OMB-SERIAL488A
User's Guide

p/n GE380-901, Rev. 2.3

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Introduction

1.1 Description

The **Serial488A, Serial488A/OEM** and **Serial488/512K** Bus Converters provide transparent communication from a serial computer to an IEEE 488 printer, plotter or other device. They also can be used to control a serial device, such as a printer or terminal, from an IEEE 488 host computer.

As a serial to IEEE 488 converter, it receives data from a serial host then automatically performs the bus sequences necessary to send this data to the IEEE 488 device. If desired, data can be requested from the IEEE 488 device and returned to the host.

As an IEEE 488 to serial converter, it functions as a peripheral to an IEEE 488 controller. Data received from the controller is sent to the serial device, and data received from the serial device is buffered for transmission to the IEEE 488 controller. The converter can inform the host, by the serial poll status byte, that it has received data from the serial device.

The **Serial488A** and **Serial488A/OEM** can communicate with RS-232 and RS-422 devices by positioning configuration jumpers located within the unit. Both devices can communicate at selectable baud rates up to 57600 baud.

The **Serial488/512K** can communicate with RS-232 devices at selectable baud rates up to 19200 baud.

This manual will refer to all three interfaces as the **Serial488A**. Differences between the **Serial488A, Serial488A/OEM** and **Serial488/512K** will be noted where applicable.

1.2 Serial488 and Serial488A Differences

The **Serial488A** is both a hardware and firmware upgrade to the **Serial488**. When issuing product improvements, we try to maintain transparent compatibility. Occasionally this is not possible. You should note the following differences between the two products.

1. The **Serial488** allocated fixed serial and IEEE input buffers of 4000 characters. The **Serial488A** utilizes a 32,000 character buffer which is dynamically allocated to the serial and IEEE input buffers as required. Refer to Section 4.2 for more details

2. The **Serial488A** has the ability to output RS-232 or RS-422 levels. The levels used are internally selectable. Refer to Section 2.8 for details.

3. As a peripheral, the **Serial488A**'s serial poll status byte has been changed to include status of the serial handshake. Other minor changes have also been included. Refer to Section 4.4 for a complete description of the serial poll status byte.

4. The internal switch settings for baud rate have been adjusted to include 57600 baud. Refer to Section 2.3 for the new switch settings.

1.3 Serial488A, Serial488A/OEM and Serial488/512K Differences

1. The **Serial488A** and **Serial488A/OEM** utilize a 32,000 character buffer which is dynamically allocated to the serial and IEEE input buffers as required. The **Serial488/512K** utilizes a 512,000 character buffer which is dynamically allocated to the serial and IEEE input buffers as required.

2. The **Serial488A** and **Serial488A/OEM** have the ability to output RS-232 or RS-422 levels. The **Serial488/512K** can only operate at RS-232 levels.

3. The **Serial488A** and **Serial488A/OEM** can operate at selectable baud rates up to 57600 baud. The **Serial488/512K** can operate at selectable baud rates up to 19200 baud.

1.4 Available Accessories

Additional accessories that can be ordered for the **Serial488A** include:

CA-7-1	1.5 foot IEEE 488 Cable
CA-7-2	6 foot IEEE 488 Cable
CA-7-3	6 foot shielded IEEE 488 Cable
CA-7-4	6 foot reverse entry IEEE 488 Cable
CA-11	12 foot IBM PC/XT/PS2 to Serial488A RS-232 Cable
CA-21	12 foot Macintosh II/SE/Plus to Serial488A RS-232 Cable
CA-22	12 foot Macintosh 512K to Serial488A RS-232 Cable
CA-23	12 foot IBM AT to Serial488A RS-232 Cable
CN-20	Right Angle IEEE 488 adapter, male and female
CN-22	IEEE 488 Multi-tap bus strip, four female connectors in parallel
CN-23	IEEE 488 panel mount feed-through connector, male and female
ABC488	IEEE 488 ABC switch
Rack488-3	5-1/4" by 19" rack mount for one **Serial488A**
Rack488-4	5-1/4" by 19" rack mount for two **Serial488As**
140-0920	Instruction Manual

1.5 Specifications

Serial488A

IEEE 488-1978

Implementation:	C1, C2, C3, C4 and C28 controller subsets.(Serial to IEEE) SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT0, E1.
Terminators:	Selectable CR, LF, LF-CR and CR-LF with EOI.
Connector:	Standard IEEE 488 connector with metric studs.

