures 1-7 and 1-8.

UL/CSA/VDE AC Wall adapter has applicable UL/CSA/VDE and CE approval.

Declaration of Conformity

Application of Council Directive(s)..89/336/EEC

Standard(s) to which Conformity is Declared.............EN 55022, EN 50082-1

Manufacturer's Name ICS ELECTRONICS CORPORATION

Manufacturer's Address 473 LOS COCHES STREET
MILPITAS, CA 95035-5422

Importer's Name

Importer's Address

Type of equipment GPIB TO DIGITAL INTERFACE

Model No. 4863

Serial No. ________ Thru ________ Year of Manufacture 1996

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directive(s) and Standard(s)

Place Milpitas California USA

(Signature)

Date 17 JULY 96

G. K. MERCOLA
(Full Name)

President
(Position)

Figure 1-7 4863 Certificate of Compliance

EMV Meßhaus Landsberg GmbH

EMV Meßhaus Landsberg GmbH
Erpftinger Straße 31, Gebäude 5 Süd
D-86899 Landsberg/ Lech

Certificate Of Conformance
89/336/EEC
EMC Directive

17.03.1997

Customer: ICS Electronics
473 Los Coches Street
Milpitas, CA 95035-5422

We certificate the compliance of the representative test sample with the below listed and relevant standards.

Producer: ICS Electronics

Tested Type: Model 2363-E, S/N 702160, Rev. 2.3

Standards: EN55022, DIN VDE 0877 Teil 3, VDE0878 T30, EN 50082-2

17.03.1997
(Date Of The Test)

(Dieter Graf)

Berichtsnummer: ICS-00297 Seite: 2 von 9

Figure 1-8 2363 Certificate of Compliance

1

1.11 INCLUDED ACCESSORIES

All Minibox units include the following accessories. OEM board versions are shipped without any accessories.

Instruction Manual
Mating 62-pin connector and hood
AC Wall Adapter, with applicable country plug
Support CD-ROM

1.12 OPTIONAL ACCESSORIES

1.12.1 4863/2363 Optional Accessories

114508	4863 Open End Cable, 5 feet long
114212	Single Large Minibox Rack Mount Kit
114213	Dual Large Minibox Rack Mount Kit
114227	Large/Small Minibox Rack Mounting Kit
104705 - 104740	IEEE 488 Bus Cables, multi-shielded design in 0.5, 1, 2 and 4 meter lengths, standard connectors.
104505 - 104520	IEEE 488 Bus Cables, multi-shielded design in 0.5, 1, and 2 meter lengths with a straight-in connector at one end.

1.12.2 OEM Board Accessories

120143	4863/2363 Instruction Manual
123021	Support CD-ROM
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 (L= cable length in CM, 90 CM standard)
125076	Kit - Mating Serial Header Connector
902270	Mating Digital I/O Connector for 4863 OEM Board
902105	Hood (Plastic) for 902270 Digital I/O Connector

2

Installation

2

2.1 INTRODUCTION

This section provides the user with directions for shipment verification, for configuring the unit and for connecting it to other devices. All directions apply equally to the 2363 unless stated otherwise.

2.2 SHIPMENT VERIFICATION

When unpacking, check the unit 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.

Verify that the following items were included with your unit and that you have the correct model dash number:

(1) Model 4863 or Model 2363 Parallel Interface
(1) AC Power Adapter
(1) Mating Digital I/O Connector and hood
(1) 4863/2363 Instruction Manual
(1) Support CD-ROM

4863 and 2363 OEM boards include the mating Digital I/O connector, Instruction Manual and Support CD-ROM.

2.3 FACTORY CONFIGURATION

When shipped, the 4863 and 2363 are set to the configurations listed in Tables 1-3 and 1-4. The configurable parameters are stored in Flash memory and can be queried and changed by the user.

2.4 CONFIGURATION DIRECTIONS

2

Configuring a 4863 or 2363 is a multistep process. First design the connections between the digital interface and the external device as directed in Section 2.8. This will provide you with the signal-pin assignments for the digital interface, byte data direction and the data transfer method(s) to be used in your application. Review the SCPI Command Tree in Section 3 to select the configuration values for the your data transfer methods. If the 4863's Status Structure is being used, it will have its own settings. Follow the instructions in this section to perform the actual configuration.

If you are using a PC as a bus controller, the easiest way to configure the unit is to use ICS's GPIBkybd program or a Minibox configuration program. The GPIBkybd program or any similar interactive program lets you enter and query all of the configuration settings and is the most flexible configuration method. The Minibox configuration programs guide you through a menu-driven configuration procedure. The menu procedure is easier to use but it is restricted to a limited number of commands. The Support CD-ROM contains both programs for the more popular GPIB controller cards and a configuration program which uses a PC's COM ports to configure 2363s. Section 2.5 describes how to use the Minibox configuration programs to change the unit's configuration.

Section 2.6 provides general instructions for configuring the 4863/2363 from the GPIBkybd program or from your own program. Section 2.6 also includes program examples in HP BASIC that can be easily adapted to your favorite programming language. If you are writing a test application program, it is a good idea to include your configuration commands in the program, at least as an option. This way, the program will still run if the 4863/2363 is ever replaced or repaired.

If you have an HP 9000 series computer that runs Rocky Mountain Basic and has a HP-IB interface, you can adapt the program shown in Figure 2-2 to configure the 4863.

2.5 THE SUPPORT CD

The Support CD-ROM contains Configuration Programs, useful utility programs and example programs for ICS's interface products. The Configuration Programs folder contains DOS and Windows configuration programs that walk the user through a menu to configure the card's power-on condition. The following configuration programs run on Intel type PCs and operate the following GPIB Controller cards and interfaces:

Windows	**Supported GPIB Controllers**
niconf_w.exe	ICS 488-USB, 488-PCI, 488-PCILt, 488-PCM, and 488-cPCI Cards, Measurement Computing or National Instruments Cards. See para 2.6
GPIBkybd.exe	ICS 488-USB, 488-PCI, 488-PCILt, 488-PCM, and 488-cPCI Cards, Measurement Computing or National Instruments Cards. See para 2.7
DOS	**Supported card or Interface**
mconfig.exe	ICS 488-PC2 Card National Inst. GPIB-PC2a Card (Set to address 2E1 and to 7210 emulation) or an NEC 7210 compatible GPIB Controller Card that is set to 2E1.
niconf.exe	National Instruments AT-GPIB Card (Set to address 2C0H)
hpconf.exe	Hewlett-Packard HP-IB Card (Set to address DC000)
sconfig.exe	PC COM1 or COM2 port

The niconf_w.exe program is in the niconf_zip.exe file on the Configuration Programs folder on the Support CD-ROM. To recover the program file, create a new directory and copy the niconf_zip.exe file to it. Double-click on the file or use Windows Zip-it utility to run the self-extracting zip file and extract the niconf_w.exe program file. The DOS programs are in the DOS directory.

2.6 USING THE CONFIGURATION PROGRAM

This section describes how to use the niconf_w.exe configuration program but the directions apply to the other configuration programs.

1. First copy the configuration program to a directory on your hard drive as directed in 2.5.

2. Connect the 4863 to the GPIB controller card in the PC as shown in Figure 2-1. Plug the AC adapter plug into the DC jack in the 4863's rear panel. Plug the AC adapter into an AC outlet.

   or

   Connect the 2363 to the COM port in the PC with a Serial Cable (See Section 2.7). Plug the AC adapter plug into teh DC jack in the 4863's rear panel. Plug the AC adapter into an AC outlet.

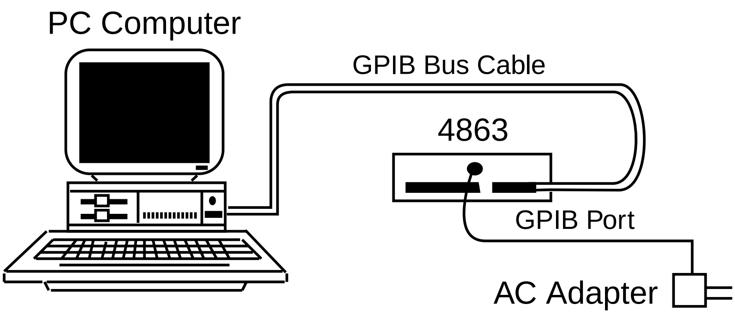


Figure 2-1 4863 Configuration Setup

3. Turn the 4863/2363 Power switch on.. The PWR LED will come on while the unit performs its self test. When the test is done, the unit will blink its current address setting before turning on the RDY LED.

4. Double click on the niconf_w.exe or use the RUN command to start the configuration program. The Main form or menu will appear.

5. In the Model window, select the model you wish to configure. DOS programs will display a list of model numbers. Enter the number

that corresponds to the model that you are configuring and press Return

e.g. **4863 <return>** for the Model 4803
2363 <return> for the Model 2303

6. Cycle Power

The program may ask that you turn the unit's power off and on. Press Return when the unit has finished its self test.

7. GPIB Address

The program will ask for the unit's current GPIB address. For a primary address, enter the address as a one or two digit primary address. For new units, use the factory default address of 4. For a secondary address, use 4 digits in the *ppss* format (2 digit primary, 2 digit secondary address, i.e. 0401).

Note - If you do not know your unit's GPIB address, turn the unit off and back on. The unit will display its primary GPIB address at the end of its self test routine by blinking its LEDs. A double blink indicates the unit is in the dual secondary address mode. The LED bit weights are:

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

8. Configuration Choices

The configuration program steps through each configurable function, displays the current setting and gives you a multiple choice or asks for a numeric entry. Refer to the SCPI commands in Tables 3-2 and 3-3 for an explanation of the choices and entry values. All choices are made by entering the corresponding choice or value and pressing the Return key. To skip a parameter, just press the Return key.

Note that the configuration program is only able to set the digital interface for string type commands. Settings for direct port I/O commands or for the Status Reporting Structure will have to be entered from the GPIBkybd program or as part of the user's program.

After the last selection, the program will ask if the unit is configured correctly. Enter **N** to go back and change the configuration; **Y** to continue.

10. Saving the New Settings

2

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

The program may ask if you want to lock the interface configuration parameters from being changed by the end user. Enter **Y** to lock; **N** to continue.

11. Configuring other units

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

2.7 CONFIGURING FROM THE GPIBKYBD AND OTHER PROGRAMS

The 4863 can be configured from any GPIB bus controller by using the following procedure and a interactive control program like ICS's GPIBkybd program. The example commands can also be imported into any program language. If you are using the GPIBkybd program, just send the unit the command strings. The following examples use NI 488.2, ICS and HP command examples. 2363s can be configured by entering the commands directly in any terminal emulation program. Refer to section 3.9 for more information about programming 2363s.

1. Connect the 4863 to the bus controller or computer as directed in 2.6 step 2. Turn both units on.

2. Determine the 4863's GPIB address or the 2363's network address by one of the following methods:

   a) Use the factory default value of 4 for new units.

   b) If you do not know your unit's address, turn the unit off and back on. The unit will display its primary address at the end of its self test routine by blinking its LEDs. A double blink indicates the 4863 is in the dual secondary address mode. The LED bit weights are:

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

3. Make a copy of Table 1-3 or 1-4 and note the parameter settings. Mark up the parameters that you want to change. You only have to change the ones that are different from the shipped (default) configuration. You can always return to the factory default settings with the CAL:DEFAULT command.

4. Verify communication with the unit by querying its IDN message. This example uses National Instrument commands.

```
Call Send(Bd, Addr,"*IDN?", EOTMode)
Instring$ = String$(Lin, 32)          'fills the string with spaces
Call Receive(Bd, Addr,  Instring$, Term)
```

2

5. Use the CAL:LOCK and CAL:DEFAULT commands to unlock the unit and to restore its default setting. (Skip the default command if you are only changing a specific parameter)

```
Call Send(Bd, Addr,"CAL:LOCK 0", EOTMode)
                                        'Unlocks all parameters
Call Send(Bd, Addr,"CAL:DEF", EOTMode)
                                        'Sets all default values
```

6. Use the equivalent OUTPUT and ENTER type statements in your computer's program to send the configuration commands to the 4863. Each new configuration statement should be followed with a query to verify that the unit accepted the new setting or visually monitor the ERR LED.

   e.g., to enable bit 5 in the unit's Event Status Enable Register:

```
Call Send(Bd, Addr,"*ESE 32", EOTMode)
Call Send(Bd, Addr,"*ESE?", EOTMode)
Instring$ = String$(Lin, 32)    'fills the string with spaces
Call Receive(Bd, Addr,  Instring$, Term)
Print A$                        'Displays the result
```

   If you entered an invalid command and the red ERR LED illuminates, send the unit the "*CLS" command to clear the error and turn off the ERR LED. Correct the command and repeat the above step.

7. Use caution when changing the unit's GPIB or network address. The change takes place immediately when the command is executed or before the 2363 returns its prompt. The unit requires a 70 millisecond delay after an address change command before it can accept another command or query.

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

```
CALL ieOutput(4, "SYST:COMM:GPIB:ADDR 20")
msDelay 70                      'wait 70 ms
CALL ieOutput(20, "SYST:COMM:GPIB:ADDR?"
Rdg$ = String$(10, " ")         'fills Rdg$ with 10 spaces
CALL ieEnter(20, Rdg$)
PRINT Rdg$                      'displays the query response
```

i.e., to change the unit's GPIB address to dual secondary

```
OUTPUT 720; "SYST:COMM:GPIB:ADDR:MODE DUAL"
OUTPUT 72000;"SYST:COMM:GPIB:ADDR:MODE?"
ENTER 72000, A$
PRINT A$                          'displays the query response
```

With the NI 488.2 command set, secondary addresses are offset by 96 and multiplied by 256 to be in the upper byte of the address variable. The above example in NI 488.2 commands becomes:

```
Addr=20; BD=0
CmdStr$="SYST:COMM:GPIB:ADDR:MODE DUAL"
Call Send(Bd,Addr,CmdStr$, EOTMode)
Addr = (256*96)+20                'primary 20, secondary  0
CmdStr$="SYST:COMM:GPIB:ADDR:MODE DUAL"
Call Send(Bd,Addr,CmdStr$, EOTMode)
Instring$ = String$(Lin, 32)      'fills the string with spaces
Call Receive(Bd, Addr,  Instring$, Term)
Print Instring$
```

8. Use the IEEE 488.2 ***SAV 0** command to save the new values in the unit's nonvolatile memory as the power-on default values. Use the unit's current GPIB address.

```
OUTPUT 720; "*SAV 0"              'primary address 20
                  or
OUTPUT 72000; "*SAV 0"            'primary 20, secondary 0
```

Caution - Do not put the *SAV 0 command in a continuously running program loop.

Figure 2-2 lists a general purpose program in HP BASIC that can be used to configure the unit. Figure 2-3 lists a short program that only changes the unit's GPIB address.

```
100     REM 4863 COMMAND ENTRY / READOUT PROGRAM
110     PRINT “”
120     PRINT “”
130     PRINT “4863 COMMAND ENTRY PROGRAM”
140     DIM A$[100]
150     DIM B$[200]
160     Devadr=4
170     Adr=704
180     ON TIMEOUT 7,5 GOTO 820
190     ON INTR 7 GOSUB 620
200     Mask=2
210     ENABLE INTR 7;Mask
220     GOSUB 470
230     CLEAR Adr
240     WAIT .1
250     GOSUB 770
260     WAIT .1
270     CLEAR Adr
280     GOSUB 520
290     PRINT “Device address = “;Devadr,
300     LINPUT “ENTER COMMAND STRING, or HELP for Directions”,A$
310     IF A$=”HELP” THEN GOSUB 520
320     IF A$=”SPOL!” THEN GOSUB 650
330     IF A$=”DATA!” THEN GOSUB 720
340     IF A$=”CMD!” THEN GOSUB 770
350     IF (A$<>”ENTER” AND A$<>”XXX”) THEN
360             PRINT “OUTPUTTING->”,A$
370             OUTPUT Adr;A$ END
380     END IF
390     IF POS(A$,”ENTER”)>0 THEN GOSUB 420
400     IF POS(A$,”?”)>0 THEN GOSUB 420
410     GOTO 290
420     REM GET ENTER DATA
430     PRINT “         WAITING FOR DATA...”
440     ENTER Adr;B$
450     PRINT “         DATA STRING ->”,B$
460     RETURN
470     REM Change address
480     INPUT “Enter Device Address (0-30) : “,Devadr
490     Adr=Devadr+700
500     A$=”XXX”
510     RETURN
```

Figure 2-2 HP BASIC Command Entry/Readout Program

```
520     REM Help Menu
530     PRINT
540     PRINT “HELP         for this menu”
550     PRINT “SPOL!        to Serial Poll Device address”
560     PRINT “DATA!        select DATA address”
570     PRINT “CMD!         select CMD address”
580     PRINT “ENTER        to read data from Device address”
590     PRINT
600     A$=”XXX”
610     RETURN
620     REM Serial Poll
630     PRINT
640     PRINT “********** SRQ INTERRUPT OCCURRED! **********”
650     PRINT “Serial Polling now...”
660     Stat=SPOLL(Adr)
670     PRINT “SRQ SPOLL RESULTS: Adr “;Devadr;”Status “;Stat
680     PRINT “********************************************************”
690     A$=”XXX”
700     ENABLE INTR 7
710     RETURN
720     REM DATA mode
730     PRINT “DATA Address selected!”
740     Adr=Devadr+701
750     A$=”XXX”
760     RETURN
770     REM CMD Mode
780     PRINT “CMD Address selected!”
785     Adr=Devadr+700
790     SEND 7;UNL LISTEN Devadr
800     A$=”XXX”
810     RETURN
820     REM GPIB Timeout routine
830     PRINT “GPIB Timeout Occurred!”
840     RETURN
850     END
```

Figure 2-2 HP BASIC Command Entry/Readout Program (Cont'd.)

```
100   REM BASIC GPIB ADDRESS SETTING PROGRAM
110   DIM A$[100], B$[20]
120   CLEAR 7
130   WAIT .1
140   INPUT “Enter Current Device Address (0-30):”, Devadr
150   Adr=Devadr+700
160   SEND 7; UNL LISTEN Devadr
170   WAIT .1
180   A$=”SYST:COMM:GPIB:ADDR?”
190   OUTPUT Adr; A$ END
200   ENTER Adr; B$
210   PRINT “ Device address = “;B$
220   INPUT “ Enter new device address (0-30): “,Newadr
230   A$=” SYST:COMM:GPIB:ADDR “
240   OUTPUT Adr;A$;Newadr END
250   WAIT .1
260   Adr=Newadr+700
270   A$=”SYST:COMM:GPIB:ADDR?”
280   OUTPUT Adr;A$ END
290   ENTER Adr; B$
300   PRINT “ New Device address = “;B$
310   OUTPUT Adr;”*SAV 0” END
330   END
```

Figure 2-3   HP BASIC Address Configuration Program

2.7 CABLE CONNECTIONS

2.7.1 4863 Connections

The 4863 has two connectors on the rear panel. J1 is the IEEE 488/GPIB bus connector and J2 is the Digital I/O Connector.

Any standard IEEE 488 bus cable may be used to connect the 4863 to the bus controller as shown in Figure 2-4. ICS can provide IEEE Bus Cables with standard double headed connectors or straight-in connectors. Always use good multi-shielded cables to minimize noise pickup. Limit the total GPIB bus cable length to 20 meters. Refer to the appendix for a GPIB signal-pin list if one is needed or contact ICS for more information about bus cables.

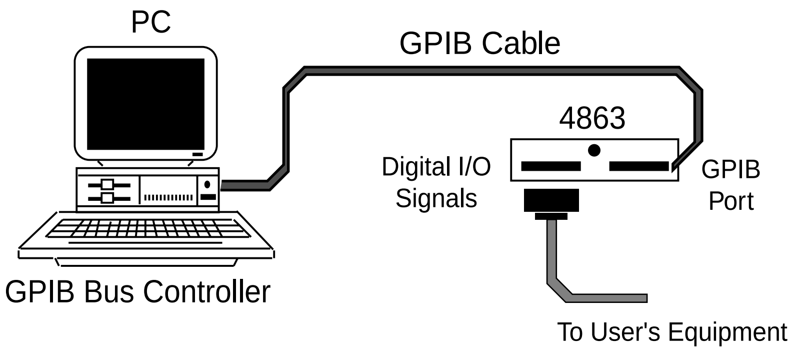


Figure 2-4 4863 Cable Connections

A user generated cable is required to connect the Digital I/O signals to the external devices or signals. Use the mating connector supplied with the unit and follow the directions in Section 2.9 to fabricate a digital I/O cable or use the 4863 Open End Cable, P/N 114508.

2

2.7.2 2363 Connections

The 2363 has two connectors on the rear panel. J1 is the serial interface connector and J2 is the Digital I/O Connector.

