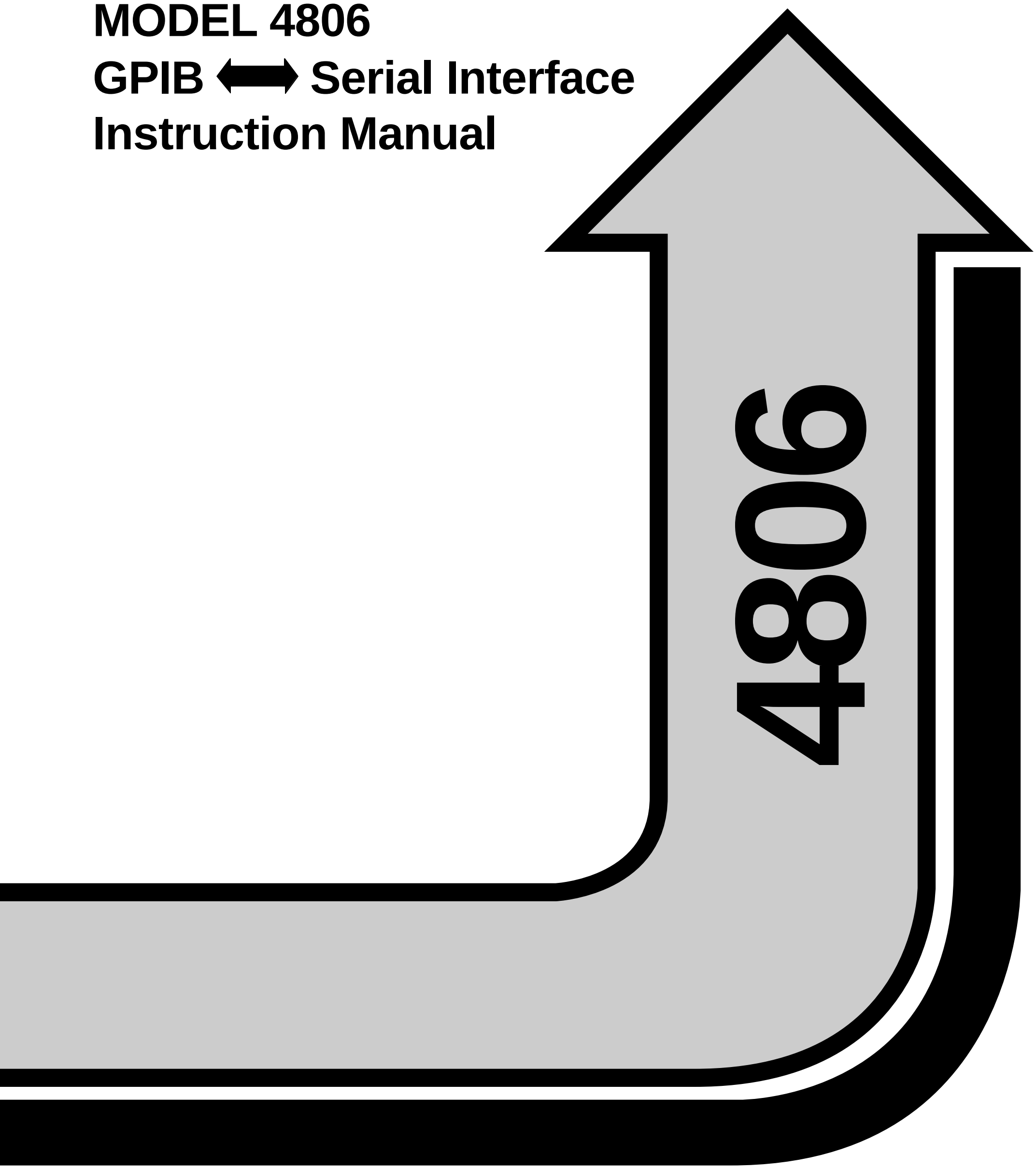
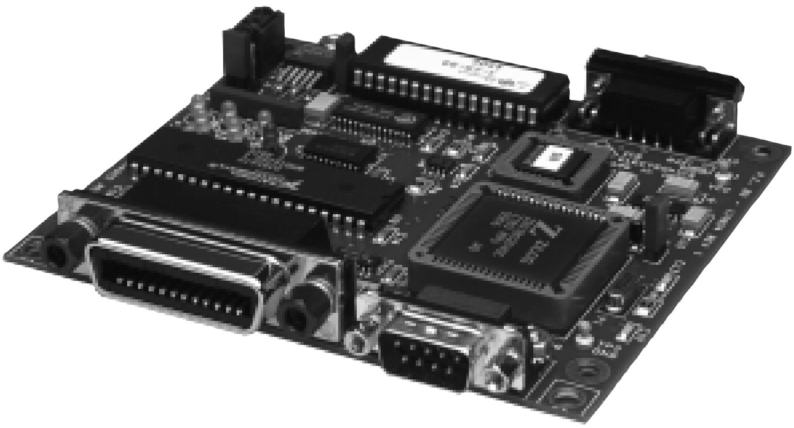
ICS
ELECTRONICS
a division of Systems West Inc.

MODEL 4806
GPIB ⟷ Serial Interface
Instruction Manual



MODEL 4806
GPIB ⟺ Serial Interface
Instruction Manual



ICS ELECTRONICS
a division of Systems West Inc.
473 Los Coches Street
Milpitas, CA 95035
Phone (408) 263-5500, Fax (408) 263-5896
Web: http://www.icselect.com

Publication Number 120153
November1999 Edition Rev X2

LIMITED WARRANTY

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

CERTIFICATION

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

EMI/RFI WARNING

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. The Model 4806 has not been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of the FCC Rules and to comply with the EEC Standards EN 55022 and EN 50082-1, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference. The Model 4806 should be tested for RFI/EMI compliance as a component in the user's equipment and has been designed for FCC and CE compliance when properly installed in the host chassis.

TRADEMARKS

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

HP is a trademark of Hewlett-Packard Corporation, Palo Alto, CA
ICS is a trademark of Systems West Inc, Milpitas, CA

REVISIONS

Revision 0 reversed the serial connector types so the DE-9P connector is on the rear panel.

Contents

General Information

1.1 INTRODUCTION

This section provides the specifications for ICS's Model 4806 GPIB to Serial Interface. The Model 4806 is a PC board product designed for placement inside another piece of equipment.

1.2 DESCRIPTION

The Model 4806 GPIB to Serial Interface is an IEEE-488.2 device that provides a GPIB and serial interface to a device with a single RS-232 serial interface. The 4806's GPIB interface provides all of the functions normally found in an IEEE-488.2 device including an IEEE-488.2 Status Reporting Structure and responses to all of the IEEE-488.2 Common Commands. In addition, the 4806 contains a SCPI parser for setting the serial interface configuration, for setting the GPIB address and configuring the SCPI Status Registers. The configuration is saved in the 4806's flash memory. The 4806's serial interface extends the device's RS-232 interface to the rear panel to give the end user a choice of using either interface to communicate to the internal serial device.

The 4806 has three modes of operation. When used with devices like transducers that periodically output a serial message, the 4806 saves the last message and outputs it on the GPIB bus when queried. When used with standard (dumb) serial devices, the 4806 transparently passes data strings onto the serial device and returns any responses

1

from the serial device to the GPIB controller. When used with a smart device like a microcontroller or an embedded computer, the 4806 transparently transfers data to the device and always expects a response from the device. In the smart mode, the serial device can set/reset bits in the 4806's Status Structure registers, query/set the 4806's GPIB address and query/use the 4806's local/remote state messages to control the serial device's control panel or operation.

The 4806 uses only one GPIB primary address for its own commands and for the data messages to the serial device. The 4806 does this by withholding all commands starting with the reserved SCPI keywords (SYSTem, STATus, CALibrate and DIAGnostic) and all IEEE-488.2 Common Commands from going to the serial device. The 4806 provides the appropriate responses to the GPIB controller for its SCPI and 488.2 commands and queries. All other GPIB messages are passed transparently onto the serial device. The exception to this rule is that the 4806 passes copies of IEEE-488.2 Common Commands on to a smart serial device.

The Model 4806 is fabricated as a small PC assembly with right-angle connectors for its GPIB and serial signals. The 4806 is designed to be mounted to the rear panel of the host chassis so the GPIB and serial connectors can protrude through the rear panel. The 4806 contains a low dropout regulator and runs off of 5 Vdc to 12 Vdc power. The 4806 also contains diagnostic LEDs which display the 4806's selftest status, error conditions and GPIB address at power turn-on time.

1.3 MODEL 4806 SPECIFICATIONS

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

4806 - X — General Model Number

Option Codes

-1	Special Crystal
-7	Special Program
-8	Custom Hardware Modification

Base Model Number

1.4 OPERATIONAL MODES

The 4806 has three operating modes to match the 4806's IEEE-488.2 capabilities to the serial device. The modes are: asynchronous, standard and smart.

1.4.1 Asynchronous Mode

The asynchronous mode is for serial devices that periodically send serial messages to the 4806. The 4806 saves the last message and outputs it when queried. The 4806 passes all GPIB messages onto the serial device except for commands that start with the reserved SCPI keywords and for all IEEE-488.2 Common Commands.

1.4.2 Standard Mode

The 4806 passes all GPIB messages onto the serial device except for commands that start with the reserved SCPI keywords and for all IEEE-488.2 Common Commands. Any responses from the serial device are saved and outputted on the GPIB bus when the 4806 is next addressed to talk. Standard mode devices are not required to respond to each GPIB message but if they do respond, they should do so within the timeout period.