Serial Interface

EIA RS-232C:	AB, BA, BB, CA, CB
EIA RS-422A:	Balanced voltage on TxD and RxD.
Character Set:	Asynchonous bit serial.
Output Voltage:	±5 volts min (RS-422A). 5 volts typical (RS-232C).
Input Voltage:	±3 volts min.; ±15v max.
Baud Rate:	Selectable 110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19,200 and 57,600.
Data Format:	Selectable 7 or 8 data bits; 1 or 2 stop bits; odd, even, mark, space and no parity on transmit.
Duplex:	Full with Echo/No Echo.
Serial Control:	Selectable CTS/RTS or XON/XOFF.
Terminators:	Selectable CR, LF, LF-CR and CR-LF.
Connector:	25-pin Sub-D male. DCE Configured.

General

Data Buffer:	32,000 characters dynamically allocated.
Indicators:	LEDs for IEEE Talk and Listen, Serial Send and Receive, and Power.
Power:	105-125V or 210-250V; 50-60 Hz, 10 VA Max.
Dimensions:	188mm deep x 140mm wide x 68mm high (7.39" x 5.5" x 2.68").
Weight:	1.55 kg. (3.6 lbs).
Environment:	0° - 50°C; 0 to 70% R.H. to 35°C. Linearly derate 3% R.H./°C from 35° to 50°C.
Controls:	Power Switch (external), IEEE and Serial parameter switches (internal). Jumper selection of RS-232 or RS-422 operation (internal).

Specifications subject to change without notice.

Serial488A/OEM

IEEE 488-1978	
Implementation:	C1, C2, C3, C4 and C28 controller subsets.(Serial to IEEE) SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT0, E1.
Terminators:	Selectable CR, LF, LF-CR and CR-LF with EOI.
Connector:	Standard IEEE 488 connector with metric studs.
Serial Interface	
EIA RS-232C:	AB, BA, BB, CA, CB
EIA RS-422A:	Balanced voltage on TxD and RxD.
Character Set:	Asynchonous bit serial.
Output Voltage:	±5 volts min (RS-422A). 5 volts typical (RS-232C).
Input Voltage:	±3 volts min.; ±15v max.
Baud Rate:	Selectable 110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19,200 and 57,600.
Data Format:	Selectable 7 or 8 data bits; 1 or 2 stop bits; odd, even, mark, space and no parity on transmit.
Duplex:	Full with Echo/No Echo.
Serial Control:	Selectable CTS/RTS or XON/XOFF.
Terminators:	Selectable CR, LF, LF-CR and CR-LF.
Connector:	25-pin Sub-D male. DCE Configured.
General	
Data Buffer:	32,000 characters dynamically allocated.
Indicators:	LEDs for IEEE Talk and Listen, Serial Send and Receive, and Power.
Power:	User supplied +5 volts ±0.25% at 1 amp. Mating power connector with 8 inch leads provided.
Dimensions:	205mm deep x 115mm wide x 28mm high (8" x 4.5" x 1.1").
Weight:	0.23kg. (0.5 lbs.)
Environment:	0° - 50°C; 0 to 70% R.H. to 35°C. Linearly derate 3% R.H./°C from 35° to 50°C.
Controls:	IEEE and Serial parameter switches. Jumper selection of RS-232 or RS-422 operation.

Specifications subject to change without notice.

Serial488/512K

IEEE 488-1978

Implementation:	C1, C2, C3, C4 and C28 controller subsets.(Serial to IEEE) SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT0, E1.
Terminators:	Selectable CR, LF, LF-CR and CR-LF with EOI.
Connector:	Standard IEEE 488 connector with metric studs.

Serial Interface.

EIA RS-232C:	AB, BA, BB, CA, CB
Character Set:	Asynchonous bit serial.
Output Voltage:	5 volts typical (RS-232C).
Input Voltage:	±3 volts min.; ±15v max.
Baud Rate:	Selectable 110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, and 19,200.
Data Format:	Selectable 7 or 8 data bits; 1 or 2 stop bits; odd, even, mark, space and no parity on transmit.
Duplex:	Full with Echo/No Echo.
Serial Control:	Selectable CTS/RTS or XON/XOFF.
Terminators:	Selectable CR, LF, LF-CR and CR-LF.
Connector:	25-pin Sub-D male. DCE Configured.