2.7.2.1 RS-232 Connections

The 2363's serial interface connector (J1) is a DB-25S 25 pin female connector with both RS-232 and RS-485 signals. The RS-232 signals are configured as a DCE device. Pin arrangements are such that the 2363 can be connected to any DTE device such as a PC COM port or to a terminal with most pin-to-pin RS-232 cables. The 2363 does not use any data handshaking since its command and response strings are fairly short.

> **CAUTION - Pins 14 and 16 contain RS-485 signals and should not be connected to any RS-232 level signals. Check standard RS-232 cables to be sure these pins are open before plugging the cable into a 2363.**

Table 2-1 lists the pin assignments for the 2363's serial interface.

TABLE 2-1 2363 SERIAL PIN ASSIGNMENTS (J1)

Signal	Pin	2363 Function
Shield	1	Chassis or shield ground
TxD	2	RS-232 Received Data input
RxD	3	RS-232 Transmit Data output
RTS	4	Internal jumper for DTE devices
CTS	5	
DSR	6	+5 Vdc
GND	7	Logic Ground
DCD	8	+5 Vdc
DTR	20	no connection
Tx/Rx+	16	RS-485 Tx/Rx Data I/O
Tx/Rx-	14	RS-485 Tx/Rx Data I/O

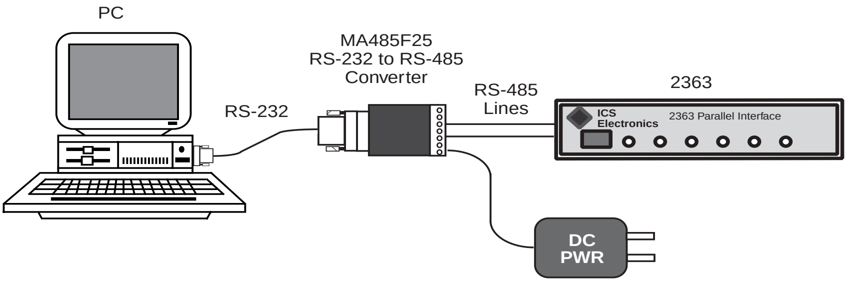
Note that the RS-232 specification defines TxD as data being transmitted to the DCE device; RxD as the data being transmitted to the DTE device. Therefore, TxD is pin 2 on the 2363's serial connector.

2.7.2.2 RS-485 Connections

The 2363 has two RS-485 signals on Pins 14 and 16 for a half-duplex RS-485 connection. Pin 14 has the Tx/Rx+ signal; pin 16 has the Tx/Rx- signal. The 2363 contains an internal network termination circuit with 1 Kohm pullup and pulldown resistors and a 220 ohm termination resistor. The termination network biases J1 pin 16 at 2.44 Vdc and pin 14 at 2.2 Vdc.

Half-duplex RS-485 networks normally have only one termination network in or close to the last device on the network. Long networks may have a termination network at both ends of the network cable. To remove an unwanted 2363 Termination Network, open the 2363, unsolder and remove resistors R5, R6 and R7 from the PC board. These resistors are located near the front of the PC board between the PWR and RDY LEDs.

The 2363's RS-485 port can be tested by using a RS-232 to RS-485 converter to connect the 2363 to a PC's COM port The recommended converter is a MA485F25 from ICSDataCom.com. Figure 2-5 shows the connections from the PC to the 2363. Use a plain pin-to-pin cable from the PC to the MA485F25 Converter. Use ICS's SERkybd program (see para 3.9.7) of an equivalent program to test the network protocols.



Notes: Wiring List
MA485F25 Terminals 4 & 5 to 2363 J1 pin 14
MA485F25 Terminals 3 & 6 to 2363 J1 pin 16
MA485F25 Terminal 7 to blk/wht wire (+ Vdc)
MA485F25 Terminal 8 to blk wire (GND)

Figure 2-5 2363 RS-485 Test Connections

2.7.3 OEM Board Cable Connections

OEM Board versions of the 4863/2363 are equipped with vertical headers for the GPIB interface, serial interface and LED drive signals. The number of headers varies with the board configuration.

The Digital I/O connector on OEM Boards is the same as on the standard 4863/2363. Use the mating connector supplied with the unit and follow the directions in Sections 2.9 to connect the OEM Board to the host equipment.

2

2.7.3.1 Power Connections

DC power is connected to the 4867/2367 OEM board at the J6 terminal block. Connect the positive lead to the 'RED' terminal shown in Figure 2-6. Use unregulated 9 to 32 volts DC power.

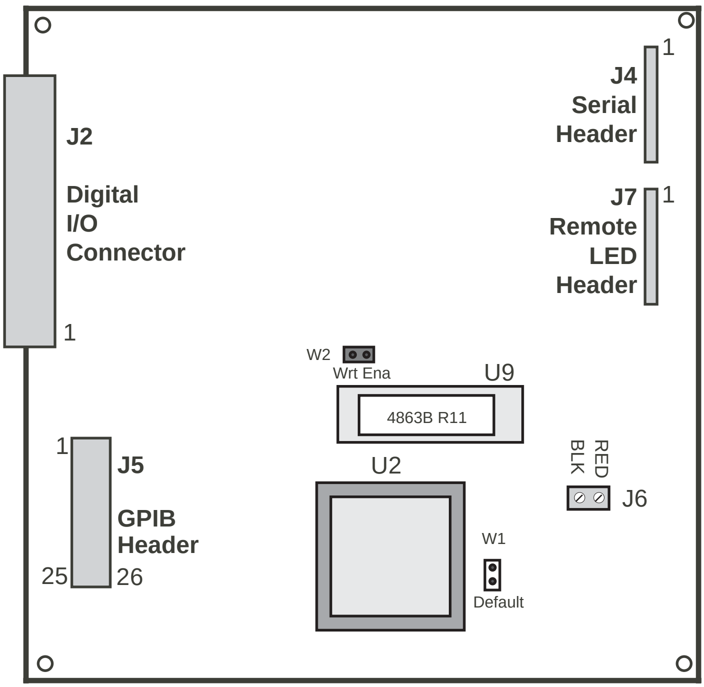


Figure 2-6 4863 OEM Board Layout

2.7.3.2 Serial Header Connections

The Serial Header J4 is a 10-pin male header with 0.025 square pins on 0.1 centers. Pin 8 is the key position and is omitted from the header. The Serial Header location on the 4863 lower board is shown in Figure 2-6. Signal pin assignments are listed in Table 2-2.

TABLE 2-2 SERIAL HEADER PIN ASSIGNMENTS

Pin	Signal	Function
1	Chassis	Shield tie point
2	RxD	RS-232 Input
3	TxD	RS-232 Output
4	Tx/Rx+	RS-485 + Signal
5	Tx/Rx-	RS-485 - Signal
6	Vcc	+ 5 Vdc
7	Ground	RS-232 Signal Ground
8	-	key
9	Ground	Logic Ground
10	Reset#	Reset Input. A low input resets the 4863.

Table 2-3 shows a suggested way to wire the serial connections to a 25-pin or a 9-pin connector. The RS-232 Standard specifies female connectors (DB-25S) for DCE devices and male connectors (DB-25P) for DTE devices. Because the 4863 only has Tx and Rx signals, the user can easily make a Data Terminal (DTE) or Data Communication (DCE) style interface for the end product by adding the appropriate jumpers and voltages as shown in Table 2-3. The recommendation is to add the jumpers in case the mating device uses hardware handshaking signals.

RS-485 interfaces do not have a standard interface connector. Our recommendation is to use a 9 pin or smaller connector. The 4863's RS-485 interface operates in the half-duplex mode and has a resistor network that biases the RS-485 signals to the mark (logic '1') condition when they are not being driven. The resistor network has 1 kohm pullup and pulldown resistors on each side of a 220 ohm load resistor. This resistor network should be sufficient for cables distances up to 100 feet in length. For longer length cables, the user should provide a similar network at the other end of the serial link.

TABLE 2-3 SERIAL HEADER-CONNECTOR WIRING EXAMPLES

Signal	DCE DB-25S Pin#	DTE DB-25P Pin#	J4 Header Pin#	J4 Header Signal
Shield	1	1	1	Chassis
RxD	2	3	2	Receive Data
TxD	3	2	3	Transmit Data
RTS	4	4		
CTS	5	5		
DSR	-	6		
GND	7	7	7	GND
DCD	-	8		
DTR	20	20		
DSR	6		6	+ 5 Vdc thru 3 KΩ

Signal	DE-9S Pin#	DE-9P Pin#	J4 Header Pin#	J4 Header Signal
RxD	2	3	2	Receive Data
TxD	3	2	3	Transmit Data
RTS	7	7		
CTS	8	8		
DSR	-	6		
GND	5	5	7	GND
DCD	-	1		
DTR	4	4		
DSR	6		6	+ 5 Vdc thru 3 KΩ

2.7.3.3 LED Header

The LED Header J7 contains the signals for driving a duplicate set of 4863 LEDs. The LED Header is a 7-pin male header with 0.025 square pins on 0.1 centers. The header is not keyed. The LED header location on the 4863 OEM board is shown in Figure 2-6. LED Signal pin assignments are listed in Table 2-4.

TABLE 2-4 LED HEADER PIN ASSIGNMENTS

Pin	Signal	Function
1	Vcc	+ 5 Vdc
2	GND	Signal Ground
3	ERR#	ERROR LED
4	SRQ#	SRQ LED
5	LSTN#	LSTN LED
6	TALK#	TALK LED
7	RDY#	READY LED

2

The LED drive signals are low true and designed to sink 15 mA of extra current to operate the remote LEDs. The remote LEDs may be powered by the +5 volts supplied on pin 1. Each LED should have a series resistor with a minimum value of 270 ohms to limit the LED operating current.

2.7.3.4 GPIB/Address Switch Header J5

GPIB header J5 contains the GPIB signals and address switch sense inputs. J5 signal-pin assignments are listed in Table 2-5 and the connector layout is shown in Figure 2-7. The table also lists the wire colors for the rainbow ribbon cable used to connect J5 to ICS's GPIB Connector/Address Switch Assembly although any GPIB connector and address switch may be used with the OEM Board.

ICS's GPIB Connector/Address Switch Assemblies are small, PC assemblies that mount a GPIB connector and an 8-bit rocker switch to the rear panel of a chassis. The assemblies are available in two layout styles. Refer to Appendix A2 for GPIB Connector/Address Switch Assembly dimensions and installation instructions.

The external address switch inputs are low true signals. Pullup resistors are provided on the OEM Board so an open is a logic '0'. To set an address externally, the address signals (ADSW1-ADSW5) are jumpered or switched to ground in a binary fashion. (The GPIB address is the sum of the binary weights of the grounded bits) To set an external address of 0, connect the SI SW input to ground and leave the other switch inputs open (OFF).

Use the **SYST:COMM:GPIB:ADDR:EXT 1** command to enable the external address switch and save the new configuration.

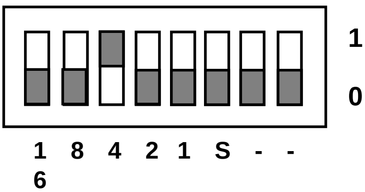
TABLE 2-5 GPIB/Address Header Signals

Signal	Pin Number	Wire Color	Bit Weights
GROUND	1	BRN 1	
ADSW5	2	RED 1	16 (MSB)
T SW	3	ORG 1	not used
L SW	4	YEL 1	not used
ADSW4	5	GRN 1	8
SI SW	6	BLU 1	0
ADSW1	7	VIO 1	1
ADSW3	8	GRY 1	4
ADSW2	9	WHT 1	2
NRFD	10	BLK 1	
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	

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

(a) J2 GPIB/Sw Connector Layout

(See Figure 1-6 for connector orientation)



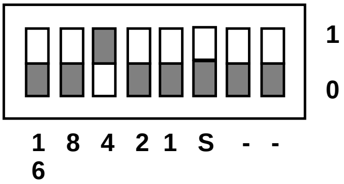
Notes: Switch shown set to address 4

(b) GPIB Address Switch Rocker Layout

Figure 2-6 4863 J2 Connector and Address Switch Layouts

e.g. **SYST:COMM:GPIB:ADDR:EXT 1**
***SAV 0**

The 4863 reads the external address switch at power turn-on time. Always cycle power off and back on after changing the address switch setting. On ICS's GPIB Connector/Address Switch Assemblies, the rocker switches are laid out as shown below in Figure 2-8.



Notes: Switch show set to address 4
Rocker 'S' used for Address 0

Figure 2-8 Address Switch Layout

2.7.3.5 Board Jumpers

4863/2363s have two jumpers on the PC board as shown in Figure 2-6. The jumper functions are listed in Table 2-6

TABLE 2-6 4863/2363 JUMPER SETTINGS

Jumper	Functions	Factory Setting
W1	Default - Returns the unit to its factory default settings when in place at power turn-on time. Leave out for normal operation	Omitted
W2	Write Enable - Must be in place to write to or save in flash memory.	Installed

2

This page intentionally left blank

2.8 DIGITAL I/O CONNECTIONS

The 4863/2363's Digital Signals are on connector J2. Connector J2 is a 62 pin metal DC shell connector with female lockstuds that mates to a DC-62S plug. Table 2-7 lists the J2 signal-pin assignments. Table 2-8 is a worksheet for recording the configuration settings when designing the interface.

2.8.1 Digital I/O Overview

The digital I/O lines are controlled by 8-bit bidirectional latches. Data direction is referred to as output when the digital lines output data received from the GPIB or Serial bus. Data direction is input when data is read in from the digital lines and is talked onto the GPIB bus or outputted serially.

The simplest data transfer is with the port type commands that address a specific I/O byte or bit commands that toggle or read a specific bit. Data polarity can be set on a bit-by-bit basis with the port commands. Bytes controlled by the port and bit commands **do not need to be configured** with the configuration commands.

String commands or transparent data strings can be used to transfer data in multiple byte wide words. Bytes used by the string commands **must be configured** into input or output strings by the configuration commands. Depending upon the selected data format, each eight-bit byte can be considered as two 4-bit nibbles where four bits of information are encoded into a single character on the GPIB or serial bus. Two characters make up a byte. Other formats let the user express a byte as decimal numbers or as a single binary byte. Data format and polarity are the same for all configured output or input bytes.

The 15 data lines in the first two bytes are sampled at a 1 kHz rate and their values saved in the Questionable Register when the bytes are used as input bytes. Changes in any of the 15 data lines can be used to generate an SRQ or service request message. The user can select bit transition directions and enable individual bits to detect signal changes and generate a Service Request when an enabled bit is set. See Section 3.4 for more information about the card's Status Reporting Structure. Bytes naturally default to inputs and **do not need to be configured** with the configuration commands to be monitored by the interface.

2.8.2 Data Transfer Methods

The 4863 transfers data between the GPIB bus and the 48 digital I/O lines with byte I/O commands, by bit manipulation, with data string commands, or by transparent data strings. The 2363 does not have transparent data transfer capability.

Commands transfer data as a parameter in the command. Byte oriented data commands like PORTn directly address and transfer data to or from an unconfigured port on the digital interface. Each port command transfers eight bits of data. Bit manipulation commands let the user set and reset bits in an unconfigured output byte or read a bit in an unconfigured input byte. Data polarity can be on a bit basis.

String oriented data commands like PORT? and VALUE pass multiple bytes of data per command to one or more ports that have been configured as inputs or outputs. Transparent data transfer passes the same data strings used with string commands without the command syntax by addressing the interface at a second address. Data polarity and format is the same for all of the bytes assigned to a string.

Characters in a data string can be HEX characters, user assigned ASCII characters, ASCII numbers or binary characters. When HEX or user assigned ASCII characters are used, each character represents a four bit BCD or HEX value on four of the digital I/O lines. It takes two HEX characters to program or to read a eight bit byte. HEX code is the sixteen characters 0 thru 9 and A thru F for a value of 0 to 15. The TALK format provides a TABLE entry command where the user can assign his own sixteen characters. The LISTEN format provides a 4833 selection with an alternate input character set of 0-9 and :;<=> ? are used for compatibility with ICS's earlier Model 4833 Parallel Interface. Listen string transfers from the GPIB bus or serial ports must end with a linefeed character with or without EOI asserted. The 4863 appends a linefeed character with EOI asserted to the input data when talking on the GPIB bus or a carriage return for serial transmissions.

e.g. The HEX characters 3E sets eight output bits to 0011 1110

ASCII numbers transfers an eight bit pattern on the digital I/O connector as a decimal value that is the sum of the binary bit weights for all of the bits set to logic 1. Value is 0 to 255. Values for multiple bytes are separated by commas.

e.g. The ASCII value 65 sets the eight output bits to 0100 0001

BINARY characters transfer an 8-bit value to the digital I/O connector as a single character. Value is 0 to 255. BINARY characters should be used with care since they are not always printable and can represent a command terminator such as the linefeed character.

2.8.3 Digital I/O Lines

The forty-eight bidirectional data lines (CH1-48) have 33 Kohm pullup resistors and are shown numbered 1 thru 6 in ascending order In Table 2-7. For BCD or HEX data transfer, each byte is split into two 4 bit nibbles. Nibbles are numbered Most Significant to Least Significant Nibble in left to right order like writing a number on paper. Data transfer order is in the MSN to LSN direction which is the same order as the ascending byte count. The user can leave the bytes unassigned for direct byte I/O or assign multiple bytes into Talk (input) or Listen (output) strings. Bytes assigned to strings must be assigned in numerical order and can start with any byte number. User sets data polarity, conversion format and data transfer handshaking. The recommendation is to use bytes 1 and 2 for inputs to preserve the monitoring capability.

2.8.4 Input Handshaking

When input handshaking is enabled, the External Data Ready (EDR) signal sets the EDR flip-flop and tells the 4863/2363 that the external data is valid and can be read. The INH output signal is generated by the EDR flip-flop. The data must stay steady from the leading edge of the EDR signal to the trailing edge of the Inhibit (INH) signal. The user can select positive or negative going EDR edge, high or low true Inhibit signal and several modes of inputting the data. Figure 2-9 shows the relative signal timing. Absolute timing depends upon the input mode selected.