1.4.3 Smart Mode

The 4806 passes all GPIB messages onto the serial device except for commands that start with the reserved SCPI keywords. The 4806 responds to all of the IEEE-488.2 Common Commands but passes a copy of the IEEE-488.2 Common Command onto the smart device in case it needs to react to the command. The smart device is expected to respond to all messages except for the copies of the IEEE-488.2 Common Commands. The smart device can set and clear bits in the 4806's Status Registers, query or change the GPIB address and IDN message and query the 4806's local/remote state to operate its front panel controls.

1.5 IEEE 488 INTERFACE

1.5.1 488.1 Capabilities

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

SH1, AH1, T6, L3, SR1, PP0, DC1, RL1, DT0, C0 and E1/E2 drivers.

1.5.2 Address Ranges

Primary addresses 0 - 30. Address saved in internal flash memory.

1.5.3 Data Transfer Rate

20,000 bytes per second

1.5.4 488.2 Common Commands

Standard 4806s conform to IEEE STD 488.2-1987. When addressed to listen in the command mode, the unit responds to the following 488.2 commands:

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

1.5.5 Terminators

The 4806 accepts GPIB messages that are terminated with a linefeed and/or with EOI asserted on the last character. If the GPIB messages are terminated with a carriage return-linefeed sequence, the 4806 will ignore the carriage return character.

The 4806 terminates all responses with a linefeed and asserts EOI on the last character. Messages from the serial device are not altered and EOI is asserted on the last character of the message from the serial device.

1

1.5.6 SCPI Parser

Standard 4806s include a SCPI parser that complies with the SCPI Standard Version 1994.0. The 4806 responds to any command that starts with the long or short from of the following keywords: SYSTem:, STATus:, CALibrate: and DIAGnostic:. These keywords are reserved and should not be used as the first word of a command that is to be passed to the serial device.

1.5.7 SCPI Status Structure

The 4806 provides a Status Reporting Structure with the following register sets:

ESR	488.2 Standard Event Status Register and Enable Registers.
STB	488.2 Status Byte Register and Enable Registers.
Operational	SCPI Operational Condition, Transition, Event and Enable Registers.
Questionable	SCPI Questionable Condition, Transition, Event and Enable Registers.

1.6 SERIAL INTERFACE

The 4806 has two serial connectors with RS-232 signals. The rear panel connector is a DE-9P connector with the signals arranged as a DTE type interface. The internal connector for interfacing to the serial device is a DE-9S connector with DCE signal arrangement. The serial output from the GPIB bus is ORed with the serial input from the rear panel to produce the output signal to the serial device. The output of the serial device drives both the rear panel connector and the GPIB bus. Signal pinouts are listed in Table 1-1.

1.6.1 Signal Levels

Transmit	+6 to +10 Vdc = Logic "0" or On
Levels	-6 to -10 Vdc = Logic "1" or Off
Receive	± 1.5 Vdc minimum, ± 25 Vdc Maximum

1.6.2 Baud Rates:

Any rate from 300 to 115,200 baud. Parser selects closest rate to specified rate when a nonstandard rate entered. Standard rates are: 300, 600, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 57600, 76800, 92160 and 115200 baud.

1.6.3 Data Character Formats:

Type	Asynchronous character
Data bits	7 or 8 data bits per character
Parity	Odd, even, none
Stop bits	1 or 2 stop bits per character

1.6.4 Data Buffers:

GPIB to Serial	2 K bytes
Serial to GPIB	2 K bytes

1.6.5 Signal Pin Assignments

1

TABLE 1-1 RS-232 SIGNAL ASSIGNMENTS

Pin #	External Connector DE-9P	Direction Thru 4806	Internal Connector DE-9S
1	DCD	+V→	DCD
2	TxD	←	TxD
3	RxD	→	RxD
4	DTR	← +V	–
5	Gnd		Gnd
6	DSR	+V→	DSR
7	RTS	← +V	RTS
8	CTS		CTS
9	nc		nc

1.7 PROGRAMMABLE FUNCTIONS

The 4806 uses IEEE 488.2 and SCPI commands to change its operating configuration and GPIB address. Table 1-2 lists the 4806's programmable functions and their factory default settings.

TABLE 1-2 4806 DEFAULT SETTINGS

Command	Functions	Factory Defaults
:MODE	Selects operation	Standard #
:ADDRess	Sets GPIB bus address	4
:BAUD	Sets transmit/receive baud rate	9600 #
:PARity	Sets parity type	NONE #
:BITs	Sets number of data bits per character	8 #
:SBITs	Sets number of stop bits/per character	1 #
:EOI	Enables EOI on last character of the received serial message.	ON #
:TIMEout	Sets wait for serial device response time	25 ms
***ESE**	Enables Standard Event Status Register bits	0
***SRE**	Enables Status Byte Register bits	0
***IDN?**	IDN message	ICS

1

1.8 INDICATORS

The 4806s have six LEDs that display the following conditions:

PWR	-	Indicates power on
RDY	-	Indicates unit has passed its selftest
TALK	-	Indicates unit has recognized its talk address
LSTN	-	Indicates unit has recognized its listen address
SRQ	-	On when 4806 asserts SRQ
ERR	-	On when 4806 has detected an error. On at power turn-on time if the 4806 is not calibrated.

When the 4806 is turned on, it turns the PWR LED on and performs an internal selftest which takes about 0.5 seconds. At the end of the selftest, the 4806 displays its current GPIB address by blinking the front panel LEDs for one-half second before turning on the RDY LED. The LED bit weights are:

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

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

1.9 4806 PHYSICAL

Size	-	4.00"L x 4.50"W x 0.7"H (10.16 cmL x 11.43 cmW x 1.78 cmH) (See Figure 1-1)
Weight	-	8 oz (0.22 kg)
Temperature	-	Operating -10° C to +55° C Storage -20° C to +70° C
Humidity	-	0-90% RH without condensation
Shock/Vibration	-	Normal handling only
Construction	-	Flame-retardant printed circuit board. Connector shells grounded to chassis supports and mounting brackets.
Power	-	5 to 12 Vdc @ 400 mA
Connectors	-	IEEE 488 Interface Amphenol 57-20240 with metric studs External RS-232 Interface Cinch DE-9S with lock studs Internal RS-232 Interface Cinch DE-9P with lock studs

1.10 4806 CERTIFICATIONS OR APPROVALS

EMI/RFI	Designed to meets limits for part 15, Class A of US FCC Docket 20780 and EEC Standards EN 55022 and 50082-1 when enclosed in a suitable chassis.
UL/CSA/VDE	Intrinsically safe because all voltages are below 12 Vdc.

Note: Refer to Figure 2-1 for Panel Mounting Dimensions

Figure 1-1 4806 Outline Dimensions

1.11 ACCESSORIES

1.11.1 Included Accessories

120153	4806 Instruction Manual
123045	Minibox GPIB Configuration Disk for PC and PC compatible computers.

1.11.2 Optional Accessories

120153	4806 Instruction Manual
123045	Minibox GPIB Configuration Disk
104705-104700-15	GPIB Bus Cables, multishielded. Lengths from 0.5 meters to 15 meters.

Installation

2.1 UNPACKING

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

2.2 SHIPMENT VERIFICATION

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

(1) Model 4806 GPIB<->Serial Interface Board
(1) 4806 Instruction Manual
(1) Minibox Configuration Program Disk

Board only 4806 orders (Part# 114516) do not include manuals or configuration disks unless ordered separately.

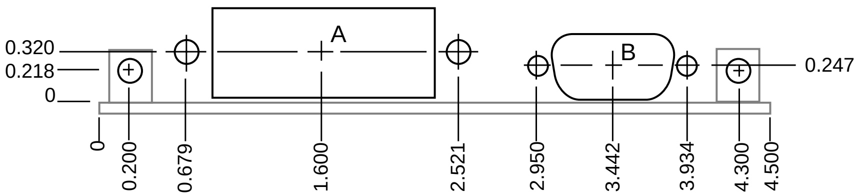
Take a moment to check ICS's website at http://www.icselect.com for any manual errata and for the latest configuration programs.

2.3 4806 INSTALLATION GUIDE

The following steps should be used as a guide to the 4806 installation.

1. Review Section 2.9 to select and/or design the serial interface cables.

2. Select a convenient location to mount the 4806. The 4806 mounts flush against the host chassis rear panel. Use the cutouts and mounting dimensions shown in Figure 2-1 for the connector cutouts. Cutout 'A' is for the IEEE-488/GPIB connector. Cutout 'B' is for the DE-9 serial connector. Do not mount the 4806 directly over a heat producing surface. Provide a 0.1 inch (2.5 mm) clearance underneath the 4806 or use an insulator if the 4806 is being mounted on a metal surface.



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

Figure 2-1 4806 Rear Panel Cutouts

3. To minimize EMI/RFI and to maximize electrical immunity, use the mounting blocks to fasten the 4806 to the rear panel of the host chassis. The 4806 connector shells fit against the rear panel. Use the recommended cutouts in Figure 2-1 to overlap the connectors.

4. Use a twisted pair of #24 wires to connect the 4806's power terminals to the host's power supply. Use either + 5 Vdc regulated power or unregulated 5.5 to 12 Vdc power. Connect the 4806 directly to the power supply to avoid noise problems. Set jumper W2 to REG if you are using unregulated power.

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

6. Review the factory settings in Table 1-1 to determine if your unit needs to be reconfigured before it can send data to the serial device. If the 4806 needs to be reconfigured, follow the instructions in Section 2.5 or 2.6 to change its configuration.

7. Sent a test GPIB message to the serial device that can get a response from the serial device. Read the response to confirm that it is the expected response.