General

Data Buffer:	512,000 characters.
Indicators:	LEDs for IEEE Talk and Listen, Serial Send and Receive, and Power.
Power:	105-125V or 210-250V; 50-60 Hz, 10 VA Max.
Dimensions:	188mm deep x 140mm wide x 68mm high (7.39" x 5.5" x 2.68").
Weight:	1.95 kg. (4.35 lbs).
Environment:	0° - 50°C; 0 to 70% R.H. to 35°C. Linearly derate 3% R.H./°C from 35° to 50°C.
Controls:	Power Switch (external), IEEE and Serial parameter switches (internal).

Specifications subject to change without notice.

1.6 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"
ATN	Attention line
CA	Controller Active
CO	Controller
CR	Carriage Return
data	Data String
DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LA	Listener Active
LAG	Listen Address Group
LF	Line Feed
LLO	Local Lock Out
MLA	My Listen Address
MTA	My Talk Address
PE	Peripheral
PPC	Parallel Poll Configure
PPU	Parallel Poll Unconfigure
SC	System Controller
SDC	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
SRQ	Service Request
TA	Talker Active
TAD	Talker Address
TCT	Take Control
term	Terminator
UNL	Unlisten
UNT	Untalk
*	Unasserted

Getting Started

2.1 Inspection

The Serial488A was carefully inspected, both mechanically and electrically, prior to shipment. When you receive the interface, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage which may have occurred during shipment. Immediately report any such damage found to the shipping agent. Remember to retain all shipping materials in the event that shipment back to the factory becomes necessary.

Every Serial488A is shipped with the following....

• **Serial488A**	IEEE 488 Bus Converter
• **140-0920**	Instruction Manual
• **Power Supply**	9 Volt Regulated TR-2; 115V or TR-2E; 220V

Every Serial488A/OEM is shipped with the following....

• **Serial488A/OEM**	IEEE 488 Bus Converter
• **140-0920**	Instruction Manual

Every Serial488/512K is shipped with the following....

• **Serial488/512K**	IEEE 488 Bus Converter
• **140-0920**	Instruction Manual
• **Power Supply**	9 Volt Regulated TR-2; 115V or TR-2E; 220V

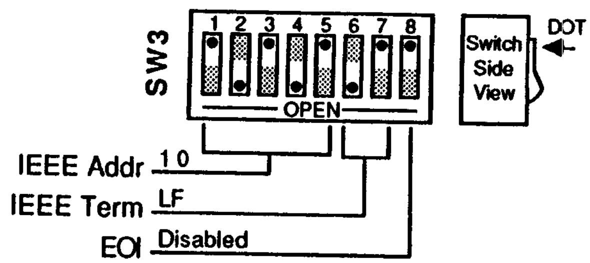
2.2 Configuration

Three DIP switches internal to the **Serial488A** set the configuration of the interface. NOTE: Selectable functions are read ONLY at power-on and should only be set prior to applying power to the interface. The following figures illustrate the factory default settings which are:

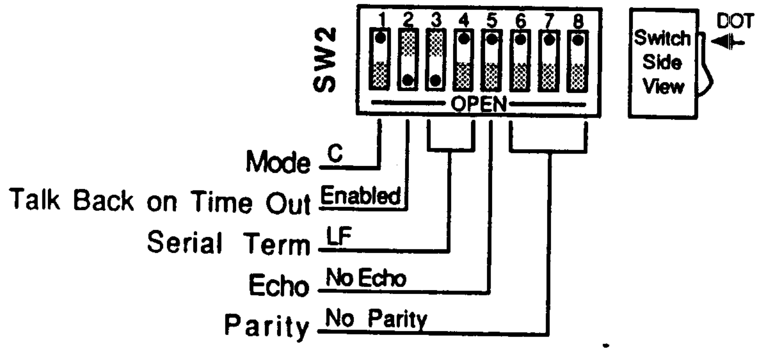
Serial Port:
9600 Baud
8 Data Bits
2 Stop Bits
No Parity
Serial Terminator = LF
Echo Disabled
RTS/CTS Handshake

IEEE:
Mode = IEEE 488 Controller
Address = 10
Bus Terminator = LF; EOI Disabled
Talk-back on Terminator Enabled
Talk-back on Time Out Enabled