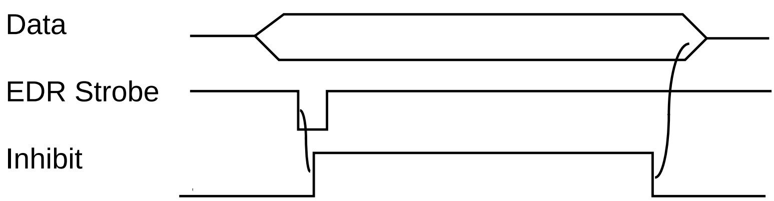


Figure 2-9 EDR Signal Timing

TABLE 2-7 4863 SIGNAL-PIN ASSIGNMENTS

Signal	Signal Weighting			Pin#	User Signals	
	Binary	BCD/HEX			Pin#	Signal
CH 8	Byte 1 Bit 7	MSN	Bit 8	39		
CH 7	Byte 1 Bit 6	MSN	Bit 4	18		
CH 6	Byte 1 Bit 5	MSN	Bit 2	60		
CH 5	Byte 1 Bit 4	MSN	Bit 1	40		
CH 4	Byte 1 Bit 3	MSN-1	Bit 8	19		
CH 3	Byte 1 Bit 2	MSN-1	Bit 4	61		
CH 2	Byte 1 Bit 1	MSN-1	Bit 2	41		
CH 1	Byte 1 Bit 0	MSN-1	Bit 1	20		
CH 16	Byte 2 Bit 7	MSN-2	Bit 8	15		
CH 15	Byte 2 Bit 6	MSN-2	Bit 4	57		
CH 14	Byte 2 Bit 5	MSN-2	Bit 2	37		
CH 13	Byte 2 Bit 4	MSN-2	Bit 1	16		
CH 12	Byte 2 Bit 3	MSN-3	Bit 8	58		
CH 11	Byte 2 Bit 2	MSN-3	Bit 4	38		
CH 10	Byte 2 Bit 1	MSN-3	Bit 2	17		
CH 9	Byte 2 Bit 0	MSN-3	Bit 1	59		
CH 24	Byte 3 Bit 7	MSN-4	Bit 8	54		
CH 23	Byte 3 Bit 6	MSN-4	Bit 4	34		
CH 22	Byte 3 Bit 5	MSN-4	Bit 2	13		
CH 21	Byte 3 Bit 4	MSN-4	Bit 1	55		
CH 20	Byte 3 Bit 3	MSN-5	Bit 8	35		
CH 19	Byte 3 Bit 2	MSN-5	Bit 4	14		
CH 18	Byte 3 Bit 1	MSN-5	Bit 2	56		
CH 17	Byte 3 Bit 0	MSN-5	Bit 1	36		
CH 32	Byte 4 Bit 7	MSN-6	Bit 8	31		
CH 31	Byte 4 Bit 6	MSN-6	Bit 4	10		
CH 30	Byte 4 Bit 5	MSN-6	Bit 2	52		
CH 29	Byte 4 Bit 4	MSN-6	Bit 1	32		
CH 28	Byte 4 Bit 3	MSN-7	Bit 8	11		
CH 27	Byte 4 Bit 2	MSN-7	Bit 4	53		
CH 26	Byte 4 Bit 1	MSN-7	Bit 2	33		
CH 25	Byte 4 Bit 0	MSN-7	Bit 1	12		

[shaded box] indicates signals also used as the Questionable Register inputs

TABLE 2-7 4863 SIGNAL-PIN ASSIGNMENTS CONT'D

Signal	Signal Weighting: Binary	Signal Weighting: BCD/HEX	Pin#	User Signals: Pin#	User Signals: Signal
CH 40	Byte 5 Bit 7	MSN-8 Bit 8	7		
CH 39	Byte 5 Bit 6	MSN-8 Bit 4	49		
CH 38	Byte 5 Bit 5	MSN-8 Bit 2	29		
CH 37	Byte 5 Bit 4	MSN-8 Bit 1	8		
CH 36	Byte5 Bit 3	MSN-9 Bit 8	50		
CH 35	Byte 5 Bit 2	MSN-9 Bit 4	30		
CH 34	Byte 5 Bit 1	MSN-9 Bit 2	9		
CH 33	Byte 5 Bit 0	MSN-9 Bit 1	51		
CH 48	Byte 6 Bit 7	MSN-10 Bit 8	46		
CH 47	Byte 6 Bit 6	MSN-10 Bit 4	26		
CH 46	Byte 6 Bit 5	MSN-10 Bit 2	5		
CH 45	Byte 6 Bit 4	MSN-10 Bit 1	47		
CH 44	Byte6 Bit 3	LSN Bit 8	27		
CH 43	Byte 6 Bit 2	LSN Bit 4	6		
CH 42	Byte 6 Bit 1	LSN Bit 2	48		
CH 41	Byte 6 Bit 0	LSN Bit 1	28		

Signal	Function	Pin
EDR	External Data Ready Input	25
INH	Inhibit Signal Output	45
Stat A	Status A Input	4
Stat B	Status B Input	44
Trigger	Trigger Pulse Output	24
Reset	Reset Pulse Output	3
Remote	Remote State Output	43
Clear	Clear Pulse Output	23
Strobe	Data Output Strobe	2
Vcc	+5 Vdc Output #	1
Gnd	Signal Ground	22
+12V	+12Vdc Unreg Input *	62
Ret	Unreg Input Return *	42
-	Shield Ground	21

Notes: # Limit load to 100 mA, * for OEM use only

2

2.8.5 Status Inputs

Two input signals (Status A and Status B) can be used to monitor the external device's condition. The Status signals are inputted thru the Operational Register in the 4863's 488.2 Status Reporting Structure so that changes to the status inputs can be used to generate a Service Request and interrupt the Bus Controller. User programs Status Signal polarities, monitor conditions and enables SRQ generation.

2.8.6 Output Data Strobe and Handshaking

The 4863/2363 pulses the Output Data Strobe after placing string data on the data lines. The Output Data Strobe can also be pulsed by the :STRobe command. If Listen handshaking is enabled, the Strobe line is only pulsed when the Status A input is in its logic 1 state. Holding the Status A input in the logic 0 state inhibits data transfer. User programs the Strobe polarity and enables/disables Listen Handshaking.

2.8.7 Trigger Output

The 4863/2363 pulses the Trigger line when the 4863/2363 receives a valid trigger command. User programs Trigger polarity. Use the Trigger output to initiate external action.

2.8.8 Reset Output

The 4863/2363 pulses the Reset line at power turn-on or when the 4863/2363 receives the 488.2 *RST command. Use the Reset output to initialize an external device.

2.8.9 Clear Output

The 4863/2363 pulses the Clear line at power turn-on or when the 4863 receives the 488.1 Device Clear command. Use the Clear output to initialize an external device's data registers.

2.8.9 Remote State Output

The 4863/2363 asserts the Remote State line when the 4863 enters the GPIB remote state. Use the Remote State false output to enable front panel or local controls that override or conflict with GPIB controlled functions. The Status B input can be used as a Request to go to Local input into the 4863 by connecting it to a button on the host chassis front panel. Momentarily pulse the Status B line and the 4863 will respond by deasserting the Remote State line if the 4863 is not in the GPIB Remote Lockout State. The 4863 will reassert the Remote State line when it is next addressed as a Listener and put in the Remote state. User programs Status B and Remote State line polarities.

TABLE 2-8 I/O CONFIGURATION CHART

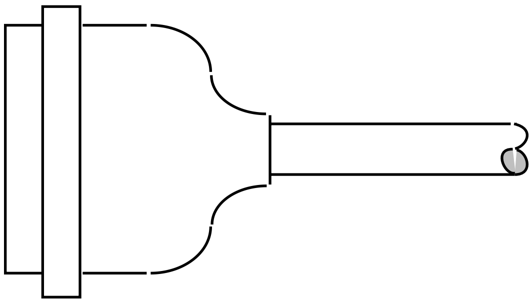
Parameter	Function	New Setting
Talk String		
:INPut	Sets number of Talk bytes	
:POLarity	Sets Input data polarity	
:HANDshake	Enables Input Handshaking	
:EDR	Sets input polarity of edge	
:INH	Sets inhibit output polarity	
:TALK	Selects Input String Format	
:TRANSlation	Sets input conversion table if needed	
:EOM	Sets End-of-message string	
Listen String		
:OUTput	Sets number of Listen bytes	
:POLarity	Sets Output polarity	
:HANDshake	Enables Output Handshaking	
:STRobe	Sets Output Strobe polarity	
:LISTen	Sets Output Format	
:ASTATus	Sets Input Polarity	
:BSTATus	Sets Input Polarity	
:CLEar	Sets Output Polarity	
:REMote	Sets Output Polarity	
:RESet	Sets Output Polarity	
:TRIGger	Sets Output Polarity	
Byte Transfer		
:POLarity	Sets Byte 1 polarity	
:POLarity	Sets Byte 2 polarity	
:POLarity	Sets Byte 3 polarity	
:POLarity	Sets Byte 4 polarity	
:POLarity	Sets Byte 5 polarity	
:POLarity	Sets Byte 6 polarity	

2.9 DDIGITAL I/O CABLE DESIGN

The following instructions will guide the user through the design of an interface cable between the Digital I/O connector and an external device. The end product is a three sheet 'A' drawing that can be used to fabricate the interface cable and a table with the 4863 configuration settings.

2

1 For custom cables, make a copy of Tables 2-7 and 2-8 as your worksheets. For the 4863 Open End Cable shown in Figure 2-10, copy Tables 2-9 and 2-8. Use these sheets when directed to record signals, pin numbers etc.



The 4863 Open End Cable is a five foot long cable terminated into a 62-pin connector. Table 2-9 lists the cable wire colors.

Figure 2-10 4863 Open End Cable Connector

2. Make a rough determination of the number of output and input signals from the external device. The number should not exceed 48.

3. If any of the signal lines are to be monitored, they should be assigned to the first two bytes (CH1-CH15). CH16 is not monitored. The monitored lines may also be read by a byte or string command.

4. If input data is being transferred by a string command, assign the external devices to the 4863 data lines starting with first free byte. For BCD/HEX coding, start with the lowest numbered data lines. Assign the signals to the proper bit weights so that the numbers come out correctly. Typical assignments for a small panel meter are:

2

TABLE 2-9 4863 CABLE SIGNAL ASSIGNMENTS

Signal	Signal Weighting: Binary	Signal Weighting: BCD/HEX		Wire Color of Pair Color	User Pin#
CH 8	Byte 1 Bit 7	MSN	Bit 8	Yel of Yel/Wht	
CH 7	Byte 1 Bit 6	MSN	Bit 4	Red of Grn/Red	
CH 6	Byte 1 Bit 5	MSN	Bit 2	Vio of Red/Vio	
CH 5	Byte 1 Bit 4	MSN	Bit 1	Wht of Yel/Wht	
CH 4	Byte 1 Bit 3	MSN-1	Bit 8	Blu of Blu/Red	
CH 3	Byte 1 Bit 2	MSN-1	Bit 4	Wht of Wht/Vio	
CH 2	Byte 1 Bit 1	MSN-1	Bit 2	Brn of Brn/Wht	
CH 1	Byte 1 Bit 0	MSN-1	Bit 1	Red of Blu/Red	
CH 16	Byte 2 Bit 7	MSN-2	Bit 8	Wht of Wht/Red	
CH 15	Byte 2 Bit 6	MSN-2	Bit 4	Org of Org/Vio	
CH 14	Byte 2 Bit 5	MSN-2	Bit 2	Blu of Blu/Wht	
CH 13	Byte 2 Bit 4	MSN-2	Bit 1	Red of Wht/Red	
CH 12	Byte 2 Bit 3	MSN-3	Bit 8	Vio of Org/Vio	
CH 11	Byte 2 Bit 2	MSN-3	Bit 4	Wht of Blu/Wht	
CH 10	Byte 2 Bit 1	MSN-3	Bit 2	Grn of Grn/Red	
CH 9	Byte 2 Bit 0	MSN-3	Bit 1	Red of Red/Vio	
CH 24	Byte 3 Bit 7	MSN-4	Bit 8	Brn of Org/Brn	
CH 23	Byte 3 Bit 6	MSN-4	Bit 4	Grn of Brn/Grn	
CH 22	Byte 3 Bit 5	MSN-4	Bit 2	Org of Org/Blk	
CH 21	Byte 3 Bit 4	MSN-4	Bit 1	Yel of Yel/Org	
CH 20	Byte 3 Bit 3	MSN-5	Bit 8	Org of Org/Grn	
CH 19	Byte 3 Bit 2	MSN-5	Bit 4	Blk of Org/Blk	
CH 18	Byte 3 Bit 1	MSN-5	Bit 2	Org of Yel/Org	
CH 17	Byte 3 Bit 0	MSN-5	Bit 1	Grn of Org/Grn	
CH 32	Byte 4 Bit 7	MSN-6	Bit 8	Yel of Yel/Grn	
CH 31	Byte 4 Bit 6	MSN-6	Bit 4	Blk of Yel/Blk	
CH 30	Byte 4 Bit 5	MSN-6	Bit 2	Brn of Yel/Brn	
CH 29	Byte 4 Bit 4	MSN-6	Bit 1	Grn of Yel/Grn	
CH 28	Byte 4 Bit 3	MSN-7	Bit 8	Brn of Brn/Blk	
CH 27	Byte 4 Bit 2	MSN-7	Bit 4	Org of Org/Brn	
CH 26	Byte 4 Bit 1	MSN-7	Bit 2	Brn of Brn/Grn	
CH 25	Byte 4 Bit 0	MSN-7	Bit 1	Blk of Brn/Blk	

[shaded box] indicates signals also used as the Questionable Register inputs

TABLE 2-9 4863 CABLE SIGNAL ASSIGNMENTS Cont'd

Signal	Signal Weighting: Binary	Signal Weighting: BCD/HEX	Wire Color of Color Pair	User Pin#
CH 40	Byte 5 Bit 7	MSN-8 Bit 8	Blu of Blu/Blk	
CH 39	Byte 5 Bit 6	MSN-8 Bit 4	Org of Org/Blu	
CH 38	Byte 5 Bit 5	MSN-8 Bit 2	Blu of Blu/Grn	
CH 37	Byte 5 Bit 4	MSN-8 Bit 1	Blk of Blu/Blk	
CH 36	Byte5 Bit 3	MSN-9 Bit 8	Blu of Org/Blu	
CH 35	Byte 5 Bit 2	MSN-9 Bit 4	Grn of Blu/Grn	
CH 34	Byte 5 Bit 1	MSN-9 Bit 2	Yel of Yel/Blk	
CH 33	Byte 5 Bit 0	MSN-9 Bit 1	Yel of Yel/Brn	
CH 48	Byte 6 Bit 7	MSN-10 Bit 8	Blu of Yel/Blu	
CH 47	Byte 6 Bit 6	MSN-10 Bit 4	Red of Org/Red	
CH 46	Byte 6 Bit 5	MSN-10 Bit 2	Grn of Grn/Blk	
CH 45	Byte 6 Bit 4	MSN-10 Bit 1	Brn of Brn/Blu	
CH 44	Byte6 Bit 3	LSN Bit 8	Wht of Wht/Grn	
CH 43	Byte 6 Bit 2	LSN Bit 4	Blk of Grn/Blk	
CH 42	Byte 6 Bit 1	LSN Bit 2	Blu of Brn/Blu	
CH 41	Byte 6 Bit 0	LSN Bit 1	Grn of Wht/Grn	
Signal	**Function**			
EDR	External Data Ready Input		Org of Org/Red	
INH	Inhibit Signal Output		Yel of Yel/Blu	
Stat A	Status A Input		Blk of Wht/Blk	
Stat B	Status B Input		Wht of Org/Wht	
Trigger	Trigger Pulse Output		Red of Brn/Red	
Reset	Reset Pulse Output		Wht of Wht/Blk	
Remote	Remote State Output		Org of Org/Wht	
Clear	Clear Pulse Output		Brn of Brn/Red	
Strobe	Data Output Strobe		Blk of Red/Blk	
Vcc	+5 Vdc Output #		Red of Red/Blk	
Gnd	Signal Ground		Red of Yel/Red	
+12V	+12Vdc Unreg Input *		Vio of Wht/Vio	
Ret	Unreg Input Return *		Wht of Brn/Wht	
-	Shield Ground		Yel of Yel/Red	

Notes: # Limit load to 100 mA, * for OEM use only

I/O Signals:	CH4-1,	CH8-5,	CH12-9,	CH16-13
Inputs:	Sign/overrange,	MSD,	MSD-1,	LSD
Example:	+	1	9	9

Several bits such as Sign and Overrange can be combined into one four line group to conserve data lines. Use jumpers to +5 Vdc and to ground to convert a single line signal such as 'Polarity' into a four bit value. See the sample cable in Section 2.9 for jumper examples. Tie all unused data input lines to the logic '0' potential (If the inputs are low true, leave the unused signals open. The 4863/2363's internal pullup resistors will pull the signals to +5 V). Record the signal names and pin numbers in your copy of Table 2-7 or 2-9.

2

If handshaking is enabled, record the external signal (and pin number) that will be connected to the EDR input in Table 2-7. If the 4863's Inhibit output is being used, record the signal name and pin number in Table 2-7 or 2-9. In Table 2-8 record the number of input bytes used, the data polarity, Talk handshaking enabled status, EDR polarity and Inhibit polarity.

5. If output data is being transferred by a string command, the output data lines should be assigned next. Start with the lowest available byte in Table 2-7 or 2-9. If the string data is HEX or BCD characters, the lowest available nibble will be the MSN in the output data. i.e. if 2 bytes were used in step 2, then MSN-4 becomes the most significant digit in the output string. i.e. for an output string of "1234", the "0001" code will appear on lines CH20-CH17.

Assign the signals to the proper bit weights so that the numbers come out correctly for the external device. For binary data use the binary bit weights. Record the device signal names and pin numbers in Table 2-7 or 2-9. Leave the unused data lines open. Record the bytes used and signal polarity in Table 2-8.

If the Output Data Strobe will be used, record the signal and pin number in Table 2-7 or 2-9. Record strobe polarity in Table 2-8.

If Listen Handshaking is to be enabled, connect a signal to Status A input in Table 2-7 or 2-9. In Table 2-8 record the Listen Handshake enabled and the polarity of the Status Input when data can be transferred.

2

6. If any data will be transferred with Byte commands, their signals should be connected now. Assign the signals to the remaining data lines. All signals on a byte should go in the same direction. Record the device signal names and pin numbers in Table 2-7 or 2-9. Record each byte's polarity in Table 2-8.

7. If any control signals are being used, they should be connected at this time. Record the signal names and pin numbers in Table 2-7 or 2-9. Record the active signal polarities in Table 2-8.

8. Complete your cable drawing by copying Figure 2-10 onto an 'A' size cover page and adding the device's mating connector. Attach your worksheets to the cover sheet.

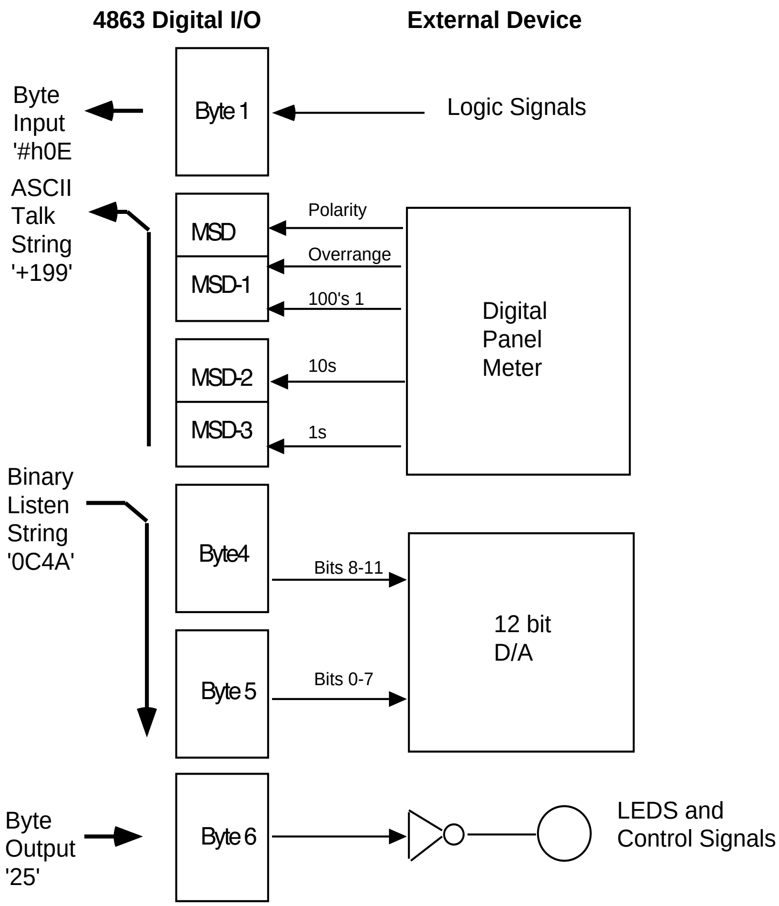


Figure 2-11 Example 4863 Digital I/O Connections

2.10 EXAMPLE DIGITAL CONNECTIONS

Figure 2-11 shows how a 4863/2363 may be used to control an external device and input data. In this example, the external device has a small Digital Panel Meter (DPM), a D/A converter, some control inputs, a four bit counter, two status outputs and three LEDs. The data arrows in Figure 2-11 show some ways that data can be transferred to/from the GPIB bus or Serial port.

Table 2-10 lists the signal connections for this example. The first byte is used to input the Counter and Status signals so that they could later be monitored by the questionable register if the need arose. The byte is directly queried with the PORTn? query. The response format used is the one set by the FORM:TALK command.

Bytes 2 and 3 are used to input data from a 2 1/2 Digital Panel Meter (DPM) data as an ASCII string. The bytes are divided into two 4 bit nibbles since each character on the GPIB bus represents only four bits on the digital interface. The DPM's 'busy' signal is used to pulse the 4863's EDR input. The string could be read either transparently or with the PORT? (@2,3) query. Typical in-range responses are -199 to + 199. The Overrange signal is wired to bit 8 of the 100s digit so Overrange responses are -999 or +999. The polarity digit is created by wiring the Plus signal to bit 1. Bits 8, 4 and 2 are fixed with jumpers to be 101. The resulting code is 1011 for plus, 1010 for negative. In the 4863's Talk Conversion Table, character 10 is set to an ASCII '-' and character 11 is set to an ASCII '+'.

Bytes 4 and 5 are used to output 12 bits to a D/A converter. The 4863 Strobe line is used to load the data into the D/A's internal latch. Data transfer is done with a string of four HEX characters, two characters per byte. Data transfer can be as a transparent data string or can be done with the SOURCe:DATA: VALUE command. In Figure 2-11, Byte 4 is sent as an '0C' and Byte 5 is sent as an '4A'. The resultant output pattern is '0C4A'

Byte 6 is directly addressed with the PORTn command to output eight bits of data. Data format can be a decimal value (0 to 255) or a hex value (#h00 to #hFF). Polarity could be set low true so a logic 1 is a low output to turn-on the LEDs.

Tables 2-10 and 2-11 show the completed Signal Assignments and Configuration Tables for this example.