2.4 CONFIGURATION PROGRAMS

The 4806's configuration is stored in nonvolatile flash memory and can be changed from any GPIB bus controller with SCPI commands. Use a configuration program or put the commands in your program. If your GPIB controller is a PC with DOS or Windows 3.1, and has one of the GPIB controller cards listed below, you can use the programs on the supplied Minibox Configuration Disk to walk you through a menu driven configuration procedure. Follow the instructions in Section 2.5 to install and use configuration programs.

2

Program	Supported GPIB Cards
mconfig.exe	ICS 488-PC2 Card National Instruments GPIB-PC2a Card Most NEC 7210 compatible GPIB Cards (Set cards to address 2E1, PC2a mode and to 7210 emulation)
niconf.exe	National Instruments AT-GPIB Card (Set to address 2C0H)
hpconf.exe	Hewlett-Packard HP-IB Card (Set to address DC000)

If you are using a PC with Windows 95/98 or NT and have one of the following cards installed in your computer, you can download the niconfig program from ICS's web site at http://www.icselect.com and use it to configure the unit. Niconf_z.exe is a self-exploding zip file that installs a configuration program that makes National Instrument type calls to control the GPIB bus.

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

2.5 USING THE CONFIGURATION PROGRAMS

The Minibox configuration programs walk the user through menu driven programs to configure the 4806's power-on settings. The configuration programs on the supplied Configuration Disk run on any IBM type PC or compatible clone with DOS 6.0 or Windows 3.1 operating system. A Windows 95/98/NT version of the configuration program can be downloaded from ICS's web site at http://www.icselect.com.

2.5.1 Installing the DOS Configuration Program from Disks

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

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

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

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

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

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

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

2

2.5.2 Installing a Downloaded Configuration Program

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

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

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

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

2.5.3 Running the Configuration Program

The configuration programs support the standard configurable items. Special 4806 settings such as the user's IDN message will have to be entered with a live keyboard program or as part of the user's program. See section 3.8.6.

1. Connect the 4806 to the GPIB controller card in the PC as shown in Figure 2-2. Connect a DC power supply to the power terminals on the 4806 card. See Figure 2-3. Set the power supply to 5 ± 0.25 Vdc.

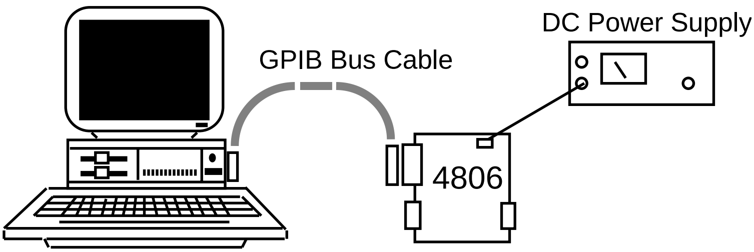


Figure 2-2 4806 Configuration Setup

2. Apply power to the 4806. After 0.5 seconds, the unit should blink its GPIB bus address on the LEDs. The selftest ends with the PWR and RDY LEDs both on and the other LEDs off.

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

   > c:\new_directory\MCONFIG <return>

2

4. Product Selection

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

   e.g. 4806 <return> 'selects Model 4806

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

5. GPIB Address

   The program branches to the selected product menu and asks for the unit's current GPIB address. Enter a one or two digit value ; i.e., 4, 04, 10. The factory default setting is 4. If you do not know the unit's GPIB address, turn the unit off and back on. The unit will blink its GPIB address on the front panel LEDs at power turn-on. Add the bit weights to get the GPIB address. LED locations are shown in Figure 2-3.

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

2

6. Configuration Choices

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

7. Saving the New Settings

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

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

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

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

8. Configuring other units

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

2.6 CONFIGURING FROM OTHER CONTROLLERS

2.6.1 General Instructions

The 4806 can be configured from any GPIB bus controller by using the following procedure. The following example commands are shown in HP BASIC for easily conversion to another language.

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

2. Determine the unit's GPIB address:

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

   b) For other units, turn the unit off and back on. At the end of its self test, the unit blinks its GPIB address on the LEDs. See Figure 2-3 for LED locations.

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

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

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

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

2

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

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

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

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

```
OUTPUT 704, "*SAV 0", END
```

2.7 POWER CONNECTIONS

Power is applied to the 4806 at P1 which is a two screw terminal block. See Figure 2-3 for P1 polarity. Use either 5 Vdc regulated power or 5.25 to 12 Vdc unregulated power.

Set jumper W2 to **REG** when using the **internal regulator**. Set jumper W2 to **P1** when using **+5Vdc** directly from the terminal block.

2.8 GPIB CONNECTIONS

The 4806 has a standard 24-pin right-angle IEEE-488 connector. The IEEE-488 connector mates with the standard IEEE 488/GPIB bus cables. Signal-pin assignments for the standard IEEE-488 connector are shown in Figure A-2 in the Appendix. Use a straight-in GPIB cable, ICS P/N 1057xx if you want to access both the GPIB and the serial connectors at the same time.

2.9 SERIAL INTERFACE CONNECTIONS

The 4806 has two right-angle 9-pin connectors for its RS-232 interfaces. Connector J2 is the rear panel connector for external signals. Connector J3 is the internal connector for connecting to the serial device. Signal pin assignments are shown in Table 2-1.

2.9.1 Rear Panel Connector J2

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

2.9.2 Internal Connector J3

Connector J3 is a DE-9S connector with DCE type signal assignments. Signal RxD on pin 3 is the output to the internal serial device. Signal TxD on pin 2 is driven by the internal serial device. DCD and DSR are pulled up to an 'ON' state and RTS is jumpered back to CTS.

TABLE 2-1 RS-232 SIGNAL ASSIGNMENTS

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

Notes: 1. Direction In is to the 4806 board. 2. - is an open pin
3. +V is + 9 Vdc.
4. Internal pins 7 and 8 are jumpered together.

2.10 JUMPER SETTINGS

The 4806 has three jumpers with the functions listed in Table 2-2.

TABLE 2-2 4806 JUMPER SETTINGS

Jumper	Functions	Factory Setting
W1	Write Enable - Must be in place to write to or save data in the 4806's flash memory. Blocks all writes when removed	Installed
W2	Power Selection - Selects input power. P1 position selects the terminal block and assumes a 5 Vdc regulated power source. REG selects the regulator output and is the position for unregulated 5.5 to 12 Vdc power.	Jumper in neutral
W3	Default - Returns the unit to its factory default settings when in place at power turn-on time. Leave out for normal operation.	Omitted

2

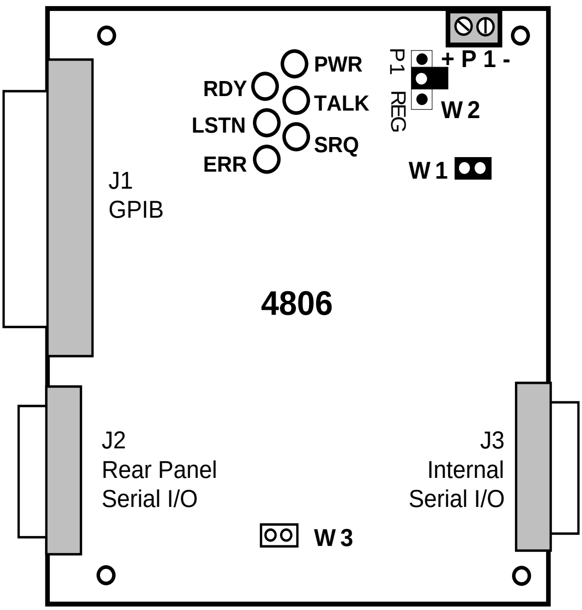


Figure 2-3 4806 Jumper Locations

3

Operation

3.1 INTRODUCTION

This section describes how the 4806 operates and converts GPIB data into serial messages and serial responses back to GPIB data.

3

3.2 OPERATION

The 4806 is an IEEE-488.2 compatible, GPIB-to-serial interface that adds an IEEE-488.2 interface to a serial device while maintaining the device's serial interface. The 4806 is designed so it be mounted perpendicular to the rear panel of a chassis or parallel to a panel or other PC board. When mounted perpendicular to the rear panel, the 4806's GPIB connector and a serial connector protrude through the rear panel. An internal serial connector provides a direct connection to the serial device.

The 4806 uses a single primary address between 0 and 30 to receive all GPIB messages. The 4806 checks each message and blocks all messages, that are IEEE-488.2 Common Commands or that start with the reserved SCPI keywords, from going to the serial device. All other GPIB messages are transparently converted into serial messages and transmitted to the serial device. The reserved keywords are the long or short forms of: SYSTem, STATus, CALibrate and DIAGnostic. The reserved words can be used in messages to the serial device as long as they are not the first word in the message.

The 4806's SCPI parser accepts all IEEE-488.2 Common Commands and the SCPI commands that start with the four reserved keywords. The 4806 responds to all of the IEEE-488.2 Common Commands listed in Section 1, provides the appropriate responses and provides a Status Reporting Structure for the serial device. The reserved SCPI commands are used to setup and configure the 4806's serial interface, to set the its GPIB address, to set its operating mode and to configure and query the SCPI registers in the 4806's Status Structure. All configuration parameters are saved in the 4806's flash memory.

The 4806 operates with three types of serial devices: Asynchronous, Standard or Smart. The 4806's default selection is the Standard device. The device type selection is made by setting the MODE parameter with the SYST:MODE command. The current mode can be made the power-on default setting by saving it with the *SAV 0 command.

In the Asynchronous mode, the 4806 works with serial devices such as transducers that output periodic serial messages. The 4806 saves the most recent message and outputs it on the GPIB bus when queried with the SYST:COMM:SER:REC:DATA? query. In the Asynchronous mode, the 4806's Status Reporting Structure can be set to generate a SRQ upon receipt of a serial message. The 4806 passes all but the blocked GPIB messages onto the serial device.