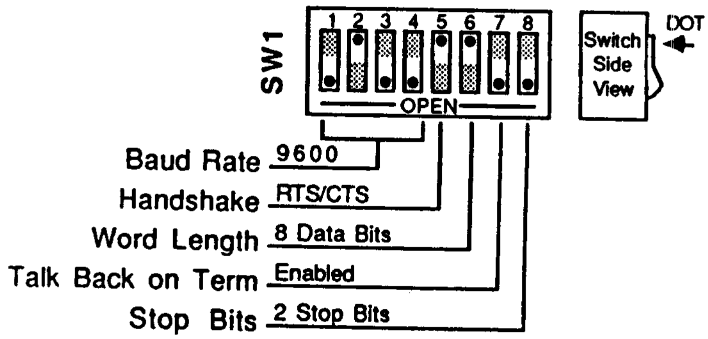
SW3 Factory Default Settings



SW2 Factory Default Settings

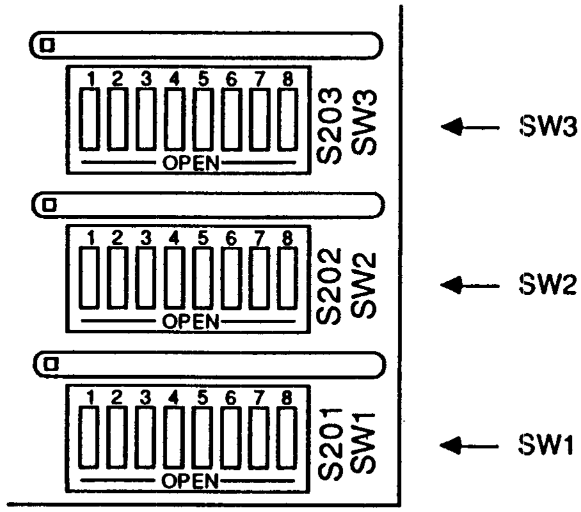


SW1 Factory Default Settings

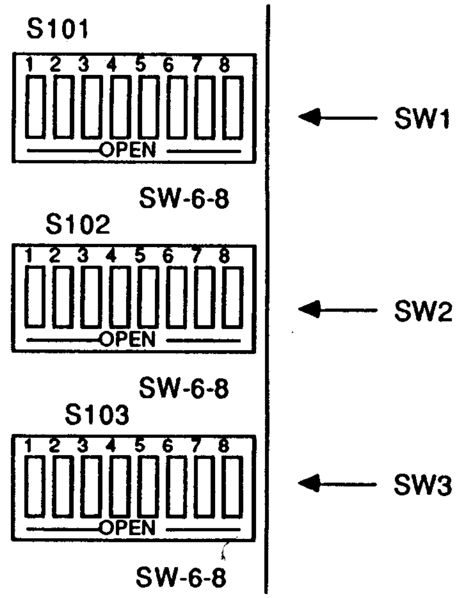


The following drawings show the locations of switches SW1, SW2, and SW3 for the **Serial488A**, **Serial488A/OEM** and the **Serial488/512K**. The top circuit board in the **Serial488A** and the **Serial488/512K** is referred to as the I/O board.

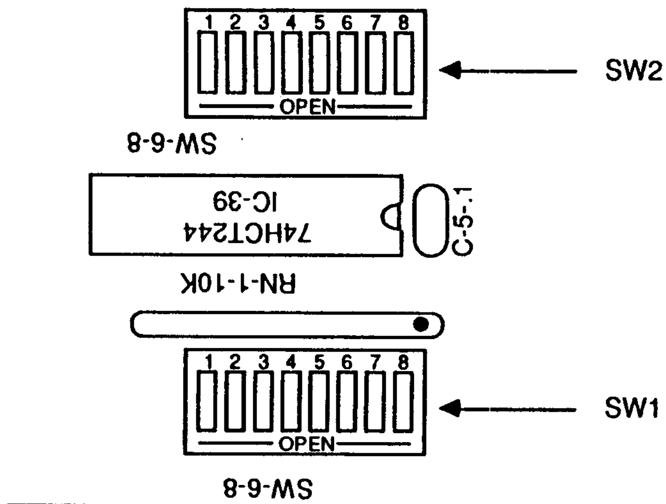
Serial488A Switch Location - I/O Board



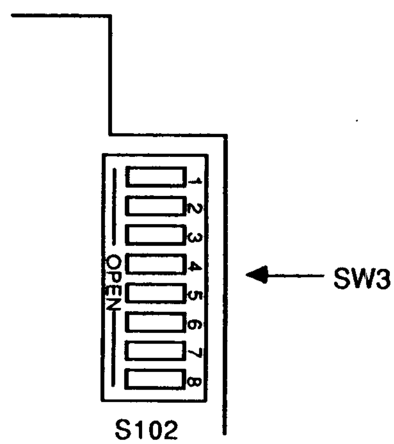
Serial488A/OEM Switch Location



Serial488/512K SW1, SW2 Location - I/O Board



Serial488/512K SW3 Location - Motherboard



Note that the **Serial488A** comes configured as an IEEE controller. In this mode the **Serial488A** is designed to allow an RS-232 computer to communicate with an IEEE peripheral such as a plotter. This controller mode is described in detail in **Section 3**.