2

TABLE 2-10 EXAMPLE CONNECTION TABLE

Signal	Signal Weighting: Binary	Signal Weighting: BCD/HEX		Pin	User Signals: Pin	User Signals: Signal
CH 8	Byte 1 Bit 7	MSN	Bit 8	39		gnd = 0
CH 7	Byte 1 Bit 6	MSN	Bit 4	18		gnd = 0
CH 6	Byte 1 Bit 5	MSN	Bit 2	60	25	Status bit 2
CH 5	Byte 1 Bit 4	MSN	Bit 1	40	26	Status bit 1
CH 4	Byte 1 Bit 3	MSN-1	Bit 8	19	27	Counter bit 4
CH 3	Byte 1 Bit 2	MSN-1	Bit 4	61	28	Counter bit 3
CH 2	Byte 1 Bit 1	MSN-1	Bit 2	41	29	Counter bit 2
CH 1	Byte 1 Bit 0	MSN-1	Bit 1	20	30	Counter bit 1
CH 16	Byte 2 Bit 7	MSN-2	Bit 8	15		+5 = 1
CH 15	Byte 2 Bit 6	MSN-2	Bit 4	57		gnd = 0
CH 14	Byte 2 Bit 5	MSN-2	Bit 2	37		+5 = 1
CH 13	Byte 2 Bit 4	MSN-2	Bit 1	16	1	Plus
CH 12	Byte 2 Bit 3	MSN-3	Bit 8	58	2	Overrange
CH 11	Byte 2 Bit 2	MSN-3	Bit 4	38		gnd = 0
CH 10	Byte 2 Bit 1	MSN-3	Bit 2	17		gnd = 0
CH 9	Byte 2 Bit 0	MSN-3	Bit 1	59	3	100s 1
CH 24	Byte 3 Bit 7	MSN-4	Bit 8	54	4	10s 8
CH 23	Byte 3 Bit 6	MSN-4	Bit 4	34	5	10s 4
CH 22	Byte 3 Bit 5	MSN-4	Bit 2	13	6	10s 2
CH 21	Byte 3 Bit 4	MSN-4	Bit 1	55	7	10s 1
CH 20	Byte 3 Bit 3	MSN-5	Bit 8	35	8	1s 8
CH 19	Byte 3 Bit 2	MSN-5	Bit 4	14	9	1s 4
CH 18	Byte 3 Bit 1	MSN-5	Bit 2	56	10	1s 2
CH 17	Byte 3 Bit 0	MSN-5	Bit 1	36	11	1s 1
CH 32	Byte 4 Bit 7	MSN-6	Bit 8	31		-
CH 31	Byte 4 Bit 6	MSN-6	Bit 4	10		-
CH 30	Byte 4 Bit 5	MSN-6	Bit 2	52		-
CH 29	Byte 4 Bit 4	MSN-6	Bit 1	32		-
CH 28	Byte 4 Bit 3	MSN-7	Bit 8	11	15	D/A bit 11
CH 27	Byte 4 Bit 2	MSN-7	Bit 4	53	16	D/A bit 10
CH 26	Byte 4 Bit 1	MSN-7	Bit 2	33	17	D/A bit 9
CH 25	Byte 4 Bit 0	MSN-7	Bit 1	12	18	D/A bit 8

[shaded box] indicates signals also used as the Questionable Register inputs (shaded rows: CH 1–8, CH 9–15)

TABLE 2-10 EXAMPLE CONNECTION TABLE CONT'D

Signal	Signal Weighting: Binary	Signal Weighting: BCD/HEX	Pin	User Signals: Pin	User Signals: Signal
CH 40	Byte 5 Bit 7	MSN-8 Bit 8	7	19	D/A bit 7
CH 39	Byte 5 Bit 6	MSN-8 Bit 4	49	20	D/A bit 6
CH 38	Byte 5 Bit 5	MSN-8 Bit 2	29	21	D/A bit 5
CH 37	Byte 5 Bit 4	MSN-8 Bit 1	8	22	D/A bit 4
CH 36	Byte5 Bit 3	MSN-9 Bit 8	50	23	D/A bit 3
CH 35	Byte 5 Bit 2	MSN-9 Bit 4	30	24	D/A bit 2
CH 34	Byte 5 Bit 1	MSN-9 Bit 2	9	25	D/A bit 1
CH 33	Byte 5 Bit 0	MSN-9 Bit 1	51	26	D/A bit 0
CH 48	Byte 6 Bit 7	MSN-10 Bit 8	46		-
CH 47	Byte 6 Bit 6	MSN-10 Bit 4	26		-
CH 46	Byte 6 Bit 5	MSN-10 Bit 2	5	40	Control bit 2
CH 45	Byte 6 Bit 4	MSN-10 Bit 1	47	41	Control bit 1
CH 44	Byte6 Bit 3	LSN Bit 8	27		-
CH 43	Byte 6 Bit 2	LSN Bit 4	6	42	LED #3
CH 42	Byte 6 Bit 1	LSN Bit 2	48	43	LED #2
CH 41	Byte 6 Bit 0	LSN Bit 1	28	44	LED #1
Signal	**Function**		**Pin**		
EDR	External Data Ready Input		25	12	DPM Busy
INH	Inhibit Signal Output		45		-
Stat A	Status A Input		4	50	Ext +5 Vdc
Stat B	Status B Input		44		gnd
Trigger	Trigger Pulse Output		24		-
Reset	Reset Pulse Output		3		-
Remote	Remote State Output		43		-
Clear	Clear Pulse Output		23	49	Ext Reset In
Strobe	Data Output Strobe		2	27	D/A Load
Vcc	+5 Vdc Output #		1		for +5V jumpers
Gnd	Signal Ground		22		for gnd jumpers
+12V	+12Vdc Unreg Input *		62		-
Ret	Unreg Input Return *		42		-

Notes: # Limit load to 100 mA, * for OEM use only

2

TABLE 2-11 EXAMPLE CONNECTION CONFIGURATION SETTINGS

Parameter	Function	New Setting
Talk String		
:INPut	Sets Talk bytes	2,3
:POLarity	Sets Input data polarity	1
:HANDshake	Enables Input Handshaking	ON
:EDR	Sets input polarity of edge	0
:INH	Sets inhibit output polarity	0
:TALK	Selects Input String Format	TABLE
:TRANSlation	Sets input conversion table if needed	0-9-+ .,E and space
:EOM	Sets End-of-message string	10 (LF)
Listen String		
:OUTput	Sets Listen bytes	4,5
:POLarity	Sets Output polarity	0
:HANDshake	Enables Output Handshaking	OFF
:STRobe	Sets Output Strobe polarity	0
:LISTen	Sets Output Format	HEX
:ASTATus	Sets Input Polarity	1
:BSTATus	Sets Input Polarity	0
:CLEar	Sets Output Polarity	0
:REMote	Sets Output Polarity	0
:RESet	Sets Output Polarity	0
:TRIGger	Sets Output Polarity	-
Byte Transfer		
:POLarity	Sets Byte 1 polarity	#h FF
:POLarity	Sets Byte 2 polarity	
:POLarity	Sets Byte 3 polarity	
:POLarity	Sets Byte 4 polarity	
:POLarity	Sets Byte 5 polarity	
:POLarity	Sets Byte 6 polarity	#h 00

3

Programming Instructions

3

3.1 INTRODUCTION

This section describes the operation of the 4863 and 2363, its IEEE 488.2 Status Structure, 488.2 Commands, SCPI commands, Programming Guidelines and Serial Interface. Functional descriptions of the 4863 apply to the 2363 unless otherwise stated.

3.2 GENERAL OPERATION

3.2.1 All Units

The 4863 and 2363's digital interface is user configurable as inputs or outputs in eight bit bytes. Interface configuration and data transfer is done by commands received from the GPIB bus or Serial interface. The digital interface configuration, data formats and transfer protocol can be saved in Flash and are automatically recalled at power turn-on or when the unit is reset.

Data transfer between the computer and the 4863 and 2363's digital interface can be by a combination of individual bits, by bytes or as strings of data values to multiple bytes.

• Bit commands set or reset specific bits in a byte or query a bit’s status.
• Byte commands set all 8 bits in a specific byte or read data from a byte.
• String transfer commands send strings of data characters to one or more output bytes to make a multi-byte output word or read a string of data from one or more input bytes. The user designates these bytes as inputs or outputs when he configures the board. By using a second GPIB address, the 4863 can transparently transfer data as strings of formatted data characters or as binary bytes.

3

The 4863 is an IEEE-488.2 compatible device. Its Status Reporting Structure lets the user monitor up to fifteen digital I/O lines, the EDR Flip-flop and two status inputs for changes and generate an SRQ when the enabled changes occur. SRQs can also be generated when external data is ready or when the output data handshake is not received.

The 4863 uses SCPI commands which are a subset of the Standardized Commands for Programmable Instruments Standard. The advantage of the SCPI commands is that they are common to a wide variety of instruments and are self documenting in the program listing. Most of the 4863's SCPI commands or queries have a corresponding short form command for easy programming. Where possible, the short form commands are the same as those used in ICS's earlier Model 4833 GPIB-to-Parallel Interface. Because most of the command settings may be individually queried, the 4863 does not provide a combined configuration message output.

The 4863 has three GPIB address modes: a single primary address, two consecutive primary addresses or a primary address with two secondary addresses. The data transfer capabilities of the different address modes are shown below in Figure 3-1. The upper row shows that the single primary or lower dual or secondary GPIB address is used for all 488.1, 488.2, SCPI and short form commands, queries and responses and for transferring data with commands. The second or upper GPIB address is only used only for transparent data transfer and for 488.1 commands. The address mode is user configured by a SCPI command. At power turn-on, the 4863 blinks the address once if a primary address mode is enabled or twice if the secondary address mode is enabled.

Address Mode			Data Transfer and Command Capabilities
Single Primary	Dual Primary	Dual Secondary	
Only GPIB Address Address blinks in Selftest	Lower GPIB Address Lower GPIB address blinks in Selftest	GPIB Address plus Secondary address 0 Primary address blinks twice in Selftest	All 488.1, 488.2, Setup commands and data as command parameters
	Higher GPIB Address (Lower + 1)	GPIB Address plus Secondary address 1	Transparent data and 488.1 commands

Figure 3-1 4803 Address Modes and Data Transfer Methods

The bit manipulation commands let the user directly set or reset a bit without having to keep a image of the byte in the user's program. When a bit is set or reset, the rest of the bits in the byte are not affected but all of the bits in the byte become output bits. Bits can be sensed or read individually. When a bit is read, all of the bits in the byte become input bits. Ports used with the bit manipulation commands are **not** configured as input or output ports.

Direct data transfer to a specific port is performed with the SENSE and SOURCe PORTn commands. These ports are **not** configured as input or output ports. Data transfer can be a decimal or hex value and data polarity can be set on a bit-by-bit basis.

To output data as strings, the user configures one or more byes as output bytes, sets the data polarity and listen format. If handshaking is disabled, the unit will accept character strings at its GPIB or serial interface, convert the characters into digital bits, set the output latches and generate the data strobe pulse. Listen data can be hexadecimal characters (0-9, A-F), the ASCII number subset (0-9 and :<=>?), decimal values that represent the sum of the binary bits in a byte, or binary bytes (4863 only). If listen handshaking is enabled, data transfer will only occur if the Status A input signal is in its logic true state. Data overrun is be reported as an Execution Error.

To input data as a string, the user configures one or more bytes as input bytes, sets the data polarity and talk format. If handshaking is disabled, the unit will simply input the digital data when addressed to talk or when queried, convert it into the selected format, append the EOM character and output the data on the GPIB bus or serial interface. If talk handshaking is enabled, data transfer will occur if the EDR (External Data Ready) signal is received and the unit has been addressed to talk. **The unit must be UNTALKed between transparent data transfers**. If queried by a command and the EDR signal is not received, the 4863 will report an Execution Error. The external device must hold the digital inputs stable until the 4863 drops its Inhibit line. The talk data may be a series of hexadecimal characters, routed through an arbitrary Hex-to-ASCII conversion table, or as a series decimal values that represent the binary sum of the logic '1' bits in each input byte.

The 4863 uses its IEEE-488.2 Status Reporting Structure to generate Service Requests (SRQs) if the EDR bit becomes set or the Status inputs change state. The 4863/2363 can also monitor fifteen of the digital I/O lines for changes with the Questionable Transition Registers. The lines are only monitored if they are assigned as inputs. Detected input signal changes set corresponding bits in the Questionable Event register. The register summary bit will appear in the Status Byte Register and if enabled, set the Service Request bit, generating an SRQ. The user can serial poll or query the Status Byte to determine the source of the SRQ and read the input data by a command or through the Questionable register.

The 4863 provides multiple control and status and control lines for interfacing with external digital devices. Output control signals include Trigger, Reset, Remote, Clear and Data Strobe lines. Inputs include two general purpose status lines that are inputted via the Operation Condition register in the 488.2 Status Reporting Structure. The user can set the Status Reporting Structure to monitor these lines and generate a SRQ when they change state or periodically poll them by querying the Operation Condition register.

3

At power turn on, the 4863 restores the last configuration and output values stored in its Flash memory by the ***SAV 0** command as the current configuration. These values are used until a command is sent to the 4863 with a replacement value which then becomes part of the current configuration. When power is turned off, the current configuration values are lost. The ***SAV 0** command must be sent to the 4863 to save the new current configuration in the 4863's Flash memory. The user should set the 4863's digital outputs to their correct power-on values before doing a ***SAV 0** command. A lock function hides the configuration settings and prevents changes to the configuration parameters when the lock function is enabled. The GPIB address, address mode and Status Register settings are not affected by the lock function.

3.2.2 2363 Differences

The 2363 is similar to the 4863 except that the 4863's GPIB interface is replaced by a RS-232 and a RS-485 (RS-422) serial interfaces. The 2363's operating functions and digital I/O commands are the same as those described for the 4863 when operated in the single address mode.

The 2363 only transfers data as command parameters or in response to a query and returns a 'prompt' when ready for the next command. GPIB SRQs are replaced by a serial Service Request Message (SRM). The 2363's serial command syntax is the same as the 4863's GPIB command syntax. The 2363 has additional SCPI commands for setting its serial interface. Changes to the serial interface are only made when the new values are recalled from memory or when the 2363 is sent a UPdate command.

The 2363 automatically responds to the serial port that received the command. 2363s use the same address parameter as does the 4863 but the 2363 address range is limited to 0 to 15.

The 2363 is capable of operating in a multi-device, RS-485 network. When Network Address protocol is enabled, the 2363 only responds to commands prefixed with the correct address string. All other commands and serial messages are ignored. When Network Packet protocol is selected, the 2363 only responds to valid packets with its address. All valid packets are responded to with an ACK packet, a NAK packet or with a reply packet. The 2363 only enables its RS-485 transmitter when responding to commands or queries.

3.2.3 OEM Board Differences

OEM Board versions of the 4863 are available with various interface combinations. Board configurations with GPIB capability, have an GPIB signal header with extra inputs for reading the GPIB address from an external switch. When the external switch is enabled, the external inputs are examined at power turn-on time and replace the internal GPIB address. The 4863 displays its GPIB bus address by blinking the front panel LEDs at the end of the power-on self test.

Some OEM Board configurations have an RS-232 or RS-485 interface for communicating serially with the host computer. The use of the serial interface is the same as described above for the Model 2363. When OEM Boards have both serial and GPIB interfaces, the serial interface is enabled at power turn-on time and disabled when the unit is addressed on the GPIB bus and put into the remote state.

OEM users should document the board as part of their system and provide instructions and command examples on how to operate the system. Only include those 4863/2363 commands that the end user should change. The setup and configuration commands can be locked out after the unit is configured.

3.3 SETTING THE 4863's GPIB ADDRESS

The 4863 can be set to any unused GPIB primary addresses between 0 and 30. GPIB addresses 0 and 21 should be avoided as they are customarily reserved for use by the GPIB bus controllers. If the dual primary address mode is selected, the lower (command) address range is limited from 0 to 29. If the secondary address mode is selected, the primary address range is limited from 1 to 30. The 4863 can not recognize secondary addresses when set to primary address 0. The selected address setting and mode is stored in the 4863's Flash. Follow the directions in Sections 2.5 or 2.6 to change the 4863's GPIB address. Note that the GPIB address change takes place immediately so any subsequent commands must use the new address and/ or address mode. The GPIB address and mode must be saved in Flash to be a permanent change.

3

The 4863 is set to single address mode and primary address 4 and when shipped from the factory.

3.4 488.2 STATUS REPORTING STRUCTURE

The 4863 includes the expanded IEEE-488.2 status reporting structure shown in Figure 3-2. 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 and Operation 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 4863. The added Questionable and Operation registers are controlled and queried with SCPI commands.

As shown in Figure 3-2, 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'. Each register is summarized in a bit in the Status Byte Register. When those bits are enabled, the Status Byte Register sets bit 6 and generates a Service Request by pulling the SRQ line low. SRQs are used to signal the bus controller that an event has occurred and/or that the unit needs service. There are three major sources of SRQs, each of which is summarized in the Status Byte Register. Two of the sources are event registers with their own enabling bits and the third is the Output Queue. The Event registers and the Output Queue are cleared when read or by the *CLS command. The RQS bit in the Status Byte Register is reset when the unit is serial polled or when the cause is gone. The *STB? query does not reset the MSS bit.

3.4.1 Event Registers

An event register **captures 0 to 1 transitions** in its associated condition register or in the standard event conditions. An event bit becomes TRUE (1) when the associated condition bit makes logical 0 to 1 transition. 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 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.

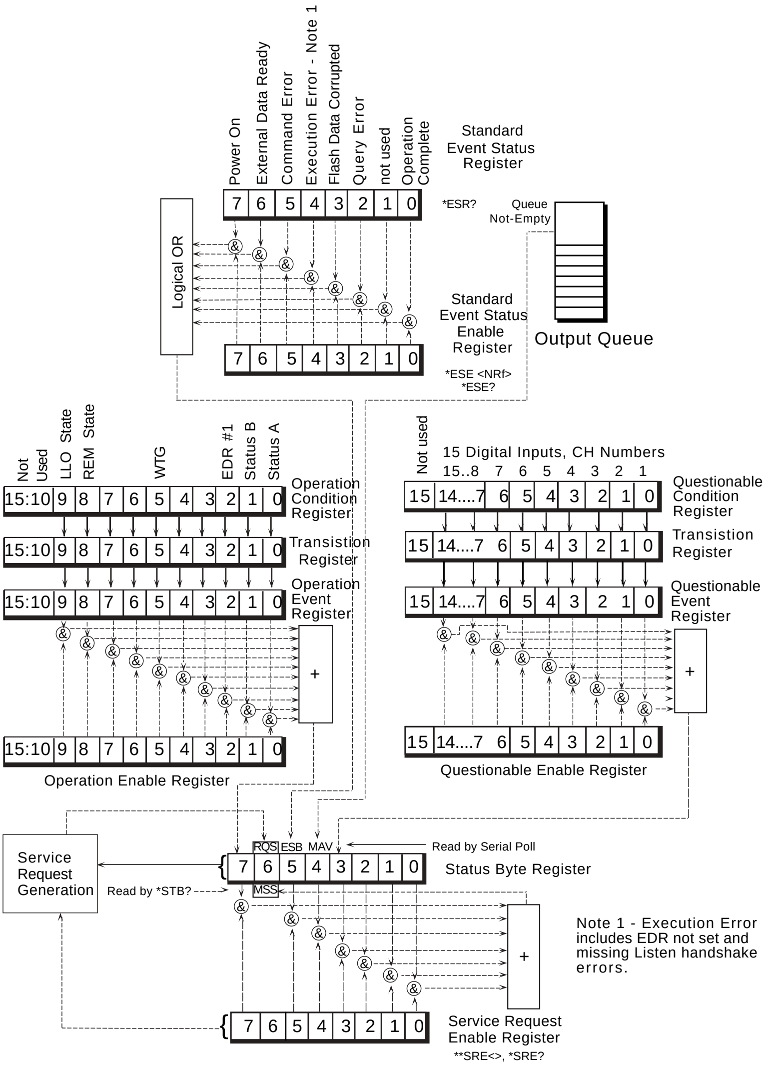


Figure 3-2 4863 Status Reporting Structure

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. The Power-on event occurs at power turn-on and can be used to signal a power off-on occurrence. The External Data Ready flip-flop is included in the Event Status Register as Bit 6. The 488.2 Operation Complete event is not implemented.

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

***ESE 60**	'enables error bits 2 through 5 for errors
***ESE 124**	'enables error bits 2 through 5 and the EDR bit
***ESE?**	'quires the enabling register setting

3

3.4.3 Questionable Registers and Digital Inputs

The Questionable Registers let the user read the first fifteen digital input lines and detect any changes in the digital inputs. Bit alignments are shown in Figure 3-2. The Questionable Transition Register filters the inputs and passes only the enabled state changes to the Questionable Event Register. The Questionable Event Register bits becomes true (1) when the positive transition bit is enabled and the associated condition register bit makes a 0 to 1 transition or when the negative transition bit is enabled and the associated condition register bit makes a 1 to 0 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.