In the Standard mode, the 4806 works with serial devices that accept messages and send back responses when queried. The 4806 passes all but the blocked GPIB messages onto the serial device. Any responses from the serial device are stored in the 4806's Rx buffer and are outputted to the GPIB bus when the 4806 is next addressed to talk. Standard serial devices are not required to respond to every serial message but when they do respond, the response must be received within the preset timeout period. If the response is received after the timeout period has elapsed, the response will not be saved. The 4806 will not accept another GPIB message while waiting for the response or until the timeout period has elapsed.

In the Smart mode, the 4806 works with programmable devices such as microprocessors or embedded computers that are capable of interacting with the 4806. The 4806 passes all GPIB messages onto the serial device except for the GPIB messages that start with the reserved SCPI keywords. Copies of the IEEE-488.2 Common Commands are passed onto the serial device in case it needs to react any of the 488.2 Common Commands. The smart device is expected to respond to every serial message it receives, except for the copies of the IEEE-488.2 Common Commands, with a response or with the '@@@OK' acknowledgment message. A response or acknowledgment message must be received before the timeout period has elapsed or the 4806 will set the receive timeout error bit in its ESR register. Response messages are saved and outputted to the GPIB bus when the 4806 is next addressed to talk.

3

Smart serial devices can communicate with the 4806 by sending the 4806 serial messages that start with an @@@ sequence. These messages let the serial device set/reset bits in the 4806's ESR, Questionable and Operational Registers, set or query the 4806's GPIB address, set the 4806's IDN message, query the 4806's Local/Remote state and request the 4806 to go to the local state. Messages from the serial device that start with @@@ are reserved for serial device to 4806 communication and are not outputted to the GPIB bus.

3.3 ADDRESSING THE UNITS

3.3.1 4806 Internal GPIB Address

The 4806 can be set to any unused GPIB primary address between 0 and 30. The Bus Controller will use the GPIB address to address the unit as a talker or as a listener. Bus addresses of 0 and 21 are not recommend as these addresses are customarily used by Bus Controllers as their own address.

3

The 4806's internal GPIB address can be set or queried with the SCPI SYST:COMM:GPIB:ADDR command. The GPIB address change takes affect when the command is executed so any subsequent commands will need to be sent to the 4806 at its new address. Use the IEEE-488.2 *SAV 0 command to save the new address value in the 4806's flash memory.

If you have forgotten the unit's GPIB address, momentarily turn the unit off and back on. At the end of the self test, the unit will blink its GPIB address on its diagnostic LEDs using the following bit weights. See Figure 2-3 for the LED locations.

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

In the Smart mode, the serial device may query the 4806's GPIB address with the @@@ADDR? query and change the 4806's address with the @@@ADDR command.

3.4 488.2 STATUS REPORTING STRUCTURE

The 4806 includes the expanded IEEE-488.2 status reporting structure shown in Figure 3-1. The expanded status reporting structure conforms to the SCPI 1994.0 Specification and builds on the IEEE 488.2 Standard Status structure with the addition of the Questionable and Operational register sets. The 488.2 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 4806. The Questionable and Operation register sets are controlled and queried with SCPI STATus commands. In the Smart mode, the serial device can set and reset bits in the 4806's status reporting structure.

3

As shown in Figure 3-1, IEEE 488.2 SRQ generation is a multilevel function and is determined by the occurrence of an event bit that has its corresponding enable bit set to '1'. The enabled register outputs are summarized in the Status Byte Register which generates the Service Request and pulls the SRQ line low. SRQs are used to signal the bus controller that an event has occurred and/or that the 4806 needs service. There are four major sources of SRQs, each of which is summarized in a bit in the Status Byte Register. Three of the sources are event registers with their own enabling bits and the fourth is the Output Queue. The event registers and the Output Queue are cleared when read or by the *CLS command.

3.4.1 Event Registers

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

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

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

3

Figure 3-1 4806 Status Reporting Structure

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

3.4.2 Event Status Register

3

The Event Status Register reports events that are common to all 488.2 devices. This includes events such as self test errors, command errors, execution errors, power on and operation complete. The Power-on event occurs at power turn-on and can be used to signal a power off-on occurrence. In the 4806, Bit 6 of the Event Status Register is used to report serial device Response Timeout Errors. The 488.2 Operation Complete event may be used by the smart serial device to signal that it has completed its assigned task.

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

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

In the 4806, Bit 3 of the Event Status Register is used to report a problem with the user settings in the Flash memory. This bit normally means that the CAL DATE is missing or has been reset and that all settings have been restored to the factory defaults. Check and change the settings as necessary for your installation. Use the CAL:DATE command to enter today's date. The LEDs will momentarily flash as the new date and settings are saved.

3.4.3 Questionable Registers

The Questionable Register is not used in the 4806. Any of its bits may be used by a smart device to report its status to the GPIB controller. The Questionable Transition Register filters the Condition Register bits and passes only the enabled state changes to the Questionable Event Register. The Questionable Event Register bits becomes true (1) when the positive transition bit is enabled and the associated condition register bit makes a 0 to 1 transition. When both transitions are selected for the same bit, the corresponding Questionable Event Register bit sets whenever the digital input changes state. The Questionable Event Register is cleared when it is read.

3

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

The following example sets the Questionable Event register to capture a positive transition on bits 12 and 13. The decimal value for bit 12 is 4096 and the decimal value for bit 13 is 8192.

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

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

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

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

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

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

STAT:QUES:COND?	'reads the questionable inputs

3.4.4 Operation Registers

The 488.2 Operation Registers lets the user read serial buffer and other status conditions and report any changes in these conditions. The Operation Registers operate in a similar fashion to the Questionable Registers described in paragraph 3.4.3.

3

The following commands demonstrate some possibilities of the Operation Registers:

STAT:OPER:PTR 1 'enables bit 0 to set when a serial message is received
STAT:OPER:ENAB 1 'enables Event bit 0
STAT:OPER:COND? 'quires the Operation Con dition Register

3.4.5 Output Queue

The Output Queue is used by the 4806 to send IEEE 488.2 messages back to the bus controller. These messages are responses to 488.2 and SCPI queries sent to the unit by the bus controller. The Output Queue reports a '1' in bit 4 of the Status Byte Register when it contains a message(s) to be read by the bus controller. Reading the contents of the Output Queue clears its summary bit. The Output Queue is read by addressing the 4806 to talk at its GPIB address. If the Output Queue is not read before sending another query, its contents will be lost and an error reported.

3.4.6 Status Byte Register

The 4806 generates a service request (SRQ) whenever any of the enabled bits in the Status Byte Register become true and the 4806 is not addressed as a talker. The Status Byte Register may be read by a Serial Poll or with the ***STB?** query. A Serial Poll resets the RQS bit; the ***STB?** query does not change the bit. The Status Byte Register is enabled by setting the corresponding bits in the Service Request Enable Register with the ***SRE** command. e.g.

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

3

3.4.7 Saving the Enable and Transition Register Values

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

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

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

3.4.8 488.2 Differences from 488.1 Devices

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

3.4.9 Smart Device Rules

The Smart serial device can set or reset any of the bits in the 4806's ESR, Questionable or Operational registers. However, the Smart device should be restricted to setting only those bits that are unused by the 4806 to avoid potential conflicts and IEEE-488.2 violations. The recommended bits are the unused bits in the Questionable Condition Register and the OPC bit in the ESR Register.

Bits set in the 4806's ESR Register stay set until the register is read or cleared with the *CLS command. Bits ORed into the Questionable or Operational Condition Registers stay set until ANDed out by the Smart device. Bits to be ORed in or ANDed out are expressed as high true.

3.5 488.2 CONFORMANCE INFORMATION

The IEEE 488.2 Standard mandated a list of commands that are common to all IEEE 488.2 compatible devices. The 4806 responds to all of the mandated common commands and to some optional commands defined in IEEE-488.2. Table 3-1 lists the IEEE-488.2 commands that apply to this unit, and describes the affect they have on the 4806 and its status reporting structure.

TABLE 3-1 IEEE-488.2 COMMON COMMANDS

COMMAND	NAME	DESCRIPTION
*** CLS**	Clear Status	Clears all event registers summarized in the status byte, except for "Message Available," which is cleared only if *CLS is the first message in the command line.
***ESE <value>**	Event Status Enable	Sets "Event Status Enable Register" to <value>, an integer between 0 and 255. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated. EXAMPLE: decimal 16 converts to binary 00010000. Sets bit 4 (EXE) in ESE to 1.
***ESE?**	Event Status Enable Query	4806 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	4806 returns the <value> of the "Event Status Register" and then clears it. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***IDN?**	Identification Query	4806 returns its identification code as four fields separated by commas. These fields are: manufacturer, model, six-digit serial number and version of firmware - e.g. **ICS Electronics, 4806, S/N 012123, Rev. 00.10 Ver 99.01.27**
***OPC**	Operation Complete Command	Causes the 4806 to generate the operation complete message in the Standard Event Status Register when all pending selected 4806 operations have been finished.
***OPC?**	Operation Complete Query	Places an ASCII character 1 into the 4806's Output Queue when all pending selected 4806 operations have been finished.

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

COMMAND	NAME	DESCRIPTION
***RCL 0**	Recall	Restores the state of 4806 from a copy stored in its E^2PROM by *SAV command.
***RST**	Reset	The 4806 restores its power-up state except that the state of IEEE-488 interface is unchanged, including: 1) instrument address, 2) Status Byte and, 3) Event Status Register.
***SAV 0**	Save	Saves current 4806 configuration in the E^2 PROM. *SAV 0 saves the current setting as the new power on setting.
***SRE <value>**	Service Request Enable	Sets the "Service Request Enable Register" to <value>, an integer between 0 and 255. The value of bit six is ignored because it is not used by the Service Request Enable Register. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register. If <value> is not between 0 and 255, an Execution Error is generated.
***SRE?**	Service Request Enable Query	4806 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	4806 returns the <value> of the "Status Byte" with bit six as the "Master Summary" bit. <value> is an integer whose binary equivalent corresponds to the state (1 or 0) of bits in the register.
***TST?**	Self-Test Query	Returns status of the last power turn-on self test. A zero response indicates no test failures. Other responses are listed in Table 5-2.
***WAI**	Wait-to-continue	Prevents the 4806 from executing any further commands or queries until the No-Operation-Pending flag is TRUE.