The **Serial488A** may also be configured as an IEEE peripheral. As an IEEE peripheral, the **Serial488A** allows an IEEE controller to communicate with an RS-232 device. The peripheral mode of operation is described in detail in **Section 4**.

To modify any of these defaults, follow this simple procedure: Disconnect the power supply from the AC line and from the interface. Disconnect any IEEE or serial cables prior to disassembly.

WARNING

Never open the Serial488A case while it is connected to the AC line. Failure to observe this warning may result in equipment failure, personal injury or death.

Place the interface upside down on a flat surface. Remove the four (4) screws located near the rubber feet. Return the interface to the upright position and carefully remove the top cover. Modify those parameters which are appropriate for your installation and then carefully re-assemble the interface using the reverse of the procedure described.

2.3 Serial Port Settings

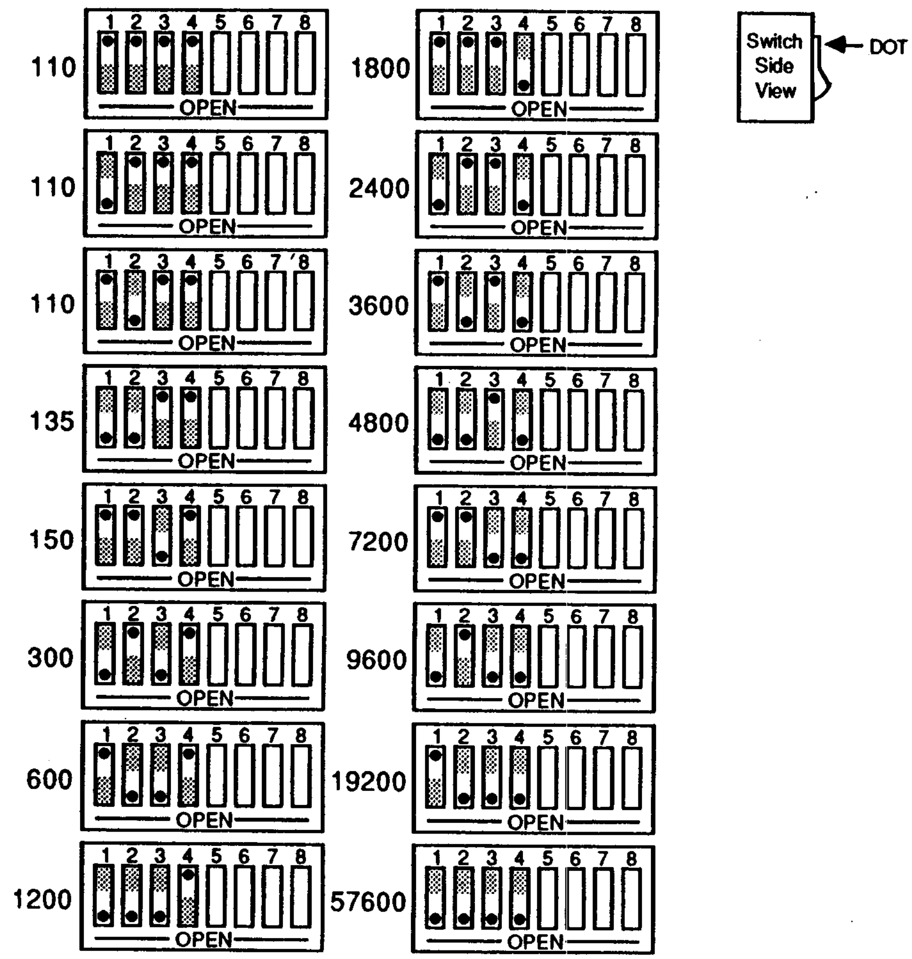
The first parameters to configure are those that correspond to the RS-232 port. These include baud rate, word length, number of stop bits, parity selection and type of RS-232 handshake. Each of these are described in the following sections.

2.3.1 Serial Baud Rate Selection

Baud rate defines the number of serial bits per second transferred into and out of the RS-232 interface. SW1-1 through SW1-4 determine the serial baud rate. The factory default baud rate is 9600 baud. Baud rates may be selected from 110 to 57600 baud (110 to 19200 for the **Serial488/512K**).