3.4.3.1 Monitoring Digital Inputs for State Changes

The 4863 can be set to monitor the digital inputs and generate a SRQ or SRM when they change state. The following example sets the Questionable Event register to monitor digital inputs CH1 and CH2 by capturing a positive transition on bit 0 and a negative transition on bit 1:

STAT:QUES:PTR 1 'enables bit 0 to set on a positive transition of CH1
STAT:QUES:NTR 2 'enables bit 1 to set on a negative transition of CH2

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 0 and 1:

STAT:QUES:ENAB 3 'enables Event bits 0 and 1

Note that the Questionable Event Register has to be cleared after an SRQ is generated by reading the register or with the *CLS command. If the register is not cleared, the event bits will remain set.

3

3.4.3.2 Reading the Digital Inputs

The Questionable Condition Register reflects the **real time** condition of the 4863's first 15 digital inputs. A logical 1 means that the corresponding digital input is high or an open contact. A logical 0 is a low input of a contact closure to ground. To read the Questionable Condition Register use the following SCPI query:

STAT:QUES:COND? 'reads the status inputs

The response is a decimal number that is the sum of the digital inputs whose levels are equal to a logical 1. Reading the Questionable Condition Register does not change its contents.

3.4.4 Operation Registers

The 488.2 Operation Registers let 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 4863 and 2363, the Operation Condition Register reports the Status A and Status B inputs, the EDR Flip-flop and the WTG (Waiting for Trigger) status. The WTG bit is true when the unit has been armed and is waiting for a trigger. The 4863 also has the LLO and REM status bits from the GPIB interface. The following commands demonstrate some possibilities of the Operation Registers:

STAT:OPER:PTR 4	'enables bit 2 to set on a positive transition of EDR F/F
STAT:OPER:NTR 2	'enables bit 1 to set on a negative transition of Status B (Front panel local request)
STAT:OPER:ENAB 6	'enables Event bits 1 and 2
STAT:OPER:COND?	'quires the Operation Condition Register

3.4.5 Output Queue

The Output Queue is used by the 4863 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 and are normally read before sending another command or query to the unit. 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 Output Queue by addressing the 4863 to talk at its GPIB address clears the summary bit. If the Output Queue is not read before sending another query, its contents will be lost and an error reported.

3.4.6 Status Byte Register

The 4863 generates a service request, SRQ, (2363 generates a SRM) whenever any of the enabled bits in the Status Byte Register become true and the unit is not addressed as a talker. The Status Byte Register may be read by a Serial Poll or with the ***STB?** query. The Status Byte Register is enabled by setting the corresponding bits in the Service Request Enable Register with the ***SRE** command. e.g.

***SRE 40**	'Sets the SRE Register to 0010 1000 which enables just the Event Status and Questionable summary bits to generate SRQs.

3

3.4.7 Saving the Enable Register Values

The ESE and SRE Register values can only be saved and recalled at power turn-on by disabling the PSC flag. **The *SAV command does not save the ESE and SRE register values.** Use the *PSC 0 command to disable the PSC flag and **save** the ESE and SRE register values. The following example saves the ESE and SRE values and enables a SRQ at power turn-on.

***PSC 0; ESE 192; SRE 32** 'enables Power-on and EDR bits

Note that the ESE and SRE 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 ESE and SRE registers to be cleared at the next power turn-on and prevent a power-on SRQ.

3

***PSC 1**

3.4.8 488.2 Differences from 488.1 Devices

The IEEE 488.1 Device Clear command **does not** reset the 4863's digital outputs as would be expected of a 488.1 device. To reset the 4863's digital outputs, use the *RST (Reset) or *RCL 0 (Recall default) command to output the default values.

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 4863 responds to these commands and to some optional common commands defined in the IEEE-488.2 Standard. Table 3-1 lists how the 4863 responds to these commands and describes their effect on the 4863 and its status reporting structure.

Any command that ends with a question mark '?' is a query. All queries should be followed by reading their response to avoid data loss and a command execution error.

3

TABLE 3-1 IEEE-488.2 COMMON COMMANDS

COMMAND	NAME	DESCRIPTION
*** CLS**	Clear Status	Clears all event registers summarized in the status byte, except for "Message Available," which is cleared only if *CLS is the first message in the command line.
***ESE <value>**	Event Status Enable	Sets "Event Status Enable Register" to <value>, an integer between 0 and 255. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated. EXAMPLE: decimal 16 converts to binary 00010000 which sets bit 4 to a logical 1.
***ESE?**	Event Status Enable Query	4863 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	4863 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.

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

COMMAND	NAME	DESCRIPTION
***IDN?**	Identification Query	4863 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, 4863, S/N 510123, Rev 00.00 Ver 98.05.18** Note: the IEEE-488.2 specification states that the word 'model' may not appear in the IDN message.
***OPC**	Operation Complete Command	Causes the 4863 to generate the operation complete message in the Standard Event Status Register when all pending selected 4863 operations have been finished.
***OPC?**	Operation Complete Query	Places an ASCII character 1 into the 4863's Output Queue when all pending selected 4863 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 save the SRE and ESE values in the E^2PROM
***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 4863 from a copy stored in its E2PROM by *SAV command. *RCL 0 restores the power on setting and pulses the Reset output signal. <value> = 0
***RST**	Reset	4863 restores its power-up state except that the state of IEEE-488 interface is unchanged, including: instrument address,

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

COMMAND	NAME	DESCRIPTION
***RST Cont'd**		Status Byte and ESR Register. Disables the trigger function and pulses the Reset output signal.
***SAV <value>**	Save	Saves current 4863 configuration in the E^2PROM. *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>, an integer between 0 and 255. The value of bit six is ignored because it is not used by the Service Request Enable Register. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated.
***SRE?**	Service Request Enable Query	4863 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	4863 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 when triggers have been enabled.
***TST?**	Self-Test Query	Queries the results of the last self test. A zero response indicates no failures. Other responses are listed in Table 5-1.
***WAI**	Wait-to-continue	Prevents the 4863 from executing any further commands or queries until the No-Operation-Pending flag is TRUE.

3.6 SCPI CONFORMANCE INFORMATION

The 4863/2363 accepts SCPI commands and command extensions to configure its digital interface, to set the data formats and to transfer data. The SCPI commands conform to SCPI Standard 1995.0 and provide an industry standard, self-documenting form of code that makes it easy for the programmer to maintain the application program.

Table 3-2 shows the 4863/2363's SCPI command tree. The 4863 and 2363 uses portions of the SCPI SYSTEM, STATUS, CONFIGURE, FORMAT, SENSE, SOURCE, INITIATE and CALIBRATE subsystems. The commands 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 in the 4863/2363 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.

e.g., **CONFigure:INPUT 1**

is the same as

CONF:INP 1

or

conf:inp 1

Table 3-3 lists the SCPI keywords and describes their functions in detail. Keywords other than those listed in the table or locked keywords will have no effect on the 4863's operation and a command error will be reported. Read the notes at the end of the table and refer to Appendix A-1 for additional information about SCPI commands.

Note: Most SCPI commands can be quired by adding a '?' to the end of the command. All queries should be followed by reading their response to avoid data loss and a command execution error.

TABLE 3-2 SCPI COMMAND TREE

<table>
<tr><th>Keyword</th><th>Parameter Form</th><th>Notes & Single Letter Commands</th></tr>
<tr><td>SYSTem</td><td></td><td>System Address</td></tr>
<tr><td>:COMMunicate</td><td></td><td></td></tr>
<tr><td>:GPIB</td><td></td><td>4863 and OEM Bd</td></tr>
<tr><td>:ADDRess</td><td>0 - 30 [4]</td><td></td></tr>
<tr><td>:EXTernal</td><td>0|1 or OFF|ON [0]</td><td></td></tr>
<tr><td>:MODE</td><td>SINGle | DUAL | SECondary</td><td></td></tr>
<tr><td>:OFFset</td><td>1 - 10 [1]</td><td></td></tr>
<tr><td>:SERial</td><td></td><td>2363 and OEM Bds</td></tr>
<tr><td>[:RECeive]</td><td></td><td></td></tr>
<tr><td>:BAUD</td><td><numeric value> [9600]</td><td></td></tr>
<tr><td>:PARity</td><td>EVEN | ODD | [NONE]</td><td></td></tr>
<tr><td>:BITS</td><td>7 | [8]</td><td></td></tr>
<tr><td>:SBITs</td><td>[1] | 2</td><td></td></tr>
<tr><td>:RS485</td><td>[1] | 0</td><td></td></tr>
<tr><td>:NETwork</td><td>[OFF] | ADDRess | PACKet</td><td></td></tr>
<tr><td>:ADDRess</td><td>0-15 [4]</td><td></td></tr>
<tr><td>:UPdate</td><td></td><td></td></tr>
<tr><td>:ERRor?</td><td>(0, "No error")</td><td></td></tr>
<tr><td>:VERSion?</td><td>(1994.0)</td><td></td></tr>
<tr><td>CONFigure</td><td></td><td>Configure Strings</td></tr>
<tr><td>[:DIGital]</td><td></td><td></td></tr>
<tr><td>:INPut</td><td><channel list> [@1:6]</td><td>N</td></tr>
<tr><td>:POLarity</td><td>0 | 1 [1]</td><td>TP</td></tr>
<tr><td>:HANDshake</td><td>OFF | ON [ON]</td><td>TH</td></tr>
<tr><td>:OUTput</td><td><channel list> [0]</td><td>LN</td></tr>
<tr><td>:POLarity</td><td>0 | 1 [1]</td><td>LP</td></tr>
<tr><td>:HANDshake</td><td>OFF | ON [OFF]</td><td>LH</td></tr>
<tr><td>:CLEar</td><td>0 | 1 [0]</td><td>C</td></tr>
<tr><td>:EDR</td><td>0 | 1 [0]</td><td>E</td></tr>
<tr><td>:INHibit</td><td>0 | 1 [1]</td><td>I</td></tr>
<tr><td>:REMote</td><td>0 | 1 [0]</td><td>R</td></tr>
<tr><td>: RESet</td><td>0 | 1 [0]</td><td>X</td></tr>
<tr><td>:STRobe</td><td>0 | 1 [0]</td><td>S</td></tr>
<tr><td>:TRIGger</td><td>0 | 1 [0]</td><td>TR</td></tr>
<tr><td>:ASTATus</td><td>0 | 1 [1]</td><td>A</td></tr>
<tr><td>:BSTATus</td><td>0 | 1 [1]</td><td>B</td></tr>
</table>

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

<table>
<tr><th>Keyword</th><th>Parameter Form</th><th>Notes & Single Letter Commands</th></tr>
<tr><td>FORMat</td><td></td><td>Format Strings</td></tr>
<tr><td>[:DATA]</td><td></td><td></td></tr>
<tr><td>:TALK</td><td>ASCii | HEX | HEXL | TABLE | BINary
[HEX]</td><td>
FT</td></tr>
<tr><td>:TRANSlation</td><td><16 char string>
[0123456789:;<=>?]</td><td>V</td></tr>
<tr><td>:LISTen</td><td>ASCii | HEXL | HEX | 4833 | BINary
[HEX]</td><td>
FL</td></tr>
<tr><td>ROUTe</td><td></td><td>Bit Manipulation</td></tr>
<tr><td>:CLOSe</td><td>byte,bit</td><td>CLOSE</td></tr>
<tr><td>:OPEN</td><td>byte,bit</td><td>OPEN</td></tr>
<tr><td>:RESET</td><td>byte</td><td>BRESET</td></tr>
<tr><td>SENSe</td><td></td><td>Bit Sense</td></tr>
<tr><td>:BIT?</td><td>byte,bit</td><td>READ?</td></tr>
<tr><td>:BYTE?</td><td>byte</td><td>BREAD?</td></tr>
<tr><td>[:DIGital]</td><td>Port Input</td><td></td></tr>
<tr><td>:DATA</td><td></td><td></td></tr>
<tr><td>[:VALue]?</td><td>format dependent</td><td>PI?</td></tr>
<tr><td>:PORT?</td><td>number or <channel list></td><td>BI?</td></tr>
<tr><td>:PORTn?</td><td></td><td>BIn?</td></tr>
<tr><td>:POLarity?</td><td>0- 255</td><td>IPn</td></tr>
<tr><td>:RESet:EDR</td><td></td><td>ER</td></tr>
<tr><td>[SOURce]</td><td></td><td>Port Output</td></tr>
<tr><td>[:DIGital]</td><td></td><td></td></tr>
<tr><td>:DATA</td><td></td><td></td></tr>
<tr><td>[:VALue]</td><td>format dependent</td><td>PO</td></tr>
<tr><td>:PORTn</td><td>0-255</td><td>BOn</td></tr>
<tr><td>:POLarity</td><td>0-255</td><td>OPn</td></tr>
<tr><td>:STRobe</td><td></td><td>SP</td></tr>
</table>

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

Keyword	Parameter Form	Notes & Single Letter Commands
STATus		
:OPERation		Status Inputs, WTG
[:EVENt]?	bits 0-2,5,8 and 9 active (0)	
:CONDition?	bits 0-2,5,8 and 9 active (0)	
:ENABle	bits 0-2,5,8 and 9 active (0)	
:ENABLE?		
:PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:QUEStionable		Digital Inputs
[:EVENt]?	bits 0-14 active (0)	
:CONDition?	bits 0-14 active (0)	
:ENABle	bits 0-14 active (0)	
:ENABLE?		
: PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:PRESet		
INITiate		**Trigger**
[:IMMeditate]		TI
:CONTinuous	1(On)| 0(Off) [0]	TC
ABORt		TA
CALibrate		**Calibrate**
:IDN	string	72 char max
:DATe	mm/dd/yy or mm/dd/yyyy	
:DEFault		
:LOCK	1(On)| 0(Off) [0]	

3

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

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 4863's E^2PROM (with the *SAV command).
7. The format for a SCPI list is (@ 1,2, n) or (@ 1:n)
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 4863's E^2PROM.
11. The CAL:DEFault command resets the Flash 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.
13. Serial interface always responds to every command with a response message. For direct interface to a terminal or to a PC running a terminal emulation program, the serial interface also supports echo-to-host and backspace characters. Echo-to-host available on RS-232 interface only. Backspace character removes the preceding serial character.

    CTRL-E turns echo On.

    CTRL-F turns echo Off. (Power-on default condition)

    The serial interface generates the following prompts when ready for the next serial command.

Echo-back Mode	RS-232 Prompt	RS-485 Prompt	Message Terminator
On	CR LF > sp	n/a	CR LF
Off	>LF	>LF	LF

TABLE 3-3 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 and dual primary address mode	Sets GPIB primary address. Values = 0 - 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 4863 primary address.
:EXTernal	OFF	Enables an external address switch to be used to input a switch setting for the GPIB address instead of the value saved in Flash. Values = 0|1 or OFF|ON.
:MODE	SINGle	Selects GPIB address mode. Single mode uses one primary address to transfer data with commands. Dual enables two consecutive primary addresses. Secondary uses one primary address and secondary address 00 and 01. The higher address is used for transparent data transfer. Values = SINGle | DUAL | SECondary. **Provide 70 ms delay after a mode change to next command.**
:OFFset	1	In the Dual Primary Address Mode, the OFFset parameter is added to the unit's primary GPIB address to determine the upper or higher GPIB address. **The user must select the OFFset and the unit's Primary Address so that the upper GPIB address is an unused GPIB address within the 1 to 30 address range.** The variable OFFset parameter is not used in the Dual Secondary Address Mode. Value is 1 to 10.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:SERial		Controls Serial Interface settings
:BAUD	9600	Sets serial baud rate. Values = 300 to 115,200 in all standard rates. Parser uses closest divider for non-standard rates.
: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 number of stop bits between characters. Value = 1 | 2.
RS485	1	Enables full duplex operation on the RS-485 port. Must be set to 1. Values are 0 and 1.
:NETwork	OFF	Selects RS-485 communication protocol. OFF selects no protocol and uses standard ASCII character messages for point-to-point operation. ADDRess enables ICS's network protocol. When network is enabled, every serial message has to be preceded by a two character, <STX><Addr> sequence and is terminated with a linefeed character. PACKet enables ICS's packet protocol. When packet protocl is enabled, the 2363 only recognizes packets sent to it at its network address and with the corect check-sum. The 2363 responds to each valid packet with an ACK, NAK or response packet. Values are OFF, ADDRess and PACKet. Use an :UPdate command or power cycle the unit to switch to the new protocol setting. Note: Earlier units did not have the packet protocol and their value choices were OFF and ON where ON enabled the network address mode.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:ADDress	4	Sets network address. Values are 0 to 15 for <Addr> character values of 30 to 3F hex. **Provide 70 ms delay after an address change before next command.** Note - The serial address and the GPIB address are the same parameter. Changing either one, changes both settings.
:UPdate	-	Updates UART with current serial settings. Changes take place after the prompt. **Allow 100 ms for changes to take affect.**
:ERRor?	0, "No error"	Requests next entry in 4863's error/event queue. Error messages are: 0, "no error" -100, "Command error" -200, "Execution error" -400, "Query error"
:VERSion?	1994.0	4863 returns the <value> of the applicable SCPI version number.
CONFigure		Starts string configuration branch
DIGital		Optional digital data identifier
INPut	(@1:6)	Defines bytes on the digital interface that are used to create the talk string. Bytes can be in any order. Value is a channel list with the format (@ 1,2,3) or (@1:3).
POLarity	1	Sets logical true level for input data. Requires that inputs be assigned first. Values is 0 or 1.
HANDshake	ON	Enables or disables input data handshaking. Disable handshaking to read static signals like switches. Values are OFF | ON.
OUTput	0	Defines bytes on the digital interface that are used to output data from the listen string. Bytes can be in any order. Value is a channel list with the format (@ 1,2,3) or (@1:3).

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

3

Keyword	Default Value	Description
POLarity	1	Sets logical true level for output data. Requires that outputs be assigned first. Values is 0 or 1.
HANDshake	OFF	Enables or disables output data handshaking. Enable handshaking to output data to devices that only accept data at certain times or that need to handshake in data. Values are OFF | ON.
CLEar	0	Sets the active level of the 4863 Clear output pulse. Values are 0 | 1.
EDR	0	Sets the active level of the EDR input signal (both EDRs in 4803). A value of 1 selects positive signal transition. A value of 0 selects negative signal edge. Values are 0 | 1.
INH	1	Sets the active level for the Inhibit output signal (both Inhibits in 4803). Values are 0 | 1.
REMote	0	Sets the active level for the Remote output signal. Values are 0 | 1.
RESet	0	Sets the active level of the Reset output pulse. Values are 0 | 1.
STRobe		Sets the active level of the Data Strobe output pulse. Values are 0 | 1.
TRIGger	0	Sets the active level of the Data Strobe output pulse. Values are 0 | 1.
ASTATus	1	Sets the logical true level for the Status A input signal. Values are 0 | 1.
BSTATus	1	Sets the logical true level for the Status B input signal. Values are 0 | 1.

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
FORMat		Starts string format branch.
DATA		Optional digital data identifier
:TALK	HEX	Sets talk string format. ASCII expresses a byte's input bit pattern as a decimal value equal to the binary sum of the input bits. Multiple bytes are separated by commas. See notes for an ASCII example. HEX converts the two four-bit nibbles in an input byte into the ASCII characters 0-9 and A-F and places two characters into the GPIB output buffer. HEXL is HEX format with a comma inserted between input bytes. TABLE allows the user to define his own conversion table for converting input nibbles. All strings end with a linefeed. BINary places each input byte directly into the GPIB output buffer without any conversion. BINary output is terminated by asserting EOI on the last character. Values are ASCii | HEX | HEXL | TABLE | BINary.
TRANSlation	0123456789 :;<=>?	Defines the HEX to ASCII conversion used for TALK strings when the TABLE format is selected. Value is a 16 character string without separators. i.e. 0123456789+-.,E? **May not be concatenated with another command.**
LISTen		Sets listen string and data output format. ASCii converts a decimal value into an eight bit binary bit pattern by bit weights. Multiple byte values are separated by commas. HEX converts the incoming ASCII characters 0-9 and A-F into the equivalent four bit HEX values. 4833 converts incoming ASCII characters in the 30 - 3F ASCII subset (0-9 and :;<=>?) into the equivalent four bit HEX values as was done in ICS's Model 4833 Parallel Interface. HEXL allows the

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
LISTEN Continued		user to insert commas between hex byte values. BINary outputs binary bytes in dual or secondary address mode. There are no separators with the BINary format and input strings end with EOI asserted on the last byte. Output handshaking not checked. BINary format generates an Output Strobe when all output bytes have been updated. Values are ASCii | HEX | HEXL | 4833 | BINary i.e. ASCii example = 161,35,69 HEX example = A12345 HEXL example = A1,23,45 4833 example = + 1 2 3 4 5 BINary example = bbb
ROUTe		Starts bit manipulation branch
:CLOSe	-	Sets a bit to the logic true value set by the SOURCe:DATA:PORTn:POL command. Bit is defined by a byte,bit parameter. Byte value is 1 to n, bit value is 0 to 7 (MSB).
:OPEN	-	Resets a bit to the logic false value set by the last SOURCe:DATA:PORTn:POL command. Bit is defined by a byte,bit parameter. Byte value is 1 to n, bit value is 0 to 7 (MSB).
:RESET	-	Resets all bits in the byte to their logical false value set by the last SOURCe:DATA:PORTn:POL command. Byte value is 1 to n bytes
SENSe		Starts direct digital byte input branch.
:BIT?	-	Reads bit selected by the byte,bit parameter. Bit levels set by the last SENSe:DATA:PORTn:POL command. Bit is defined by a byte,bit parameter. Byte value is 1 to n, bit value is 0 to 7 (MSB). Response is 0 and 1.