3

3.6 SCPI CONFORMANCE INFORMATION

The 4806 accepts SCPI commands and command extensions to configure its GPIB/Serial interfaces, to set the data formats and to transfer data. The SCPI commands conform to SCPI Standard 1994.0 and provide an industry standard, self-documenting form of code that makes it easy for the programmer to maintain the application program.

Table 3-2 shows the 4806's SCPI command tree. The command tree uses portions of the SCPI SYSTem, STATus, CALibrate and DIAGnostic subsystems. The 4806 and 4806 follow SCPI's hierarchal 'tree like' structure which starts with a root keyword and branches out to the final action keyword. Each command can be used as a query except where noted. The SCPI commands are **not** case sensitive. The portion of the command shown in capitals denotes the abbreviated form of the keyword. Either the abbreviated or whole keyword may be used when entering a complete command. Bracketed keywords are optional and may be omitted. There must be a space between the command and the parameter or channel list.

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

Table 3-3 lists the SCPI keywords and describes their functions in detail. Keywords other than those listed in the table or locked keywords will have no effect on the 4806's operation and a command error will be reported. Refer to Appendix A-1 for additional information about SCPI commands.

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

TABLE 3-2 4806 SCPI COMMAND TREE

Keyword	Parameter Form	Notes
SYSTem		**System Address**
:MODE	ASYNc | [STANdard] | SMARt	
:COMMunicate		
:SERial		
[:RECeive]		
:BAUD	<numeric value> [9600]	
:PARity		
[:TYPE]	EVEN | ODD | [NONE]	
:BITS	7 | [8]	
:SBITs	[1] | 2	
:EOI	0|1 or OFF|ON [0]	
:TIMEout	1-65,535 ms [25]	
:UPdate		
:GPIB		
:ADDRess	0 - 30 [4]	
:ERRor?	(0, "No error")	
:VERSion?	(1994.0)	
STATus		**Status Structure**
:OPERation		Status Inputs, WTG
[:EVENt]?	bit 1 and 5 active (0)	
:CONDition?	bit 1 and 5 active (0)	
:ENABle	bit 1 and 5 active (0)	
:ENABLE?		
:PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:QUEStionable		Modbus Error Bits
[:EVENt]?	bits 0-2, 12, 13 active (0)	
:CONDition?	bits 0-2, 12, 13 active (0)	
:ENABle	bits 0-2, 12, 13 active (0)	
:ENABLE?		
: PTRansition	0-#h7FFF [All 1s]	
:PTRansition?		
:NTRansition	0-#h7FFF [0]	
:NTRansition?		
:PRESet		

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

Keyword	Parameter Form	Notes & Short Form Commands
CALibrate		**Calibrate**
:IDN	string	
:DATe	mm/dd/yyyy	
:DEFault		
:LOCK	1(On)| 0(Off) [0]	
DIAGnostic		**Diagnostic**

3

Notes:

1. Parameter enclosed by [ ] - denotes factory default
2. Parameter enclosed by ( ) - denotes power on default
3. SCPI name ends with ? - denotes query only
4. Unless otherwise noted SCPI command is also a query
5. Keyword enclosed by [ ] - denotes optional use
6. Only a configuration command that has one of its parameters enclosed by [ ] can change its parameter setting and have this setting stored in the 4806's E^2ROM (with the *SAV command).
7. The format for a SCPI list is (@1,2, n) or (@ 1:n). There must be a space between the @ and the first number and parenthesis are required. A list of numbers is separated by commas or uses a colon to denote a range of numbers.
8. Numeric entries conform to IEEE-488.2 section 7.7.2.4 for decimal numeric parameters.
9. ASCII formatted data is a series of decimal values (0-255) for each byte separated by commas. e.g. 64, 132, 8
10. The CAL:DATe commands stores the CAL:IDN and CAL:DATe parameters in the 4806's E^2ROM.
11. The CAL:DEFault command resets the E^2ROM memory to it factory settings. Caution - All user settings will be overridden by this command.
12. Most parameters can be output in various numeric formats (radix). The parameters with decimal 0-255 value ranges may also be output as HEX using #h00-#hFF or Binary using #b00000000-#b11111111. Conversely, the parameters shown with HEX (#h) values can also be output in Decimal.

TABLE 3-3 SCPI COMMANDS AND QUERIES

Keyword	Default Value	Description
SYSTem	-	Starts System command branch.
:MODE	STAN	Sets operational mode to match the serial device connected to J3. See paragraph 3.2 for a description of the modes operation. Values are: ASYNc,STANdard and SMARt.
:COMMunicate	-	Identifies communication subsystem commands
:SERial		Controls Serial Interface settings
[RECeive]		Optional keyword
:DATA?		Reads buffered data string from Asynchronous serial devices.
:BAUD	9600	Sets serial baud rate. Vallues for the 4806 are 300 to 115200 baud.
:PARity	NONE	Sets serial parity. Values are EVEN, ODD or NONE.
[TYPE]		Optional parity keyword
:BITS	8	Sets number of data bits per character. Values are 7 | 8.
:SBITs	1	Sets minimum number of stop bits between characters. Values are 1 | 2.
:EOI	OFF	Enables EOI to be asserted when teh last character in the Rx buffer is talked onto the GPIB bus. Values are 1 and 0 or ON and OFF.
:TIMEout	25	Sets response message timeout period. Values are 1 to 65,535 ms. Do not set the tiomeout period to periods longer than required by the internal serial device as the 4806 cannot accept GPIB messages during the timeout period.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES
(CONTINUED)

Keyword	Default Value	Description
:UPdate	-	Updates the UART with the current serial parameters.
:GPIB	-	Controls GPIB (IEEE 488) port settings
ADDRess	04	Sets 4806's GPIB address. Values are 0 to 30. **Provide 70 ms delay after an address change before next command**
:ERRor?	0, "No error"	Requests next entry in 4806's error/event queue. Error messages are: 0, "no error" -100, "Command error" -200, "Execution error" -400, "Query error"
:VERSion?	1994.0	4806 returns the <value> of the applicable SCPI version number.
STATus	-	Starts Status Reporting Structure
:OPERational	-	Identifies Operational registers.
:QUEStionable	-	Identifies Questionable registers.
[:EVENt?]		Returns contents of the event register associated with the command.
:CONDition?		Returns contents of the condition register associated with the command.
:ENABle	0	Sets the enable mask which allows the true conditions in the associated event register to be reported in the summary bit.
:PTRansition	#h7FFF	Sets positive transition enable register. Value = 0 to #h7FFF in decimal or HEX.
:NTRansition	0	Sets the negative Transition register. Values = 0 to #h7FFF in decimal or HEX.

3

TABLE 3-3 SCPI COMMANDS AND QUERIES (CONTINUED)

Keyword	Default Value	Description
:PREset		Sets the selected Enable Register, PTR and NTR registers to their default values (0, #h7FFF and 0 respectively) so the 4806 detects a positive changes
CALibrate		Starts calibrate branch
:IDN <string>		Sets user IDN message. String is up to 72 characters and consists of four fields (manufacturer, model code, serial number and firmware revision) separated by commas. e.g. **ICS Electronics, 4806, S/N 90701, Rev 00.10, Ver 99.01.27.**
:DATe <date>		Saves IDN message and date. The save operation lights all the LEDs. Date is in mm/dd/yyyy format. A *CLS will clear the ERR LED after a CAL:DATE command.
:DATe?		Queries the calibration date. The response is 00/00/0000 for factory default settings.
DEFault		Sets 4806 memory to factory default settings.
:LOCK	0	Disables configuration commands when On. Values = 0|1 or OFF|ON. Table 1-2 lists the locked commands.
DIAGnostic		Starts diagnostic branch Diagnostic commands are reserved for factory use.

3

3.7 SMART DEVICE MESSAGES

A smart serial device can communicate with the 4806 by sending the 4806 serial messages that start with an @@@ sequence. The messages let the smart serial device set/reset bits in the 4806's Status Reporting Structure, set/query the GPIB address and IDN message and query/change the Remote/Local state. The messages are described in Table 3-4.

Commands that set bits in a 4806 register have a value which is the sum of all of the bits being set. The value can be expressed as decimal value from 0 to 32,767 or as a hex number from #h0 to #h7FFF. Commands that reset bits have a value that is the sum off all of the bits being reset.

3

The 4806 and the smart serial device communicate using a message-response protocol. The protocol assures is every message sent will receive a response. The response can be either an answer to a query or an acknowledgment that the responding device received a message. Response messages are not acknowledged.