Refer to the following diagram for specific baud rates. Note: on the **Serial488/512K**, selecting 57600 baud will have the same effect as selecting 19200 baud.

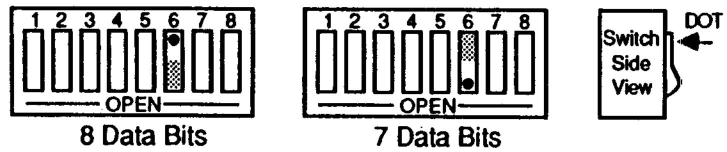
SW1 View for Serial Baud Rate Selection



2.3.2 Serial Word Length Selection - Data Bits

SW1-6 determines the number of data bits, often referred to as word length, for each serial character transmitted or received. The factory default is 8 data bits.

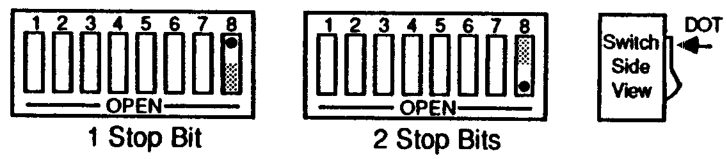
SW1 View of Serial Word Length (Data Bits) Selection



2.3.3 Serial Stop Bit Selection

Switch SW1-8 determines the number of stop bits contained in each serial character transmitted and received. The factory default is 2 stop bits.

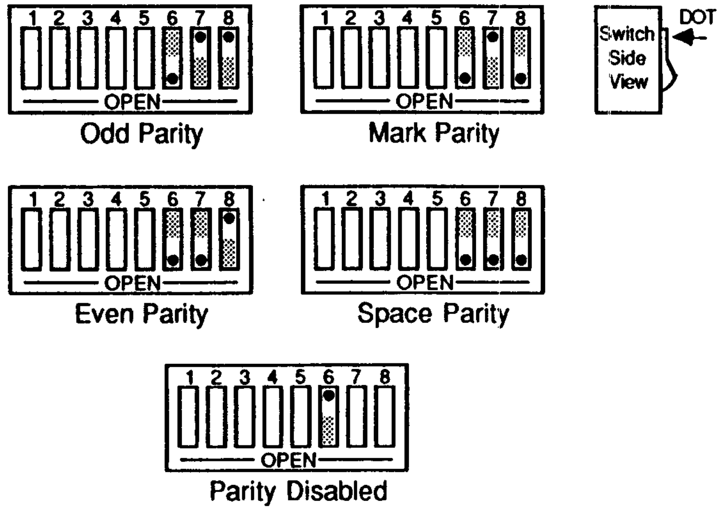
SW1 View for Serial Stop Bit Selection



2.3.4 Serial Parity Selection

Serial Parity is selected with S2-6 through S2-8. The **Serial488A** generates the selected parity during serial transmissions but it does not check parity on data that is received. The factory default is parity disabled.

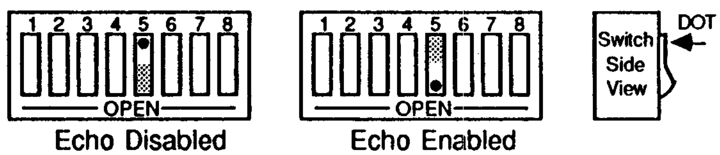
SW2 View for Serial Parity Selection



2.3.5 Serial Echo Selection

Serial data sent to the **Serial488A** will be echoed back to the serial host if SW2-5 is set to the open position. Factory default is Echo Disabled.

SW2 View for Echo Selection



2.3.6 Serial Handshake Selection

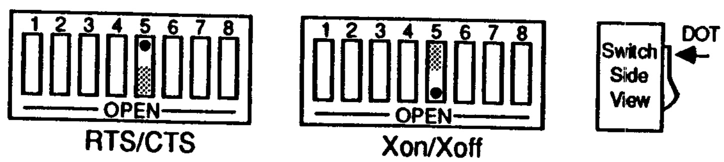
Switch SW1-5 is used to select between hardware **[RTS/CTS]** or software **[Xon/Xoff]** serial handshake control.

With **Xon/Xoff**, the **Serial488A** issues an **Xoff** character [ASCII value of $13] when its buffer memory is near full. When issued, there is greater than 1000 character locations remaining to protect against buffer overrun. When it is able to accept more information it issues an **Xon** character [ASCII value of $11]. The **Serial488A** also accepts **Xon/Xoff** on transmit from the serial host it is communicating with. **RTS/CTS** serial control becomes inactive when **Xon/Xoff** is enabled. The **RTS** output is, however, set to an active high state. The **CTS** input is not used for this handshake and may be left floating (unconnected).