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:BYTe?	-	Reads byte selected by the byte parameter. Bit logic levels set by the last SOURCe:DATA:PORTn:POL command. Byte value is 1 to n bytes. Response is 0 to 255.
[:DIGital]	-	Optional digital data identifier
:DATA	-	Digital data identifier
[:VALue]?	-	Reads input ports selected by Configure branch. Format set by FORMat:TALK setting.
:PORT?	-	Returns value of bytes specified in the attached number or channel list. Format set by FORMat:TALK setting. Value is 1 to 6
:PORTn?	-	Queries the value of byte n. Format set by FORMat:TALK setting. Value of n is 1 to 6
:POLarity	255	Sets input byte logic true polarity for the SENSe branch. Bits may be set independently. Value is 0 to 255. May be queried to read current polarity.
:RESet:EDR		Resets 4863 EDR or 4803 EDR #1 flip-flop without reading the input data.
[SOURce]		Starts direct digital byte output branch Optional branch identifier
[:DIGital]	-	Optional digital data identifier
:DATA	-	Digital data identifier
[:VALue][?]	-	Writes data string to output bytes set by the Configure branch. Format set by FORM:LISTen setting. Can be queried to read back last output value.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:PORTn	-	Writes the attached value to byte n. Value of n is 1 to 6. Value of parameter is 0 to 255. Bits set to the logic true polarity set by the last SOUR:DATA:PORTn:POL command. May be queried to read back last output value.
:POLarity	255	Sets logic true polarity for ROUTe and SOURce branches. Bits may be set independently. Value is 0 to 255. May be queried to read current polarity.
:STRobe		Pulses the Data Strobe without having received string data. Use to transfer SOURce data to the external device.
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	255	Sets positive transition enable register. Value = 0 to 255.
:NTRansition	0	Sets the negative Transition register. Values - 0 to 255.
:PREset		Sets the selected Enable Register, PTR and NTR registers to their default values (0, 255 and 0 respectively) so the 4863 detects a positive changes

3

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
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>	ICS Electronics	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,4863, S/N 012345, Rev 0.6 (05-19-06)**.
:DATe <date>	00/00/0000	Saves IDN message and date. The save operation lights all the LEDs. Date is in mm/dd/yy format.
:DATe?		Queries the calibration date. The response is 00/00/0000 for factory default settings.
DEFault		Sets Flash to factory settings.
:LOCK	0	Disables configuration commands when On. Values = 0|1 or OFF|ON. Tables 1-4 and 1-5 list the locked commands.

Notes - (1) Setting changes are immediate after command execution.
Provide 15 msec. delay for internal reconfiguration.
(2) Setting other than the default value are not saved permanently.

3

3.7 SHORT FORM COMMANDS

The 4863/2363 accepts short form commands as well as the SCPI commands to configure its digital interface and to transfer data bytes. The short form commands are one to three characters long and are not case sensitive. The short form commands have the advantage of reduce the typing load on the programmer when operating the interface from a computer or keyboard. In a program they reduce GPIB bus or serial link's transmission time and the 4863/2363's parser execution time.

Table 3-2 shows the short form commands alongside the SCPI commands. Their parameter form is the same as that of the SCPI commands. A space is required between the command and the parameter. Short form commands ending with a lower case n are really six commands, one for each byte. The SCPI command descriptions in Table 3-3 apply to the short form commands. The following are some short form command examples:

e.g. **TP 1**
is the same as
CONFigure:INPut:POLarity 1

N (@1:3)
is the same as
CONFigure:INPut (@1:3)

BO5 32
is the same as
SOURce:DATA:VALue:PORT5 32
or **DATA:PORT5 #h20** using hex and without optional words.

BI3?
is the same as
SENSe:DATA:PORT3?

Note: Most Short Form commands can be quired by adding a '?' to the end of the command. All queries should be followed by reading their response to avoid data loss and a command execution error.

3.8 PROGRAMMING GUIDELINES

This section shows the user how to program the 4863/2363 to transfer data, to configure the unit, how to use the IEEE-488.2 Status Reporting Structure to generate SRQs, binary data transfer, programming the IDN message and saving the configuration. For additional examples, check the Application Notes page at http:\\www.icselect.com.

3.8.1 Simple Port and Bit Commands

The Digital I/O lines can be programmed or read without configuring the unit. The SOURce:DATA:PORTn command writes directly to any byte. The ROUTe commands set or clear a specific output bit. The SENSe:DATA:PORTn? query reads from any byte and SENSe:BIT? reads a specific bit. These commands automatically set the port as an input or as an output. The SENSe and SOURce branches have a :POLarity command that sets each bit's logically true state.

3

The following commands set bits on the Digital I/O connector:

SOURCe:DATA:PORT1 5	'sets CH1 and CH3 on using the default polarity
SOURCe:DATA:PORT2:POL #h00	'sets byte 2 to low true polarity
SOURCe:DATA:PORT2 0	'sets all byte 2 lines false (high)
SOURCe:DATA:PORT2 #h13	'sets CH 9, 10 and 13 true (low) (bits 0,1and 4)
SOURCe:DATA:PORT2?	'reads last output value
ROUTe:CLOSe 2,1	'sets bit 2 on byte 2 true
ROUTe:OPEN 2,5	'sets bit 6 on byte 2 false

The following commands read data from the Digital I/O connector. The input lines will be high unless connected to some device.

SENSe:DATA:PORT3	'reads state of byte 3 lines
SENSe:BIT? 3,0	'reads bit 1 on byte 3

To read contact closures to ground at byte 3, set the polarity to low true.

SENSe:DATA:PORT3:POL 0	'sets byte 3 polarity for low true

3.8.2 Program Example

This section shows how to use the configuration the 4863 to control the example device shown in Figure 3-3 and used to make the example cable described in Section 2.10. Before you start, review Section 2.10 to learn how the 4863 is connected to the example device.

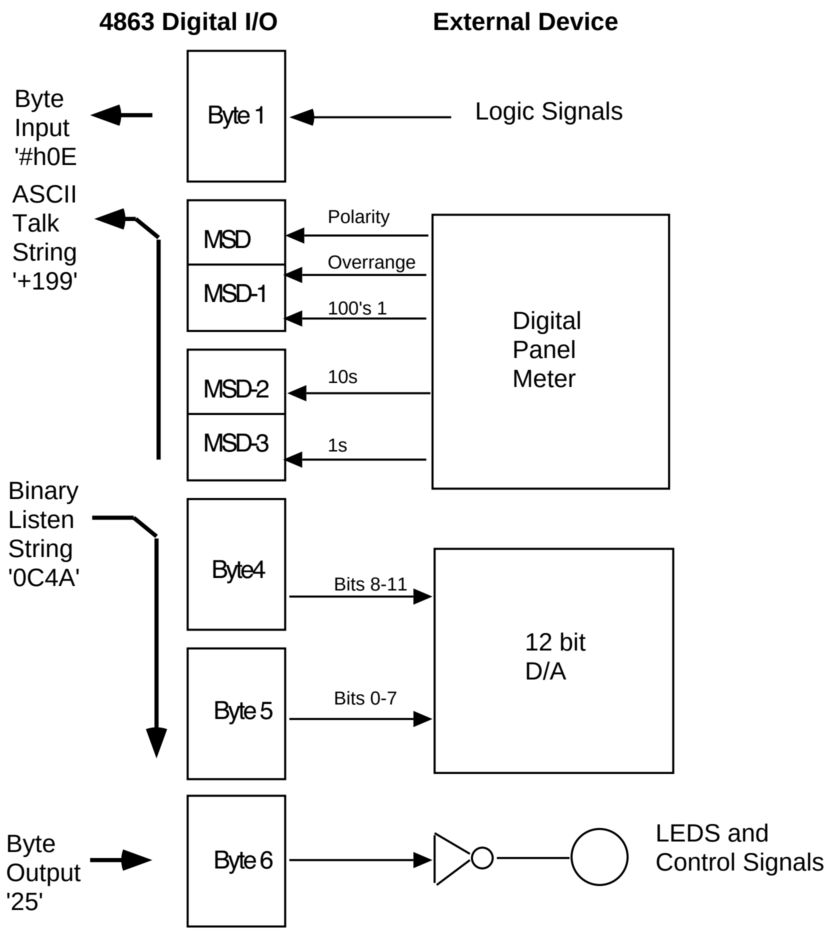


Figure 3-3 Example Digital I/O Connections

3.8.2.1 Configuring for a Talk String (Digital Inputs)

The SCPI CONFIGURE and FORMAT subsystems are used to configure the digital interface for an input string before inputting data with string commands. In Figure 3-3, bytes 2 and 3 are used to input 16 bits of data as BCD/HEX characters from a digital panel meter (DPM). The DPM has a 2 1/2 digit output with a polarity signal and a conversion done signals The data signals are positive true and the conversion signal is a high true BUSY pulse. The DPM accepts a low true HOLD signal to inhibit future conversions. The configuration commands are:

e.g.	**CONFigure:INPut (@ 2,3)**	'selects bytes 2 and 3
	CONFigure:INPut:POL 1	'high true input data
	CONFigure:INP:HAND ON	'enables input handshake
	CONFigure:EDR 0	'for low going input when BUSY ends
	CONFigure:INH 0	'for low going INHIBIT output to hold the DPM

Note that the above commands use the values recorded in Table 2-10 when the example cable was designed.

The 4863 uses the DPM's polarity output signal to generate plus and minus signs in the talk data string. The technique to do this is to jumper three of the four input lines for a nibble so that the DPM's polarity signal will switch between two values and therefore two characters in the Talk Translation Table. In the example, the digital inputs are wired so the input code is 101P with P being the DPM's Plus signal. When Plus is true, the input code is 1011 or a HEX B. When plus is false, the input code is 1010 or a HEX A. The custom Talk Translation Table will include the ten numbers (0-9), and replace the A and B characters with - and +. The characters " ., E" and a space will be used to finish the table. The format commands then are:

e.g. **FORMat:TALK TABLE** 'selects a custom conversion
FORMat:TALK:TRANS 0123456789-+,.E space
'specifies the custom table.

Note that space is a space character, not the letters **s p a c e.**

3

3.8.2.2 Configuring the Listen String (Digital Outputs)

The SCPI CONFIGURE and FORMAT subsystems are used to configure the digital interface before an output string can be used to output data to an external device. The FORMat:LISTen command provides the user with four formats for outputting data to the 4863. ASCii lets the user send the byte output value as decimal numbers from 0 to 255. The numbers are separated by commas. HEXL is similar but it uses two HEX characters from 00 to FF also separated by commas. HEX is the same as HEXL but without the commas between bytes. 4833 is like HEX but it uses the 4833 rule of only accepting ASCII characters with hex values of 30 thorough 3F. These are the numbers 0-9 and :;<=> and ?. The following example shows the four formats being used to output 24 bits of data or three bytes:

e.g. Outputting (0000 0001 0001 0111 1111 1110) with four formats:

Format	Sequence
ASCii	1,23, 254
HEXL	01,17,FE
HEX	0117FE
4833	0117?>

The output string can be sent using the SOURce:DATA command using a single primary address. The output string can also be sent as a transparent string using a second primary address or the upper secondary address.

For most applications, the user will send one set of data in a single command that is terminated with a linefeed or by asserting EOI. However, in some applications, the user might want to send multiple values to the output ports. The 4863 will accept multiple sets of output values in the same command line and generate data strobes for each set as long as they are separated by commas. An output strobe pulse will be generated for each comma between the sets of data. List formats like ASCii or HEXL that use commas to separate bytes, require a double comma to generate an output strobe. If the comma is missing, the 4863 will not generate a data strobe. The 4863 will repeat outputting data to the output ports listed in the last CONFigure:OUTput command until the command terminator is reached. Some examples are:

Format	Sequence
ASCii	1,23, 254,,129, 255,1 <nl>
HEX	1234,ABCD,1AC2 <nl>

In Figure 3-3, bytes 4 and 5 are used to output 12 bits of data to a D-to-A converter. The data output is two bytes of low true bits with a low true pulse to load the data into the D-to-A convert's latch. HEX characters will be used to transfer the data as they are easy to program and debug. No handshaking is required. The example configure and format commands are:

e.g.	**CONFigure:OUTput (@ 4,5)**	'selects bytes 4 and 5
	CONFigure:OUTput:POL 0	'low true input data
	CONFigure:OUT:HAND OFF	'no output handshaking
	CONFigure:STR 0	'sets low going data strobe
	FORMat:LISTen HEX	'selects HEX coded data

3.8.2.3 Outputting Data Strings

3

The output data string for the above commands is four HEX characters terminated with linefeed character or with EOI asserted on the last character. The data can be sent with the SOURce:DATA command if the 4863 is set for a single primary address or if the lower primary or secondary address is used. If dual addresses are used, the output data string can be sent to the upper primary or secondary addresses without a command. Use the GPIB Controller's Output or Send commands to send the commands of data strings. Some output examples are:

e.g.	**Output(dev, "SOURce:DATA 12FC")**	'sends 4 HEX characters to 4863's single primary address

To send data transparently without a command, first put the unit in the dual or secondary address mode, then send data to the second or upper address.

e.g.	**Output(dev, "SYST:COMM:GPIB:ADDR:MODE DUAL")**	
	Output(dev+1, "12FC")	'Dual Primary address example
or		
	Output(dev, "SYST:COMM:GPIB:ADDR:MODE SEC")	
	Output(dev+sec01, "12FC")	'Secondary address example

In the examples, some of the commands may be the same as the factory default settings. They are shown for completeness. Include all of the configuration parameters in the configuration code in case a different value was stored in the 4863's Flash.

3.8.2.4 Outputting Data with Listen Handshake Enabled

When Listen Handshaking is enabled, the 4863 tests the Status A input and only outputs data if the Status input is in its true state. Else the 4863 will report an Execution Error and discard the data. The recommended procedure is for the user to test the Status A input before outputting data if Listen handshaking is enabled. If multiple sets of data are being outputted to a device with Listen Handshaking enabled, the 4863 will wait 100 msec for the external device to complete its handshaking before starting the next data set. If the external device does not complete its handshake within the 100 msec period, the 4863 will abort the command and report an Execution Error.

3

3.8.2.5 Configuring Individual Input Bytes

It is not necessary to configure individual bytes with the Configuration commands. Individual input bytes are configured and read with the SCPI SENSE subsystem. The :PORTn? command automatically sets the byte for inputting data when it is read. The :POLarity command configures the input polarity on a bit-by-bit basis. In the example cable, the bits are all high true. The example sense commands are:

```
e.g.  SENSe:DATA:PORT1:POLarity #hFF     'sets all bits high true
      SENSe:DATA:PORT1?                  'reads byte 1
```

3.8.2.6 Configuring Individual Output Bytes

It is not necessary to configure individual bytes with the Configuration commands. Individual output bytes are configured and outputted with the SCPI SOURCE subsystem. The SOURce:DATA:PORTn command sets the byte as an output when data is written to the byte. The :POLarity command configures the output polarity on a bit-by-bit basis. For the sample cable, the output bits are all low true. Polarity is programmed first so the example source commands are:

```
e.g.  SOURce:DATA:PORT6:POLarity #h00    'sets all bits low true
      SOURce:DATA:PORT6 0                'selects byte 6 for output,
                                          and sets bits to 0
```

3.8.2.7 Configuring the Control Signals

The 4863's control signals are configured with the CONFIGURE subsystem. In the example, Clear is wired into the D/A to reset it to zero, Reset is wired to the DPM to reset the meter at power turn-on time. Trigger could be used by the DPM or it could be set to free run. The Status A input is connected to 5 Vdc on the external board to monitor the external power supply. Status B and the Remote output signal are not used in the example. The example configure commands are:

```
e.g.   CONFigure:CLEar 0          'selects low true output
       CONFigure:REMote 0         'selects low true output
       CONFigure:RESet 0          'selects low true output
       CONFigure:TRIGger 0        'selects low true output
       CONFigure:ASTATus 1        'selects high true input
       CONFigure:BSTATus 0        'selects low true input
```

3

3.8.2.8 Reading the Digital Input String

Now that the talk string has been defined and digital interface has been configured, it is time to input data. Be sure that the external device is pulsing the External Data Ready line or that talk handshaking is disabled. To read the data with a command use the SENSe:DATA:PORT? or PI? query. Read the response from the lower GPIB address or secondary address 00.

```
e.g.   CALL OUTPUT(dev, "SENSe:DATA:PORT?")
       CALL ENTER(dev, Rdg$)           'Dual Primary address
                                         example
or
       CALL OUTPUT(dev+sec00, "SENSe:DATA:PORT?")
       CALL ENTER(dev+sec00, Rdg$)     'Secondary address
                                         example
```

For transparent data transfer, use the enter, input or receive command in your GPIB command library to input the data from the upper primary address if in the Dual Primary Address mode or from secondary address 01 if in the Secondary Address mode).

```
e.g.   CALL ENTER(dev+1, Rdg$)         'Dual Primary address
                                         example
or
       CALL ENTER(dev+sec01, Rdg$)     'Secondary address
                                         example
```

If the 4863's input string is being read transparently, **it is necessary to Untalk the 4863 between readings.** This can be done by addressing another instrument, by serial polling the 4863, by sending the 4863 a command or by adding a Untalk to your program. Most GPIB command libraries have a CMD or SEND instruction that can output an Untalk command (UNT or the ASCII _ character).

```
e.g.   CALL ieSend("UNT")              'ICS's 488-PC2 syntax
       CALL SendCmd(Bd%, "\x5F",1)     'NI 488.2 syntax
```

If you do not receive data, query the Event Status Register (ESR) to determine the state of the 4863's EDR flip-flop. (It must be on to read data if handshaking is enabled.). If ESR bit 6 is set, the data is ready to be read. If the bit is not set, the EDR input line was not pulsed.

3

```
e.g.   *ESR?                 'Queries register status
       ENTER dev, A          'Reads 4863 response
       AND( A, #h40)         'Tests bit 6
```

3.8.3 Other Functions

The following sections describe other useful 4863 functions.

3.8.3.1 Triggering the External Device

Triggering provides a way to control when an external device performs some action. Triggers can be single shot or continuous. Use the SCPI INITATE subsystem to enable the trigger output.

```
e.g.   INITiate:CONTinuous ON      'Enables continuous triggers
       *TRG                        'Generates the trigger pulse
```

The 4863 responds to a *TRG command or a 488.1 GET command by pulsing its trigger output line. Any other action depends upon the external device. Provide a delay for the triggered action to occur before reading the input data.

3.8.3.2 Using SRQs to Input Data

The 4863 can be set to monitor the status inputs, digital lines or the EDR flip-flop and generate an SRQ (2363 generates a SRM) when selected input signals change state. The signal changes are collected in the individual event registers and summarized in the Status Byte Register. Refer to the 488.2 Status Structure in Figure 3-2.

When the SRQ is sensed, the user queries the Status Byte and then the appropriate event register to determine the cause of the SRQ. The following example generates an SRQ whenever the EDR flip-flop becomes set (external data is ready) or the Status A signal goes low (In the example of 2.10, the external device has lost power).

Setup:

```
STAT:OPER:PTR 4          'enables bit 2 to set on EDR
STAT:OPER:NTR 1          'enables bit 0 to set on a nega-
                         tive transition of Status A
STAT:OPER:ENAB 5         'enables Event bits 0 and 2
*SRE 128                 'Status Byte bit 7 enabled
```

After the SRQ occurs:

```
Serial Poll the 4863
If Status Byte bit 5 is on then
    Read the ESR Register
   If ESR bit 6 then
      Read the input data
   Else
     Report error!!!
  End If
Else If Status Byte bit 7 is on then
    Print "External Unit off line" + (TIME)
     Beep
Else
     Report error!!!
End If
```

3

3.8.3.3 Outputting Binary Data (4863 Only)

The 4863's BINary format outputs 8-bit bytes to the 4863's configured output ports and generates a Output Data Strobe when all ports that are configured as outputs have been updated. The BINary data format works with transparent data in the Dual or Secondary address mode. The output message size must be a multiple of the number of configured output bytes and must be terminated with an EOI on the last character. Data rates can be > 50 Kbs. The following example configures two ports as outputs and outputs four 16-bit words:

Setup:

```
CONF:IN (@ 1:6)          'sets all ports as inputs
CONF:OUT (@ 1,2)         'sets two output ports
FORM:LIST:BIN
SYST:COMM:GPIB:ADDR:MODE DUAL
                         'enables two primary addresses
```

3

Sending 8 data bytes to the upper primary address:

```
23102301                 'data ends with EOI on last byte
```

3.8.3.4 Locking the Setup Parameters

All of the digital interface configuration parameters can be locked to prevent accidental change by the end user. These lockable parameters are noted by a # symbol in Tables 1-4 and 1-5. 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:

e.g.
```
CAL:LOCK ON              'blocks unauthorized changes
*SAV 0
```

While lock is ON, the user can change and save any non-locked parameter such as the GPIB address or a serial baud rate.