TABLE 3-4 SMART DEVICE MESSAGES

Syntax	Meaning
@@@ESR value	Sets bits in the ESR register. ESR register bits are reset when read or by the *CLS command. Value for is 1 to 255.
@@@OPER| value	ORs bits into the Operational Condition Register. Value is 0 to 32767
@@@OPER& value	ANDs compliment of the value to reset bits in the Operational Condition Register. Value contains only the bits to be reset and is 0 to 32767.
@@@OPER?	Queries the Operational Register value.
@@@QUES| value	ORs bits into the Questionable Condition Register. Value is 0 to 32767.
@@@QUES& value	ANDs compliment of the value to reset bits in the Questionable Condition Register. Value contains only the bits to be reset and is 0 to 32767.
@@@QUES?	Queries the Questionable Register value.
@@@ADDR value	Sets the 4806 GPIB address. The address change takes affect immediately and is automaticly saved. Value is 0 to 30.
@@@ADDR?	Queries the 4806's GPIB address. The response is @@@ADDR value.
@@@SAV	Saves all 4806 setup parameters in the 4806's flash memory. Same function as the IEEE-488.2 *SAV 0 command.
@@@IDN string	Replaces the 4806's IDN message with the contents of the string. The replacement IDN message can be saved with the @@@SAV command.
@@@REM?	Queries the 4806's Remote/Local state and enables automatic reporting of Remote/Local state changes. The responses are: @@@REM 1 when the 4806 is in the remote state. @@@REM 0 when the 4806 is in the local state. @@@REM 3 when the 4806 is in the local lockout state

3

TABLE 3-4 SMART DEVICE MESSAGES (CONTINUED)

Syntax	Meaning
@@@LOC?	Asks the 4806 to go to the local state and enables automatic reporting of Remote/Local state changes. The responses are: @@@REM 1 when the 4806 is in the remote state. @@@REM 0 when the 4806 is in the local state. @@@REM 3 when the 4806 is in the local lockout state.
@@@NOREM	Disables automatic reporting of Remote/Local state changes.
@@@OK	The acknowledgement message. Indicates that the sending device received a valid message that does not have anotehr response.

3

TABLE 3-5 4806 MESSAGES TO SMART DEVICE

Syntax	Meaning
@@@ADDR value	Response to address query.
@@@REM value	Response to query or to a Local/Remote state shange. Values are 1 when in remote, 0 when in local and 3 for the local-lockout state.
@@@ERR	The error message. Indicates that the sending device received a message that was not correct or was not understood. The receiving device did not take any action as a result of the original message.
@@@OK	Acknowledgement message to acknowledge receipt of a valid message.

Notes:
1. All values are in decimal. To enter HEX values, the value must be preceded with a #h . i.e. 100 decimal = #h64
2. All messages are terminated with a linefeed character

3.8 PROGRAMMING GUIDELINES

The following section provides information on how to program the 4806 to set its configuration and how to serial messages to the serial device.

3.8.1 Query the 4806's IDN Message

Read the 4806's IDN message to verify that you have a good GPIB communication with the 4806. To read the IDN message, follow the sequence suggested below.

Send IFC	'gets control, asserts REN
Send *IDN? **Read response**	'read current setting
Display the response	'verify that the initial IDN message is **ICS Electronics, 4806, S/N 90701, Rev 00.00, Ver 99.01.27.**

3

3.8.2 General Configuration Guidelines

New units are factory set so that they are normally ready to be used when received. Table 1-1 lists the Factory Configuration. To change the configuration, the user should follow the sequence outlined below. Select the SCPI command from Table 3-2.

Send IFC	'gets control, asserts REN
Send SCPI command	'send new value. Verify the ERR LED is off.
Send SCPI query	'read current setting
Send "*SAV 0"	'save the new setup

The *SAV 0 command saves the current configuration in the 4806's flash memory. When saving the configuration, the 4806 will blink all but one of its LEDs. If any serial parameters were changed, use the :UPdate command to update the UART.

Pay close attention to the ERR LED when sending commands. If the ERR LED comes on, the unit's parser detected a problem with the command and DID NOT execute it.

3.8.3 Sending Serial Messages

3

After the 4806 is correctly configured to match the serial device's settings, use the GPIB output command to send the 4806 a test message that it can pass onto the serial device.

```
Call ieOutput(04, "test string")      'send test string
```

3.8.4 Reading Serial Messages

Set the 4806's timeout to a value 10 times longer than the serial device's expected response time. Next, send the serial device a message that will generate a response.

```
Call ieOutput(04, "get-response")  'send test string
Call ieEnter(04, Rdg$)             'read response
```

3.8.5 Changing the GPIB Address

The 4806's GPIB address is changed or queried with the SCPI SYST:COMM:GPIB:ADDR command. The change takes place immediately so the user will have to change the 4806's address in the computer before sending the 4806 the next command. A Quick Basic example is:

```
CmdStr$ = "SYST:COMM:GPIB:ADDR 07"
Call ieOutput(04, CmdStr$)          'new GPIB address
CmdStr$ = "*SAV 0"
Call ieOutput(07, CmdStr$)          'saves new value
```

3.8.6 Changing the 4806's IDN Message

The 4806's IDN message is changed with the CALIBRATE subsystem commands. Change the IDN message when you want to identify the overall assembly as being from your company or to record product history or revision dates. The IDN message is a lockable parameter and if locked, needs to be unlocked before being changed. The format for the IEEE 488.2 IDN message is four fields (company, model#, serial number and revision) separated by commas and a maximum of 72 characters long. The word "model" may not be used in an IEEE-488.2 IDN message. An example IDN message change sequence is:

3

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

3.8.7 Locking Setup Parameters

All of the 4806's configuration parameters can be locked to prevent accidental change by the end user. These lockable parameters are noted by a # symbol in Table 1-1. Locked parameters cannot be queried or changed while locked. Any command that addresses a locked parameter is not executed, the Command Error bit in the Event Status Register is asserted and the ERR LED is lit. The lock function is saved by the *SAV 0 command. An example of using the lock command is:

```
CAL:LOCK OFF       'unlocks the setup parameters

      ...          'change 4806 setup

CAL:LOCK ON        'set lock on
*SAV 0             'saves lock condition
```

While lock is enabled, the end-user can only change and save any parameter that is not controlled by the lock function.

3.9 OEM Documentation Guidelines

OEM users of the 4806 Interface Boards should provide the end user with the necessary instructions to operate the complete system. In most cases this includes directions for:

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

3

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

OEM users are hereby given permission to copy any portion of this manual for the purpose of documenting systems that incorporate the Model 4806 Interface Board. Reproduction of this manual for other purposes without the expressed written consent of ICS Electronics is forbidden.

4

Theory of Operation

4.1 INTRODUCTION

This section describes the theory of operation of the 4806.

4.2 BLOCK DIAGRAM DESCRIPTION

A block diagram of the 4806 is shown in Figure 4-1. The 4806 is a microprocessor based device that transparently passes data between the GPIB (IEEE 488) bus and a serial device over a serial link. The 4806 also has a serial I/O connector for communicating directly with the serial device. The 4806 is made up of seven major elements, most of which are interconnected to the microprocessor by a common data, address and control signal bus.

Incoming GPIB bus data and commands are received by the GPIB controller chip. Each received character interrupts the microprocessor to place the characters in the GPIB received data buffer. When a complete message has been received, the parser checks the message to see if it contains an IEEE-488.2 Common Command or if it starts with one of the reserved keywords. If it does, then the message is a 4806 command and the parser completes the command interpretation process.

Valid messages are acted upon and used to set control parameters, perform an operation or query a parameter. Responses are placed in the GPIB buffer so they can be returned to the host controller when the

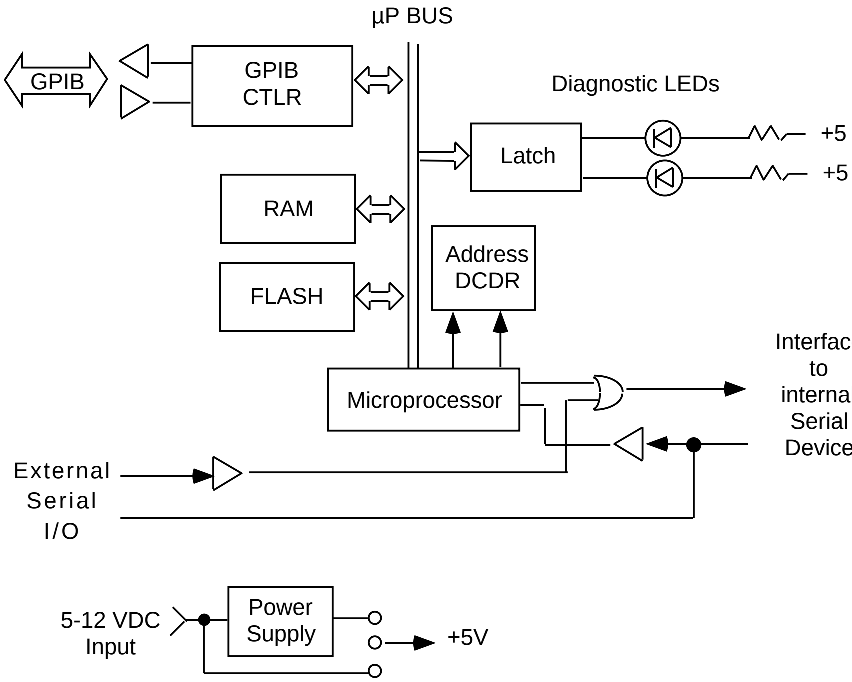


Figure 4-1 4806 Block Diagram

unit is next addressed to talk. Invalid messages cause a bit to be set in the unit's Event Status Register and turn on the ERR LED.

If the message is not a 4806 command, then the data characters are sequentially placed in the microprocessor's UART where they are serialized, passed through the OR gate and outputted at RS-232 levels to the serial device. The other input to the OR gate comes from the serial I/O connector.

Incoming data from the serial device is received as RS-232 levels, converted into TTL levels and applied to the UART's input. The received RS-232 signal is also routed to the serial I/O connector. Each received character is stored in the received data buffer where it can be transferred out onto the GPIB bus when the unit is next addressed to talk.