With **RTS/CTS**, the **Serial488A** un-asserts **RTS** (low) when its buffer memory is near full. When un-asserted, there is greater than 1000 character locations remaining to protect against buffer overrun. When it is able to accept more information it asserts (high) **RTS**. The **Serial488A** will not transmit data to the serial host if it detects the **CTS** input un-asserted (low) when configured for this hardware handshake.

The factory default serial control is hardware, **RTS/CTS**.

SW1 View for Serial Handshake Selection



2.4 Terminator Selection

The **Serial488A** can be configured to provide RS-232 to IEEE 488 and IEEE 488 to RS-232 terminator substitution. This is useful when interfacing an RS-232 device which only issues carriage return [CR] as an output terminator to an IEEE controller which expects a carriage return followed by a line feed [CR-LF].

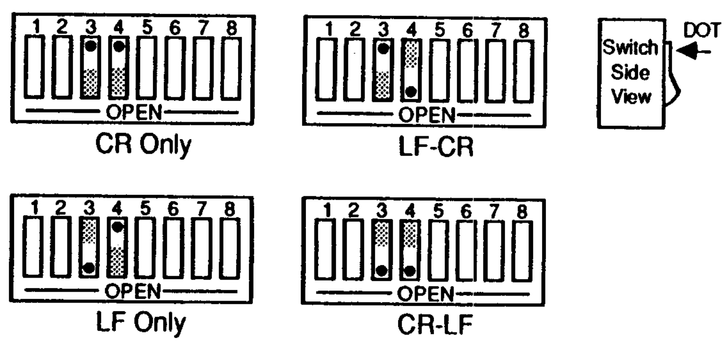
In the above example, the serial terminator should be selected for CR Only while the IEEE terminator is set to CR-LF. When a serial CR character is received, it is discarded and substituted with an IEEE CR-LF. In the IEEE to RS-232 direction, the IEEE CR is unconditionally discarded. Upon receipt of the IEEE LF, a serial CR is substituted.

The **Serial488A** can be made totally data transparent by setting both the serial and IEEE terminators to be CR Only or LF Only.

2.4.1 Serial Terminator Selection

SW2-3 and SW2-4 select the serial terminators for the serial input and output. The factory default is LF Only.

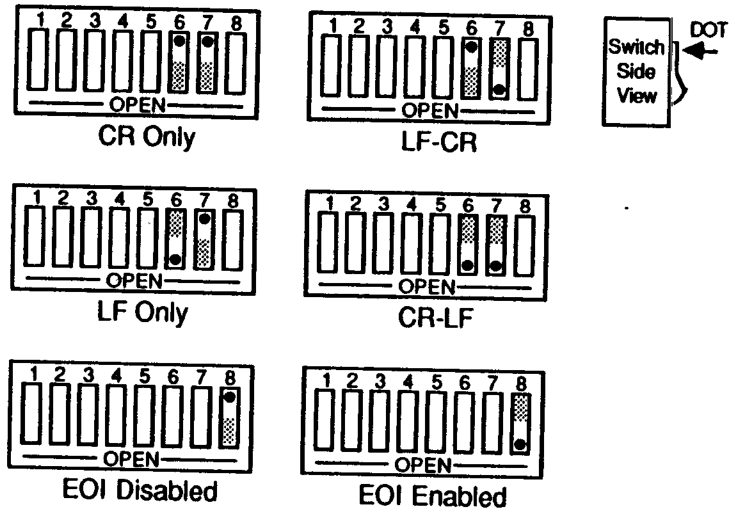
SW2 View for Serial Terminator Selection



2.4.2 IEEE Bus Terminator Selection

SW3-6 through SW3-8 set the IEEE bus terminators used for data sent or received by the **Serial488A**. EOI, a line used to signal the end of a multiple character bus transfer, may also be enabled. If enabled, EOI is asserted when the last selected bus terminator is sent. Factory default is LF Only with EOI disabled.