3.8.3.5 Configuring the IDN Message

The 4863's IDN message can be changed with the CALIBRATE subsystem commands to personalize the 4863 or OEM board as part of the system or product. The format for the IEEE-488.2 IDN message is four fields separated by commas. The fields are: manufacturer, model code, serial number and revision. The word "model" may not be used in the IDN message. Maximum message length is 72 characters. The IDN message is a lockable parameter and if locked, needs to be unlocked before being changed.

e.g. **CAL:LOCK OFF** 'unlocks all parameters
CAL:IDN company, model, serial number, revision/date
CAL:DATe 06-09-98 'saves new IDN message
'Note-use the current date

3

Use either the CAL:DATe or *SAV 0 command to save the IDN message.

3.8.3.6 Saving the Configuration

The 4863 uses the 488.2 *SAV 0 command to save the current configuration in the 4863's Flash or the 4863's Flash Memory. The saved configuration is automatically recalled at power turn-on time. **CAUTION - Because the Flash Memory has a finite number of write cycles, the *SAV command should not be used inside a program loop.**

***SAV 0** 'saves current values

***RCL 0** 'recalls saved configuration

3.8.3.7 Bit manipulation Commands

The bit manipulations provide a way to directly set, reset and to read individual bits. Bytes are numbered from 1 to 6. Bits are numbered from 0 (LSB) to 7 (MSB). Logic true polarity is set by the SOURce and SENSe polarity commands. Examples are:

ROUTe:CLOSe 1,0 'sets first bit in byte 1
ROUTe:OPEN 2.2 'resets bit 3 in byte 2
SENSe:BIT? 5,7 'reads bit 8 in byte 5

3.9 SERIAL INTERFACE OPERATION DIFFERENCES

The 2363's serial interface is factory set to operate at 9600 baud, with 8 data bits, 1 stop bit and no parity. For user convenience, serial commands that change the UART or Network settings do not take affect until the next power turn-on or until a *RST, *RCL 0, or SYST:COMM:SER:UPdate command is received. To change the serial settings, send the 2363 the new serial parameters, verify them by querying the unit and then save the configuration with the *SAV 0 command. Turn power off and back on or use the *RCL 0 or SYST:COMM:SER:UPdate command to update the UART.

NOTE: You must change the controller's serial settings before sending the 2363 the next command.

3.9.1 Echo and Prompts

The 2363 provides echos and prompts on its RS-232 interface. The 2363 defaults to echo on at power turn-on for keyboard operation. The echo function can be disabled for computer operation by sending the unit a CNTL-F. A CNTL-E turns the echo function on.

For terminal operation, use CNTL-E to turn echo on. If echo is on, the unit will echo back the command or query, output a response if one is required and a prompt.

For program operation, use CNTL-F as the first command to turn echo off. The unit will respond to a command by returning a response if one is called for and a prompt when ready for the next command. The user's program should check and wait for the expected response and then the prompt before sending the unit another command. The following examples show the unit's response to user commands in bold type. The prompt formats are:

Echo back	RS-232 Prompt	RS-485 Prompt	Message Terminator	Prompt Sent
On	CR LF > sp	n/a	CR LF	Yes
Off	> LF	>LF	LF	Yes

3.9.2 Terminal Operation Examples

e.g. Terminal command with echo on:

```
CNTL-E                    'power on default
*ESE 255 CR LF
```
***ESE 255 CR LF**
CR LF > space

e.g. IDN query from a terminal with echo on:

*IDN? CR LF
***IDN? CR LF**
ICS Electronics, 2363, S/N 002123, Rev. 1.1 CR LF
CR LF >

3

3.9.3 Program operation examples

Send CNTL-F before any serial commands to turn echo off.

CNTL-F 'turns echo off

e.g. Program command (echo off):

*ESE 255 LF
> LF

e.g. Program IDN query (echo off):

*IDN? LF
ICS Electronics, 2363, S/N 402123, Rev. X1.X0 LF
> LF

e.g. Baud rate change from a computer program:

SYST:COMM:SER:BAUD 4800 LF	9600 baud
> LF	9600 baud
SYST:COMM:SER:UPDATE LF	9600 baud
> LF	9600 baud
*SAV 0	4800 baud
> LF	4800 baud

3.9.4 RS-485 Protocol

When no Network protocol is selected (SYST:COMM:SER:NET OFF), the RS-485 interface operates like the RS-232 interface does with echo off. See the examples in paragraphs 3.9.2 and 3.9.3.

3.9.5 Addressed Network Protocol

The 2363 supports a multi-device or multidrop network connections when the NETwork Address mode is selected. When Address is selected, the 2363 recognizes commands addressed to it by checking the first two characters of the message for the correct address sequence. The first character must be the STX character. The second character must be the address character (0-9 and :;<=>?). All responses are the same as for normal RS-232 and RS-485 operation. The Address command message format is

STX Addr Message LF

e.g. The IDN query example becomes:

STX 4 *IDN? LF
ICS Electronics, 2363, S/N 910123, Rev. X1.X0 LF
> LF

3.9.6 Packet Protocol

Network messages can be made more secure by enclosing the commands in a packet that is protected with a checksum. When Packet Protocol is selected, the 2363 recognizes commands addressed to it by checking the first two characters of the packet message for the correct address sequence as described in 3.9.5. The Packet is then checksumed and the result compared against the checksum at the end of the packet. If the checksum matches, the command is parsed and executed (assuming it is a valid command). If the checksum does not match, a NAK response packet is sent to the controller.

If Packet Protocol is enabled, messages must conform to the following format:

STX Addr Message Characters.. ..ETX Checksum

Where

STX and ETX are ASCII characters
Addr is the 2363's address
Checksum is a 8 bit checksum created by exclusive ORing all of the characters from the STX to the ETX character.

The 2363 responds to each valid packet with an ACK response packet if the message was a query or with an acknowledgment response if the message was not a query. The command acknowledgment packet contains the ESR register value. e.g.

ACK Addr ESR Register Value ETX Checksum

ACK Addr Optional query response..ETX Checksum

A NAK packet with the following format is returned if the original packet had a bad checksum or if there was a command problem.

NAK Addr ESR Register Value ETX Checksum

Some command-response examples are:

Message	Packet
*CLS	02 34 2A 63 6C 73 03 63
ACK 4 0	06 34 30 03 01
*xxx	02 34 2A 78 78 78 03 67
NAK 4 32	15 34 33 32 03 23
*esr?	02 34 2A 65 73 72 3F 03 44
ACK 4 32	06 34 33 32 03 30
*idn?	02 34 2A 69 64 6E 3F 03 43
ACK 4 ICS-..	06 34 49 43 53 20 45.....2E 32 32 03 08

3

3.9.7 SERkybd Program

ICS supplies an interactive SERkybd program that lets you send commands to any serial unit and receive back responses. SERkybd can operate with basic serial messages, in the Network Address protocol and in Network Packet protocol. SERkybd is in its own directory on the Support CD-ROM that is shipped with the 2363. SERkybd is installed by copying the files in the SERkybd directory to your hard disk. Place a short cut to SERkybd on your desktop.

SERkybd is run by double clicking on the program name or on the shortcut. Next click on the 'Initialize COM Port' button to display the COM Port form. Set the baud rate and click on the appropriate COM port button. Use the default 9600 baud setting unless you have set the serial device to a different baud rate. Click the 'Return to Main Form' button to exit the COM Port form.

3

Enter any commands to be sent to the 2363 in the Device Command window. The default settings add a linefeed to the message and use a basic serial protocol. Press SEND to transmit the command. If the command is a query and contains a '?', the Auto Query function will read the Receive Buffer and display its contents in the Device Response window.

The Network Protocol and Device Address frame contains three radio buttons that let the user select the serial protocol. The default selection is OFF or none. The OFF protocol sends messages just as they would be sent by a terminal emulation program. The Addr selection is for Network Addressing and prefixes a STX character and an address character at the start of each message. The user enters the serial device's address in the window and presses the 'Set Address' button to enter the address into the program. A device address is required for the Addr and Packet protocols. Use the default address of 4 unless you have changed the 2363's address. (See paragraph 2.7 for directions on changing the 2363's address.)

The Packet button selects the Network Packet protocol which causes the SERkybd program to format all massages as a packet and to attach a checksum to each packet. The SERkybd program looks for a response packet from the serial device and displays the packet type (ACK of NAK) received in the Device Response window along with any response message.

The two Control Character buttons let the user send Control-E and Control-F to the serial device. In the 2363, Control-E enables the 2363 to echo back any received messages. Control-F disables the echo function. Echoing is only useful when you are controlling the 2363 from a terminal. The echoed messages only confuse a computer program.

The IDN Message Loop frame lets you run a continuous series of IDN messages to test he serial device. The message rate defaults to twice a second. The Device Command Loop lets you send the command string in the Device Command text box at a twice a second rate. If the string contains a '?', the Auto Query function will read and display a
the serial device's response.

3

3.9.8 Programming Suggestions

For Windows 98, NT, 2K or XP programming, use a late version of Visual Basic (revision 5 or later) with the MSComm control to access the computer's serial ports. Refer to ICS's Support CD for Visual Basic programming examples.

For Windows 3.1 and Win 95 programming, use a serial command library to get control of the computer's serial ports. Recommended communication utility packages are:

COMBIOS from ICS Electronics
Phone 1-800-952-4499

COMM-DRV from Willies Computer Software Co.
2470 So. Dairy Ashford, Houston TX 77077.
Phone 713-498-4832.

For DOS programming, some of the languages have commands that give you access to the computer's serial ports. Two such languages are Quick Basic or Visual Basic for DOS.

3.10 OEM DOCUMENTATION GUIDELINES

Users of the 4863 OEM Boards should provide the end user with the necessary instructions to operate the complete system. In most cases this includes directions for:

1. Setting the GPIB (or Serial) Address.
2. Controlling the overall system. Show which commands the user needs to use to control the system and how to read back data or system status.
3. Using the trigger functions
4. Using the 488.2 Status Reporting Structure. (Define what the digital inputs mean if they are part of the system)

3

The SCPI Standard requires that the SCPI command tree and SCPI conformance information be passed on to the end user. Locked commands become invisible to the end user and can be omitted from the passed on SCPI command tree and list.

OEM users are hereby given permission to copy any portion of this manual for the purpose of documenting systems that incorporate the Model 4863, 2363 or a 4863 OEM Board. Reproduction of this manual for other purposes without the expressed consent of ICS Electronics is forbidden.

4

Theory of Operation

4.1 INTRODUCTION

This section describes the theory of operation of the 4863 and 2363.

4.2 BLOCK DIAGRAM DESCRIPTION

4

A block diagram of the 4863 is shown in Figure 4-1. The 4863 is a microprocessor based device that accepts commands from the GPIB (IEEE 488) bus to control its relay outputs and to read digital inputs. The 4863 is made up of nine 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 accept GPIB command and data characters When the command terminator is sensed, the command characters are parsed and used to charge the 4863’s operational settings or invoke some action or response. The typical action for the Model 4863 is to transfer data from the GPIB or Serial interfaces to/from the digital I/O latches.

The 4863's digital input circuits are designed to accommodate contact closures, TTL logic inputs or CMOS logic inputs. Each of the 48 digital inputs has a 33 Kohm pullup resistor to +5 Vdc. The Status inputs and the EDR input are HCT type inputs.

The Flash memory contains all of the 4863’s program instructions, command tables, and power turn-on/self test routines. At power turn-on, the 4863 performs a self test on each functional block to determine whether there is a gross system failure. Any self test error is displayed by blinking the front panel LED(s).

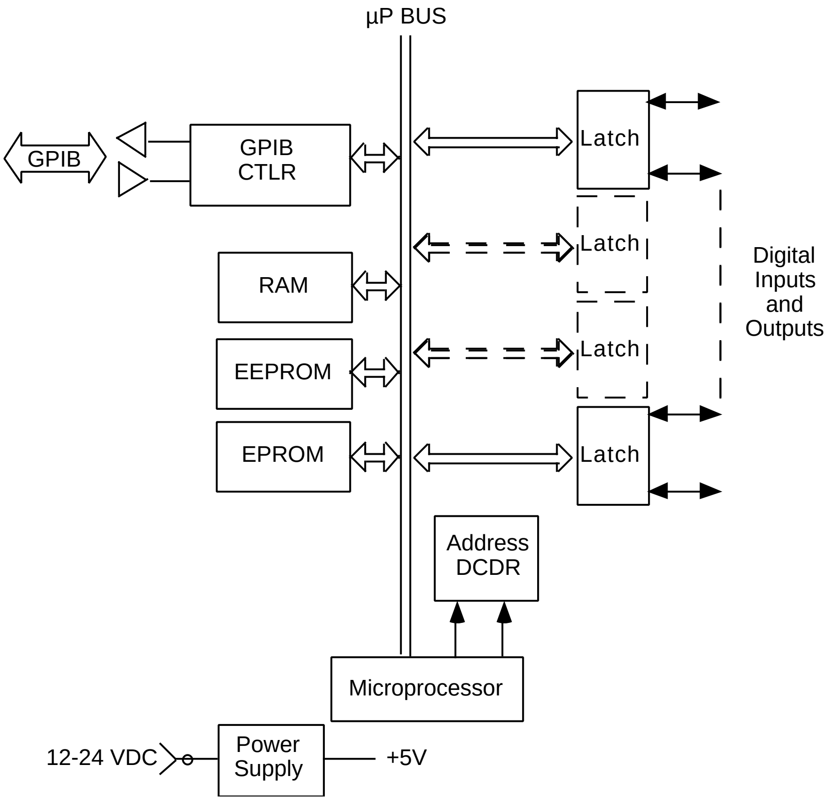


FIGURE 4-1 4863 BLOCK DIAGRAM

The Flash memory also contains all of the 4863'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 memory where it is used to operate the unit. Any changes to the configuration settings are not stored in the Flash memory until the user sends the *SAV command.

4

The RAM is a 8K x 8 static memory that is used to hold the operating variables and configuration settings. All data stored in the RAM memory is lost when power is turned off.

The power supply is a switching regulator that converts any unregulated DC input to +5 Vdc to run the 4863's internal logic chips. Satisfactory input range is +9 to +32 Vdc. Due to the use of CMOS circuits, the 4863's power consumption is very low. Less than 220 mA at 12 Vdc when the outputs are unloaded.

4.2.2 2363 DIFFERENCES

The 2363 has an serial interface in place of the 4863's GPIB interface. The Block Diagram for the 4863 OEM Board in Figure 4-2 shows how the serial interface is connected to the 2363's microprocessor. The RS-232 transceiver incorporates a charge-pump that provides the ± 8 Vdc necessary to drive the RS-232 line. The RS-485 transceiver provides the differential transmitter and receiver functions for the RS-422 or RS-485 differential signal pair. The remainder of the 2363's internal circuits are the same as those of the 4863.

4

The serial interface has both RS-232 and RS-485 capability. The RS-232 is a full duplex interface with transmit and receive signals. There are no handshaking lines because of the short string lengths of the 2363 commands and responses. The RS-485 (RS-422) is a half duplex interface with a single + and - signal pair. The transmitter is only enabled when the 2363 is responding to a query. Pullup and pulldown resistors are provided to bias the RS-485 signals into the logic 1 or mark state and avoid receiver errors.

The serial parameters default to 9600 baud, 8 data bits, no parity and one stop bit. The 2363 detects which serial input is being used and directs query responses back on the appropriate interface.

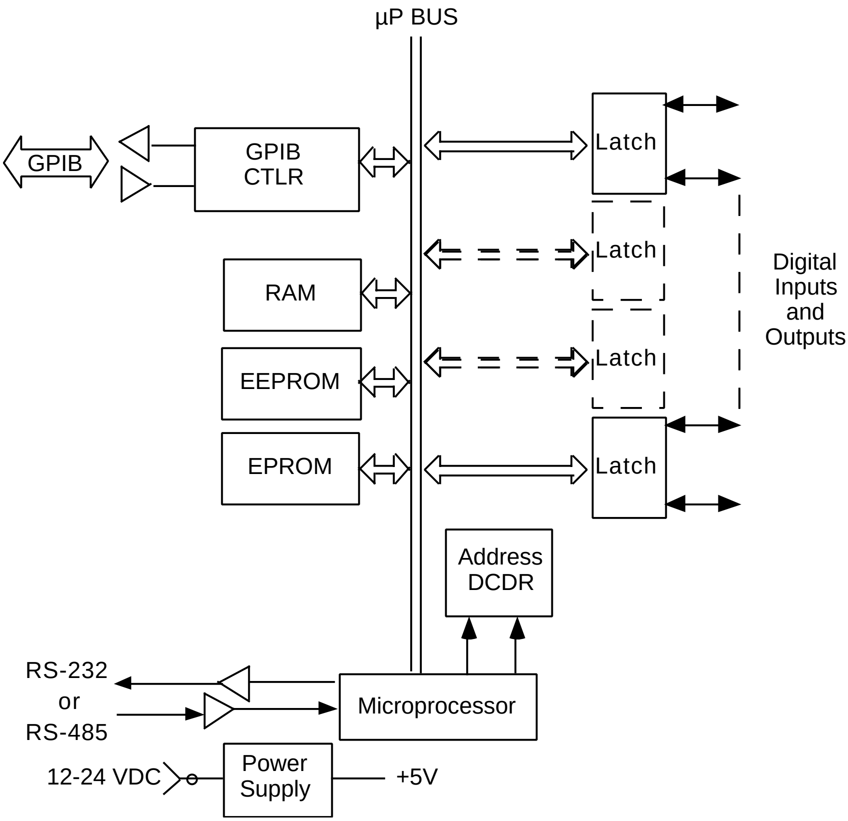


FIGURE 4-2 OEM BOARD BLOCK DIAGRAM

4.2.3 4863 OEM BOARD DIFFERENCES

Versions of the 4863 OEM Board have an additional serial interface as shown in Figure 4-2. The serial interface has both RS-232 and RS-485 capability. The RS-232 is a full duplex interface with transmit and receive signals. There are no handshaking lines because of the short string lengths of the 4863 commands and responses. The RS-485 (RS-422) is a half duplex interface with a single + and - signal pair. The transmitter is only enabled when the 4863 is responding to a query. Pullup and pulldown resistors are provided to bias the RS-485 signals into the logic 1 or mark state and avoid receiver errors.

When the serial interfaces are installed, they are activated at power turn-on and remain active until the unit is addressed on the GPIB interface and put into the Remote state. The serial parameters default to 9600 baud, 8 data bits, no parity and one stop bit. The 4863 detects which serial input is being used and directs query responses back on the appropriate interface.

Extra digital inputs are provided on the OEM board to read a GPIB address from an external address switch. The address inputs are low true contact closures to ground.

Extra digital inputs are provided on the OEM board to read a GPIB address from an external address switch. The address inputs are low true contact closures to ground.

4

This page left intentionally blank

5

Troubleshooting and Repair

5.1 INTRODUCTION

This section describes the maintenance, troubleshooting and repair procedures for the Model 4863 and 2363 Parallel Digital Interfaces. All comments and errors apply equally to both units unless otherwise stated.

5.2 MAINTENANCE

The 4863 and the 2363 do not require periodic calibration and have no internal adjustments. However, if the 4863 is used in an application where the IEEE 488 bus cables are frequently changed or if the input signals appear erratic, the 4863's connectors may require cleaning to remove wax and dirt buildup. New GPIB 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 4863's connector to cause intermittent operation.

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

5.3 TROUBLESHOOTING

Troubleshooting is broken down into self test error and those that are caused during usage.

5.3.1 Self Test Errors

The 4863 and 2363 indicate self test errors by blinking one or more of their LEDs at a 2 Hz rate. Self test is performed at power turn-on time. 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. 4863 Faults and solutions apply to the 2363 also unless otherwise noted.