The Flash contains all of the 4806's program instructions, command tables, and power turn-on/self test routines. At power turn-on, the 4806 performs a self test on each functional block to determine whether there is a gross system failure. Any self test error is displayed as a pattern of blinking LEDs on the front panel. The error pattern is repeated until the unit is turned off. Just after completing the self test routine, the 4806 displays its current GPIB address setting on the front panel LEDs. Bit weights are read from right to left with the least significant bit on the far right. The RDY LED comes on to indicate a successful completion of the self test routine.

The Flash also contains all of the 4806's configuration settings, serial number and other parameters that are subject to change. At power on time, the microprocessor copies the configuration from Flash memory to RAM where it is used to operate the unit. Any changes made to the settings during run time are <u>not</u> stored in the Flash memory until the user sends the 4806 the *SAV 0 command.

In the 4806 , the RAM is a 8 bit wide memory that is primarily used for data storage, operating variables and configuration settings. The 4806 data buffers are several times larger than any anticipated message so no data loss ever occurs. GPIB bus data is never lost since the 4806 simply inhibits further Bus handshakes until there is room in the GPIB buffer for more data.

The 4806 can be powered from 5 Vdc regulated power or from + 5.5 to + 12 Vdc unregulated power. The unregulated voltage is regulated to + 5 Vdc by a linear regulator on the 4806 PC board. A DC-DC converter in the RS-232 transmitter IC makes the ± 9 Vdc necessary to power the RS-232 drivers.

5

Troubleshooting and Repair

5.1 INTRODUCTION

This section describes the maintenance, troubleshooting and repair procedures for the Model 4806 GPIB <-> Serial Interface.

5.2 MAINTENANCE

The 4806 does not require periodic calibration and have no internal adjustments. However, if the 4806 is used in an application where the IEEE 488 bus cables are frequently changed or if the input signals appear erratic, the 4806's GPIB connector may require cleaning to remove wax and dirt buildup. New bus and other 'blue ribbon' type connectors are shipped with a brightener on them. (The brightener is a thin wax like film) Depending upon cable usage, enough of the brightener may buildup on the 4806's connector to cause intermittent operation.

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

5.3 TROUBLESHOOTING

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

5.3.1 Self Test Errors

The 4806 and 4806 indicate self test errors by blinking one or more of its LEDs at a 2 cps rate. Verify the error by turning the unit off for 10 seconds, disconnect the unit from any other equipment and then turn the power back on. If the error persists it is a true self test error. The self test error codes and their most likely problems are listed in Table 5-1.

5.3.2 Operating Failures

Use the fault isolation information in Table 5-2 to narrow the 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. See paragraph 5.6 for repair instructions.

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 4806 - SELF TEST ERROR CODES

Blinking LED	Error	Possible Fault
All	Flash Memory	Flash Memory corrupted and unit reset Flash 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 4806 Flash, U2. Defective 4806 Flash that cannot be reprogrammed. Replace U2. Defective decoder GAL. Test and/or replace GAL U5 in 4806.
RDY	Flash	4806 Flash failed write test.. Replace Flash, U2, and reprogram configuration. See paragraph 5.5.
TALK	Flash	4806 Flash loose in its socket or has a bent pin. Check Flash for a bent pin then press Flash into its socket. Flash Memory dropped a bit. Replace with a known good Flash. When the 4806's Flash Memory is replaced, the configuration will have to be reprogrammed. See paragraph 5.5 Defective decoder GAL. Test and/or replace GAL U5.
LSTN	RAM	Defective decoder GAL. Test and/or replace GAL U5. Defective RAM. Replace RAM U3 in 4806.

5

TABLE 5-1 4806 - SELF TEST ERROR CODES CONTINUED

Blinking LED	Error	Possible Fault
SRQ	GPIB	Defective GPIB controller chip. Replace GPIB Controller U8.
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, U8 with NI 7210.
ERR + LSTN	CPU	Wrong CPU type. Should be Zilog Z8S18020VSC (SL1919 Enhanced Version) Replace U9 with correct part.
ERR + LSTN + TALK	CPU	Wrong CPU type. Should be Zilog Z8S18020VSC. Replace U9 with correct part.

Solid LED	Error	Possible Fault
PWR (After blinking address)	Program hung	Open GPIB chip selection line or grounded interrupt into Z180

5

TABLE 5-2 TROUBLESHOOTING GUIDE

Symptom	Possible Fault	Action or Check
Unit will not turn on	Power cord not plugged in	Push power cord into DC receptacle
	Power at AC outlet	Check outlet and power adapter
	High output lines shorted to ground	Disconnect output signals and reapply power to test the unit. If it powers on, remove the short or put resistors in the offending circuit path.
Unit shows a blinking LED at power turn on	Self test fault	Check Self Test errors in Table 5-1
ERR LED on at power turn-on	Flash data lost	Use *CLS to clear the LED. Use CAL:DATE command to 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	No delay after an the address change	Provide a 70 ms minimum delay after changing 4806's GPIB address
	Insufficient delay	Program running on faster CPU or in a compiled form runs faster than expected. Change delay function to a called time function that tests CPU clock.
String data transfer fails	Wrong GPIB address	Check device address in program.

5

TABLE 5-2 TROUBLESHOOTING GUIDE CONTINUED

Symptom	Possible Fault	Action or Check
String data transfer fails - Continues	String starts with a reserved keyword or asterisk.	Check command string for asterisk or reserved keywords.
No response data	No talk data	Serial unit did not send string Query Operational Condition Register for buffer status
	Serial unit response longer than timeout	Monitor serial input with an oscilloscope to check timing.
	Serial response not read into computer	Wrong or missing string terminator. Check serial input to the 4806 with a terminal to verify/correct string termination.
LEDs power on in strange pattern and unit no longer runs	Decoder U5 bad	Replace GAL U5 with new IC. Contact factory for a replacement part.
	Wrong voltage	Check TP+5 for 5 ± 0.2 Vdc.
	Wrong Power Selection Jumper position.	Check Power Selection Jumper position and correct.
	Regulator very hot	Input voltage over regulator limit. Reduce input voltage or add heatsink to regulator.
	Good voltage but?	Board may have been damaged. Contact factory.

5

5.4 4806 FACTORY DEFAULT RECOVERY

If a 4806's configuration gets into an unknown setting, it can be restored to the factory default configuration with the following procedure:

1. Turn the 4806's power off.

2. Place a jumper on W1 if the jumper is missing.

3. Place a jumper on W3 or short out the two posts on W3.

4. Turn the 4806's power on. Wait until the LEDs stabilize and the 4806 has finished its save procedure (takes approximately 10 seconds).

5. Remove the jumper from W3 and operate the unit normally. All parameters should have been restored to the default settings listed in Table 1-2.

5.5 FLASH REPLACEMENT

The following steps should be followed when replacing the Flash chip in an ICS unit:

1. Turn power off and leave the chassis grounded to the AC outlet.

2. Ground yourself by touching the chassis with both hands.

3. Gently pry the existing Flash up out of the socket using a rocking motion.

4. Remove the new Flash from its protective case without touching the leads.

5. Adjust the new Flash's leads if necessary by gently pressing each row of pins against the edge of the table. Adjust both rows equally. Line up all pins and press the Flash into the socket.

6. Place a jumper on W1 if the jumper is missing.

7. Turn the unit on. All LEDs will flash on while the factory defaults are being programmed. Turn the unit off and on. The ERR LED will come on after selftest.

8. Review the default settings in table 1-2 and make any changes necessary for your installation.

9. Send the unit the CAL:DATE command with today's date. The LEDs will all blink on while the date is being saved.

10. Remove the jumper if one was installed in step 5 above.

5.6 REPAIR

Repair of the 4806 is done by the user or by returning the unit to the factory or to your local distributor. Units in warranty should **always** be returned to the factory or else repaired only after receiving permission to do so from an ICS customer service representative.

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

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

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

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

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

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

Appendix

A

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

A1.1 IEEE 488.1

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 lines 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.

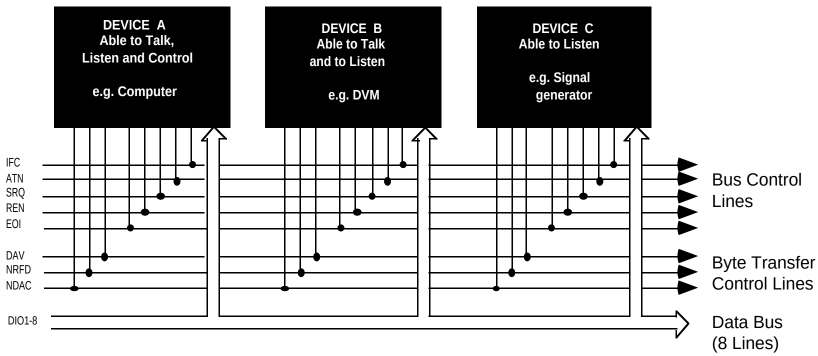


Figure A-1 IEEE 488 Bus

A1

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 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.