5.3.2 Operating Failures

Use the fault isolation information in Table 5-2 to narrow the 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 4863/23633 - SELF TEST ERROR CODES

Blinking LED	Error	Possible Fault
All	Flash Memory	Flash Memory corrupted and reset to factory settings. 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. Loose 4863 Flash U9 Defective 4863 Flash U9 that cannot be reprogrammed. Replace U9 Defective decoder PLD. Test and/or replace PLD U8 in 4863 or in 2363.
RDY	Flash	4863 Flash failed write test. Replace Flash, U9, and reprogram configuration. See DEFAULT command.
TALK	Flash	4863 Flash loose in its socket or has a bent pin. Check Flash, U9, for a bent pin then press Flash, U8, into its socket. Flash Memory dropped a bit. Replace with a known good Flash. If the Flash Memory is replaced, the configuration will have to be reprogrammed. Refer to the DEFAULT command. Defective decoder PLD. Test and/or replace PLD, U8 in 4863.
LSTN	RAM	Defective decoder PLD. Test and/or replace PLD, U8 in 4863. Defective RAM. Replace RAM, U10 in 4863,

TABLE 5-1 4863/2363 - SELF TEST ERROR CODES CONTINUED

Blinking LED	Error	Possible Fault
SRQ	GPIB	Defective GPIB controller chip. Replace GPIB Controller, U4 in 4863.
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, U4 with NI 7210.
LSTN + ERR	CPU	Wrong CPU type. Should be Zilog Z8S18020VSC (SL1919 Enhanced Version)
TALK + LSTN + ERR	CPU	Wrong CPU type. Should be Zilog Z8S18020VSC
RDY + LSTN + ERR	PLD U8	Wrong decoder PLD, U8, or wrong version PLD

TABLE 5-1 4863/2363 - SELF TEST ERROR CODES CONTINUED

Solid LED	Error	Possible Fault
PWR (After blinking address)	Program hung	Open GPIB chip selection line or grounded interrupt into Z180
ERR (When RDY on)	Cal Date reset	Flash data reset and all calibration settings lost. Reset defaults, reconfigure the unit and send the unit a new CAL:DATE command with today's date. Note: *CLS will clear the ERR LED

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 configuration data lost	Use *CLS to clear the LED. Use CAL:DATE command to accept default configuration and clear the error so the ERR LED will not come on at next power-on time. Reload your configuration and use the *SAV 0 command to save the new configuration.
Unit fails to respond or responds wrong after an address change or address mode change	No delay after an the address or address mode change	Provide a 70 ms minimum delay after changing 4863's GPIB address or address mode setting.
	Insufficient delay	Program running on faster CPU or in a compiled form runs faster. Change to a called time function and test CPU clock.
String data transfer fails	Wrong GPIB address	Send string data to next higher primary address or to secondary address 01. Read string data from next higher primary address or from secondary address 01.

5

TABLE 5-2 TROUBLESHOOTING GUIDE CONT'D

Symptom	Possible Fault	Action or Check
String data transfer fails Continued	No talk data	EDR flip-flop not set
	Talk handshake on when not needed for static data	Set talk handshake off
	Wrong Format	Selected data format Inserting commas or terminators in HEX or 4833 between data bytes.
	Controller did not receive line feed (Unit hangs with TALK LED on)	Use bus analyzer to check outputted data Set controller to recognize shorter strings to see if any data outputted. EDR missing with Talk handshake enabled.
	Dual Address Mode No talk data	Talk address not recognized unless unit is UNTALKed between messages in Dual Address Mode. Send UNT command after reading data
Polarity setting wrong or changed	Polarity command out of sequence	Assign inputs or outputs before polarity command
Unit fails to respond correctly to National Instruments GPIB Controller Card	REN not asserted	Check Bus REN signal Add ibsre command to program to set the REN line.

5

5.6 RESTORING FACTORY SETTINGS

A 4863 or 2363's can be restored to the factory default settings with the following procedure. Note that you will have to reconfigure the unit for your application after restoring the factory settings.

1. Turn the 4863/2363's power off. Disconnect all cables from the unit.

2. Undo the two screws on the rear cover and slide the PC assembly out from the case. Place it on a nonconductive surface.

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

4. Connect the power supply to the 4863/2363. Turn the 4863/2363's power on. Wait 10 seconds for the unit to finished its save procedure. The LEDs will become stable when the procedure is completed. Turn power off and again disconnect the power supply cable. All configuration parameters have now been restored to the settings listed in Table 1-4 or 1-5.

5. **Remove the jumper from W1** and insert the unit back into its case. Be sure the power switch comes through the front panel and that the case is completely closed before screwing the rear panel to the case. **Do not over tighten the screws**.

6. Reconnect the GPIB (or serial), power and Digital I/O cables to the unit.

7. Turn power on. The unit should go through its self test and leave the red ERR LED on. This indicates that the unit is uncalibrated. Use the CAL:DATE mm/dd/yyyyy command to enter a new configuration date and *CLS to turn off the ERR LED. Use the other SCPI commands in Table 3-2 to set the unit's configuration for your application.

5.7 REPAIR

Repair of the 4863 is done by the user or by returning the unit to the factory or to your local distributor. Units in warranty should **always** be returned to the factory or else repaired only after receiving permission to do so from an ICS customer service representative.

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

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

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

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

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

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

This page left intentionally blank

5

Appendix

Appendix **Page**

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

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

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

A1.1 IEEE 488.1 Bus

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

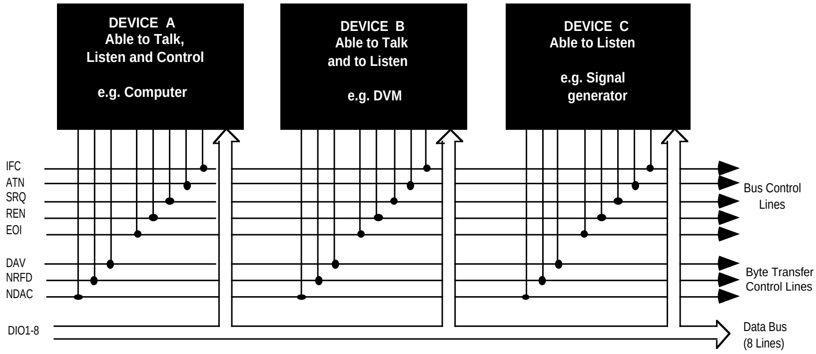


Figure A-1 IEEE 488 Bus

Two types of messages are transferred over the bus:

Interface messages - for bus management

Device-dependent messages - for device control and data transfer

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

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

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

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

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

A

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

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

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

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

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

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

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

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

MSD / LSD	0		1		2		3		4		5		6		7	
	ASCII	MSG	ASCII	MSG	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16	`	▲	p	▲
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	Q	17	a	MEANING DEFINED BY PCG CODE	q	MEANING DEFINED BY PCG CODE
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	▼

Groups: columns 0 = ADDRESSED COMMAND GROUP; columns 1 = UNIVERSAL COMMAND GROUP; columns 2–3 = LISTEN ADDRESS GROUP; columns 4–5 = TALK ADDRESS GROUP; columns 6–7 = SECONDARY COMMAND GROUP. Columns 0–5 = PRIMARY COMMAND GROUP (PCG)

Notes:
1. Device Address messages shown in decimal
2. Message codes are:
   - DCL -- Devices Clear
   - GET -- Device Trigger
   - GTL -- Go to Local
   - LLO -- Local Lockout
   - PPC -- Parallel Poll Configure
   - PPU -- Parallel Poll Unconfigure
   - SDC -- Selected Device Clear
   - SPD -- Serial Poll Disable
   - SPE -- Serial Poll Enable
3. ATN off, Bus data is ASCII; ATN on, Bus data is an IEEE MSG.

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

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

Address Commands

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

Universal Commands (to all devices)

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

Addressed Commands (to addressed listeners only)

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

A

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

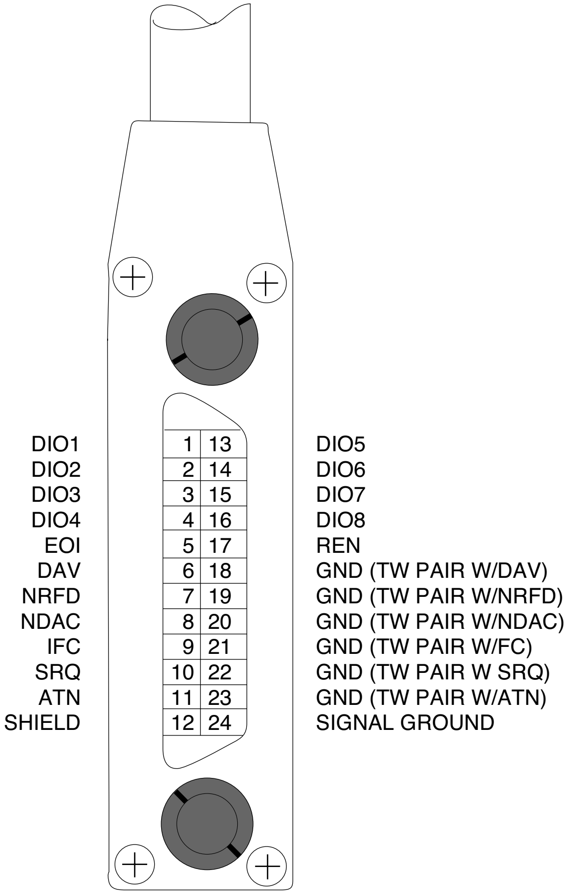


Figure A-2 GPIB Signal-Pin Assignments

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.

A

A

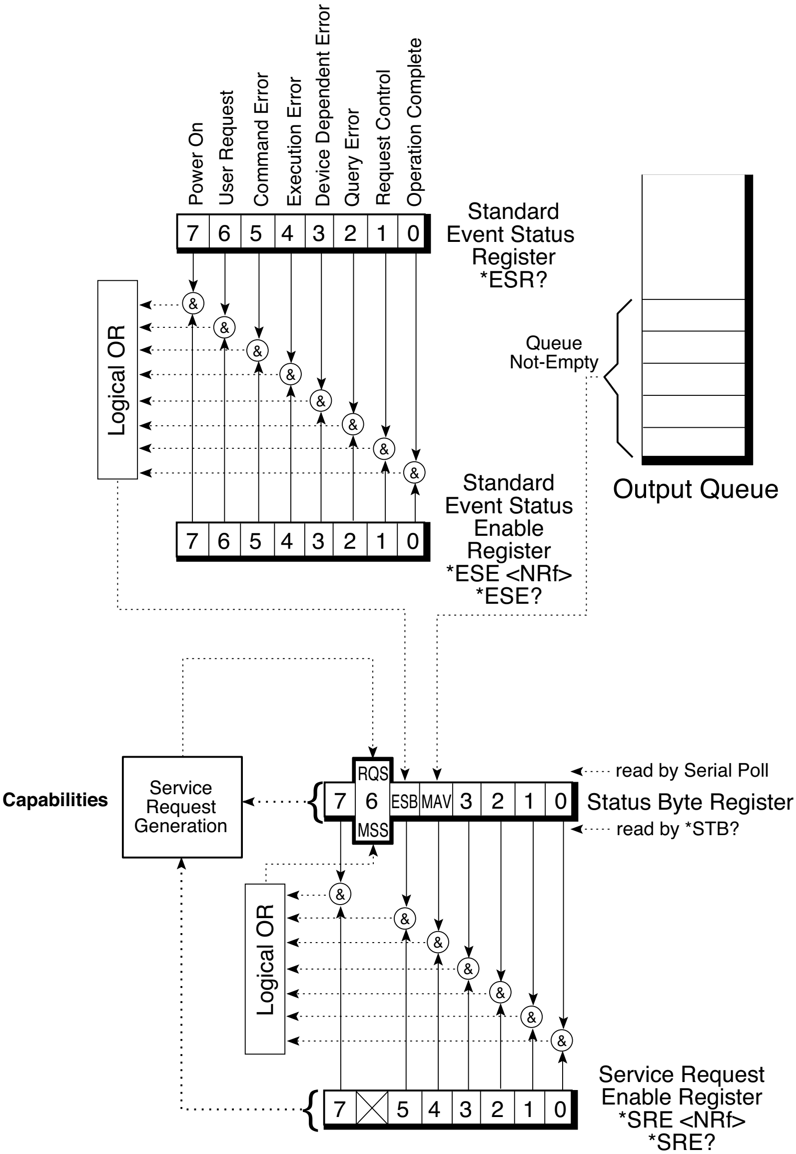


Figure A-3 488.2 Required Status Reporting Capabilities

TABLE A-2 IEEE 488.2 COMMON COMMANDS

Required common commands are:

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

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

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

Devices that support Device Trigger must support the following commands:

***TRG**	Trigger Command

Controllers must support the following command:

***PCB**	Pass Control Back Command

Devices that save and restore settings support the following commands:

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

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

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

A

A1.2.3 IEEE 488.2 Common Commands

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

A1.2.4 IEEE 488.2 Differences From IEEE 488.1

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

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

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

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

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

A1.3 SCPI COMMANDS

A1.3.1 Introduction

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

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

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

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

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

A1.3.2 Command Structure and Examples

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

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

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

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

Figure A-4 SCPI Command Examples

A

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

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

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

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

Figure A-5 Compound Command Examples

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

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

A1.3.3 Variables and Channel Lists

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

A

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

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

Figure A-6 Channel List Examples

A1.3.4 Error Reporting

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

A1.3.5 Additional Information

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

A

A2 SERIAL DATA COMMUNICATIONS BACKGROUND

A2.1 INTRODUCTION TO SERIAL COMMUNICATION

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

1. Data speed or baud rate
2. Data format
3. Transmission medium
4. Clocking method

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

110 - for old mechanical teletypes

300, 1200 - for low speed devices an older modems

9600 to 38400 baud for high speed devices and newer modems

Data format refers to the method or pattern the transmitter uses to send the data word or character as a series of bits so that the receiver will know how to recognize the pattern and reassemble the bits back into the original data word. The most common method and the one used in the 2303, is called asynchronous transmission because each character is sent one at a time with an undetermined amount of time between

A2

characters. Each asynchronous character has a low going start bit, a number of data bits, an optional parity bit and 1 or 2 high stop bits. The transmitter automatically extends the stop bit when it has no more characters to transmit. The receiver uses the start bit to resynchronize its clock with the data at the start of each character as shown in Figure A-7.

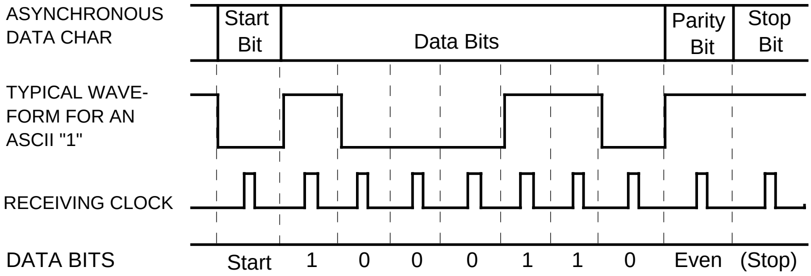


Figure A-7 Asynchronous Data Character Waveforms

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

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

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

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

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

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

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

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

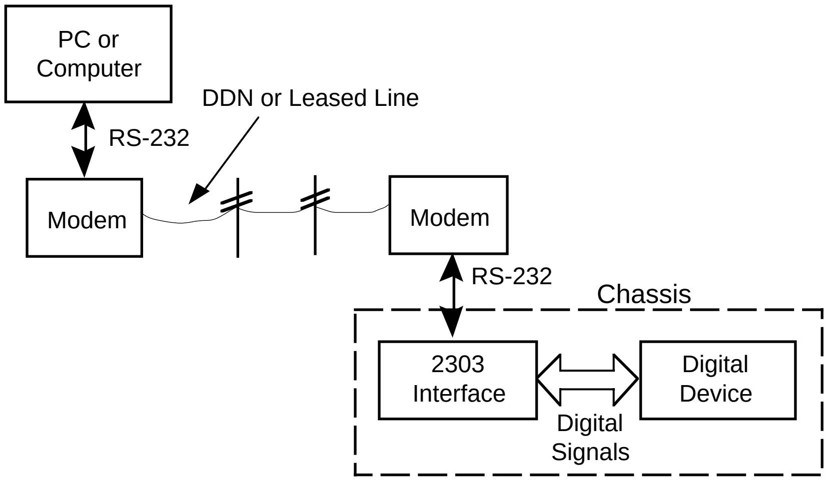


Figure A-8 Long Distance Communication using Modems

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

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

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

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

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

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

A2

A2.2 RS-232 STANDARD

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

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

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

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

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

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

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

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

A2.3 RS-422 AND RS-485 STANDARDS

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

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

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

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

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

A2.4 RS-530 PINOUTS

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

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

A2.5 SERIAL INTERFACE PROBLEMS

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

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

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

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

   i.e.,*!1-

A2

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

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

A2

A3 GPIB CONNECTOR/SWITCH BOARDASSEMBLIES

A3.1 BOARD DESCRIPTIONS

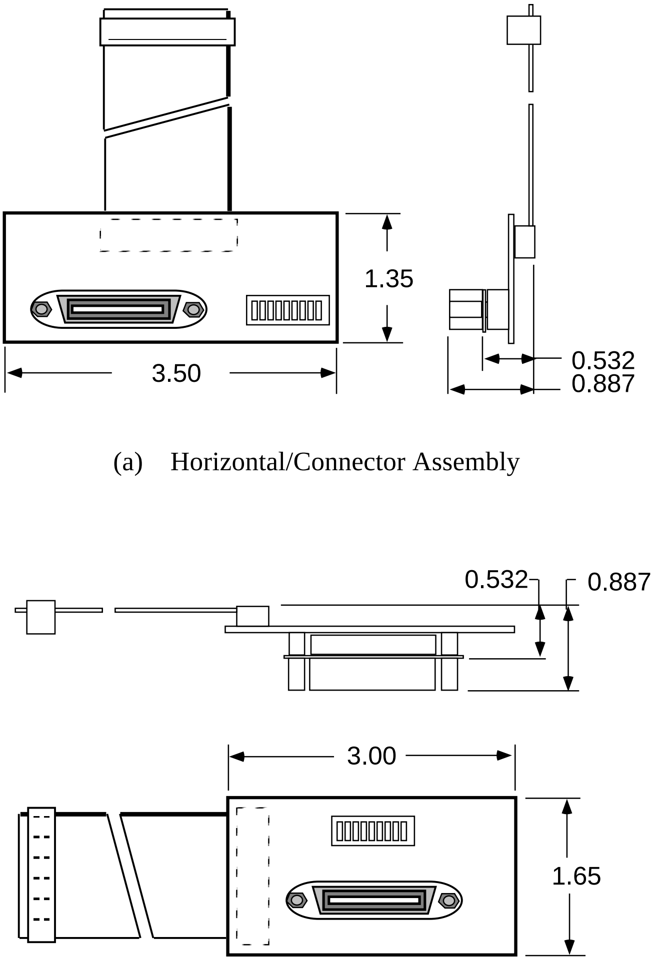
The GPIB Connector/Switch Board Assemblies are small printed circuit boards that provide a convenient way to mount an IEEE-488 Connector and an Address Switch on the rear of the host unit. They connect to the 4803 with a flat ribbon cable that plugs into the GPIB/Address header (J2).

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

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

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

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



(a) Horizontal/Connector Assembly

(b) Vertical Connector/Switch Assembly

A3

Figure A-9 GPIB Connector/Switch Board Assemblies

A3.2 BOARD INSTALLATION

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

1. Select the appropriate cutout from Figure A-11

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

3. Machine the cutouts.

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

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

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

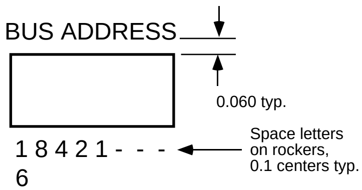
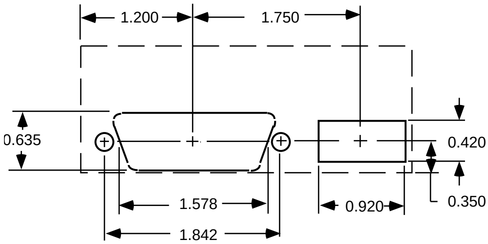
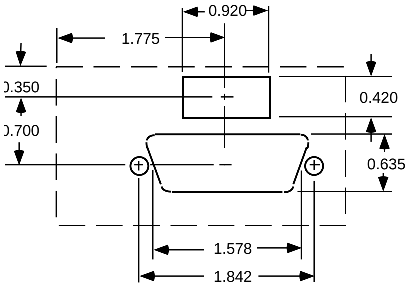


Figure A-10 Switch Silkscreen Detail



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

(a) Horizontal Connector/Switch Assembly Mounting Dimensions



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

(b) Vertical Connector/Switch Assembly Mounting Dimensions

Figure A-11 GPIB Connector/Switch Board Cutouts

Index

I

C

D

I

E

F

G

H

I

I

L

M

N

I

I

T

V

I