TABLE A-1 IEEE 488 COMMAND AND ADDRESS MESSAGES

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

LSD \ MSD	0 ASCII	0 MSG	1 ASCII	1 MSG	2 ASCII	2 MSG1	3 ASCII	3 MSG1	4 ASCII	4 MSG1	5 ASCII	5 MSG1	6 ASCII	6 MSG	7 ASCII	7 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	▼

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

PRIMARY COMMAND GROUP (PCG) (MSD 0–5)

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

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

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

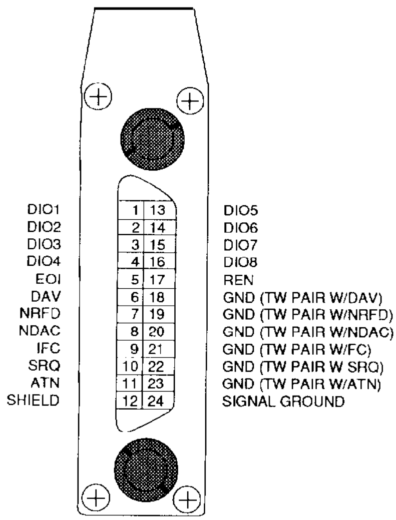


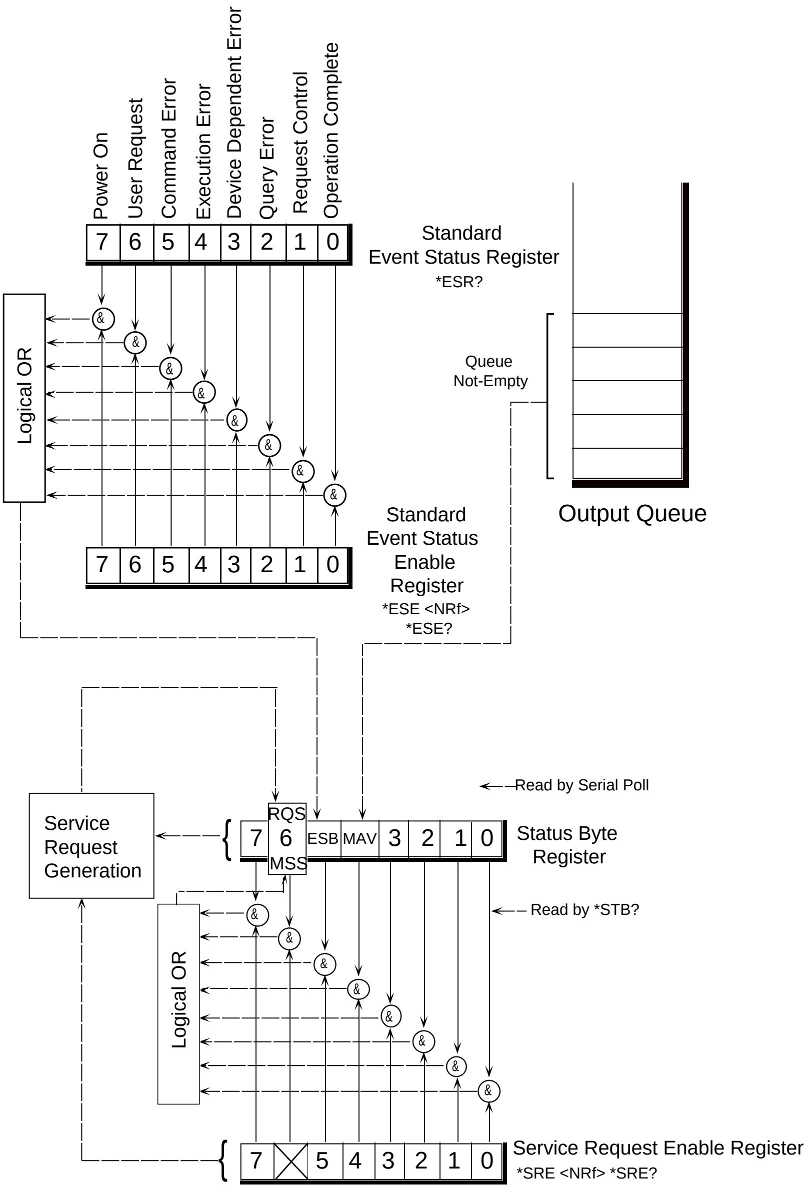
Figure A-2 GPIB Connector Signal Pinouts

A1.2 IEEE 488 MESSAGE FORMATS (IEEE 488.2)

IEEE Std 488.2 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 RESPONSE MESSAGE.
(*)	Asterisk defines a 488.2 common command.
(?)	Ends a query where a reply is expected.

With 488.2, status reporting was enhanced from the simple serial poll response byte in 488.1 to the multiple register concept shown in Figure A-3. Each 488.2 device must implement a Status Byte Register, a Standard Event Status Register and an Output Message Queue. Both registers must have enabling registers that can control the generation of their reporting bits and ultimately SRQ generation. This 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. A 488.2 device outputs the Status Byte Register contents plus the RQS bit in response to a serial poll. A device may include any number of condition registers, event registers and enabling registers providing they follow the model shown in Figure A-3.



A1

Figure A-3 488.2 Required Status Reporting Capabilities

The 488.2 specification also mandated a list of common commands that all devices will support. These commands are:

1	***CLS**	Clear Status Command
2	***ESE**	Standard Event Status Enable Command
3	***ESE?**	Standard Event Status Enable Query
4	***ESR?**	Standard Event Status Register Query
5	***IDN?**	Identification Query
6	***OPC**	Operation Complete Command
7	***OPC?**	Operation Complete Query
8	***RST**	Reset Command
9	***SRE**	Service Request Enable Command
10	***SRE?**	Service Request Enable Query
11	***STB?**	Status Byte Query
12	***TST?**	Self-Test Query
13	***WAI**	Wait-to-Continue Command

In addition to the above common commands, 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

A1

A1.3 SCPI COMMANDS

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.

COMMANDS

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 queuing a serial interface. Refer to Table 3-3 for a complete description of the SCPI commands used by this unit.

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

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

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

Figure A-4 SCPI Command Examples

Multiple SCPI commands may be concatenated together as a compound command using semi colons as command separators. The first command is always referenced to the root node. Subsequent commands are referenced to the same tree level as the previous command. Starting the subsequent command with a colon puts it back at the root node. Common commands and queries (start with *) 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

The response will be: **9600; 0; 8; XON; 16 <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 **16** (ESR) which means execution error - caused by the BIT 6 command.

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. Table 3-3 lists the SCPI errors reported by this unit.

A2 SERIAL DATA COMMUNICATIONS BACKGROUND

A2.1 INTRODUCTION

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 teletypes

300, 1200 - for low speed devices and telephone modems

4800, 9600, 19200 - for high speed devices and modems

A2

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 format is called asynchronous data transmission because each character is sent one at a time with an undetermined amount of time between characters.

Each asynchronous character has a low going start bit, a number of data bits, an optional parity bit and 1 or 2 high stop bits. The transmitter automatically extends the stop bit when it has no more characters to transmit. The receiver uses the start bit to resynchronize its clock with the data at the start of each character as shown in Figure A-6.

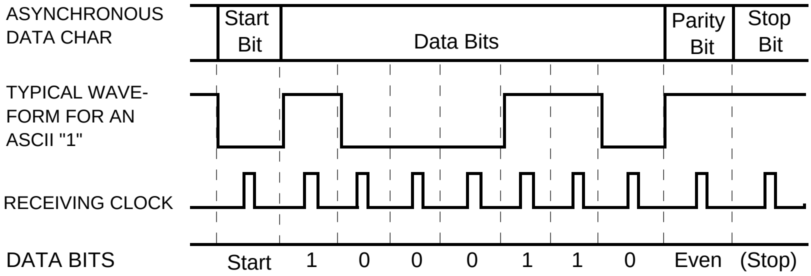


Figure A-6 Asynchronous Data Character Waveforms

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-6 shows how the ASCII "1" is transmitted. Compare the binary code in Figure A-5 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 8 bit characters or in hex form as 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.

A2

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. Most GPIB-to-Serial interfaces that detect parity errors along with data overrun and framing errors, will typically generate a Bus SRQ message for each data error and indicates the error by setting 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-7. 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.

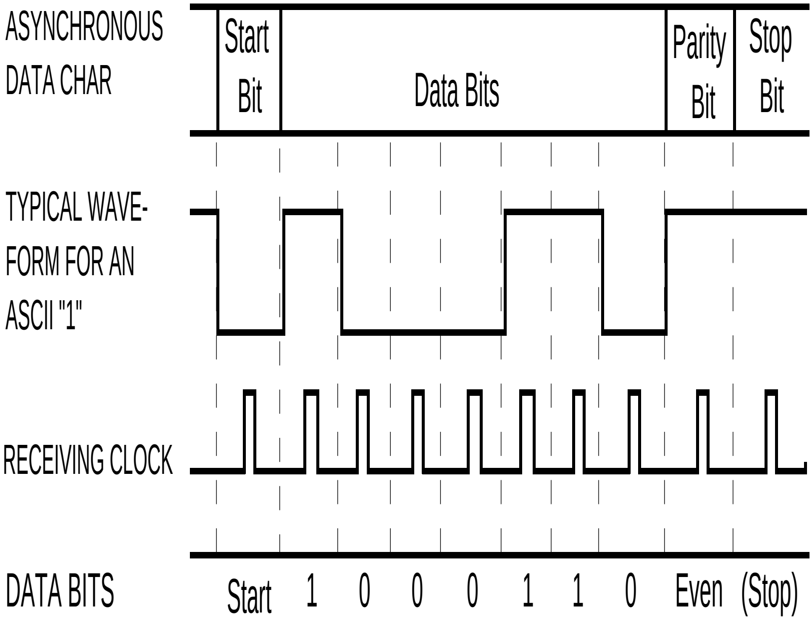


Figure A-7 Long Distance Transmission Using 4894's

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

A2

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 4806 normally uses asynchronous 8 bit data characters with no parity and with single start and stop bits. The 4806 will also work with 7 bit data characters. The unused data bits are outputted on the GPIB Bus as fixed zeros. The user can also add a parity bit and the second stop bit if required for his system.

A2.2 RS-232 STANDARD

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

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

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

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

Electrically, RS-232 signals are bi-polar 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)

A2

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

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

A2.3 RS-422 AND RS-485 STANDARDS

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

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

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

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

A2

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

A2.4 RS-530 PINOUTS

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

A2.5 SERIAL INTERFACE PROBLEMS

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

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

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

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

   i.e.,*!1-

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

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

A2

Index

I

I