SW3 View for IEEE Bus Terminator Selection



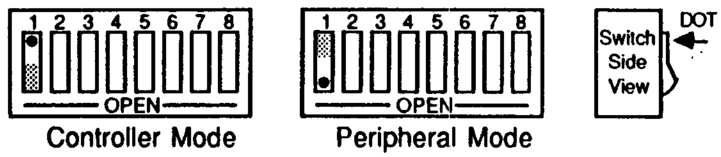
2.5 Mode Selection

SW2-1 sets the major operating mode of the **Serial488A.** The **IEEE Controller** (RS-232 to IEEE Converter) mode allows a serial host device to send data to a single IEEE bus peripheral. Applications include interfacing a listen-only or addressable IEEE printer/plotter to a serial printer port. Refer to **Section 3** for more detailed information on the controller mode of operation.

The **Peripheral** mode is used when interfacing a serial device to an IEEE controller. Data which is sent by the IEEE controller to the **Serial488A** is transmitted out its serial port. Data received from the serial device is buffered by the **Serial488A** until read by the IEEE controller. Refer to **Section 4** for more detailed information on the peripheral mode of operation.

The factory default is the **IEEE Controller** mode, an RS-232 to IEEE converter.

SW2 View for Mode Selection

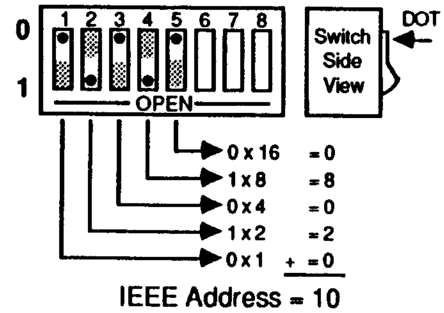


2.6 IEEE Address Selection

SW3-1 through SW3-5 select the IEEE bus address of the Serial488A when in the **IEEE Peripheral** mode. These same switches are used in the **IEEE Controller** mode to select the address of the device that will be controlled. [Refer to **Sections 4** and **3** respectively for additional information]. The address is selected by simple binary weighting with SW3-1 being the least significant bit and SW3-5 the most significant. The factory default is address 10.

Listen Only is a special type of **Peripheral** operation. In the **Listen Only** mode the **Serial488A** accepts all data transmitted on the bus, ignoring any bus addressing, and transfers it out its serial port. The **Serial488A** is set to **Listen Only** mode by setting its address to 31. If the IEEE address is set to 31 in the peripheral mode, it is adjusted to 30.

SW3 View for IEEE Address Selection



2.7 Feature Selections

The functions of the remaining switches are dependent on the mode selected. A brief description of each of these features follows. You should refer to the listed sections for additional information.

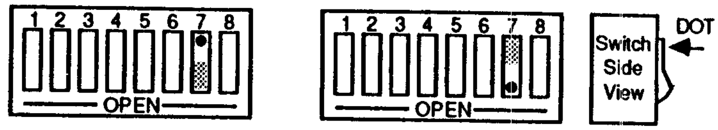
2.7.1 Controller Features

In the **IEEE Controller** (RS-232 to IEEE 488 Converter) mode, SW1-7 is used to determine whether the interface should, after sending the IEEE bus terminators, address the attached bus device to talk. The factory default is **Talk-back On Terminator** enabled.

SW2-2 selects whether the **Serial488A** should address the attached bus device to talk when the **Serial488A** has nothing more to send to that device. The factory default is **Talk-back On Time Out** enabled.

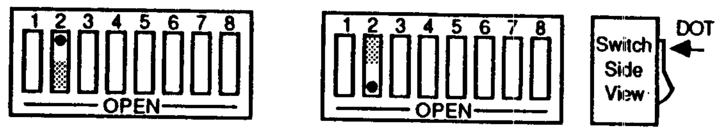
Refer to **Section 3** for complete details on these features.

SW1 View for Controller Talk-Back on Terminator Selection



Talk Back on Terminator Disabled — Talk Back on Terminator Enabled

SW2 View for Controller Talk-Back on Time-Out Selection

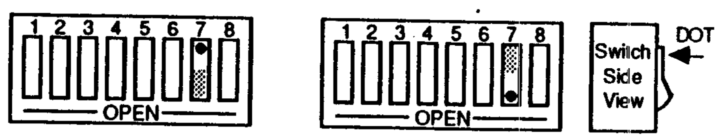


Talk Back on Time Out Disabled — Talk Back on Time Out Enabled

2.7.2 Peripheral Features

In the **IEEE Peripheral** (IEEE 488 to RS-232 converter) mode, SW1-7 enables the interface to assert the **SRQ** IEEE bus interface line to indicate that it has received the last switch selected serial terminator character from the serial device.

SW1 View for Peripheral SRQ on Last Serial Terminator



SRQ on Last Terminator Disabled — SRQ on Last Terminator Enabled

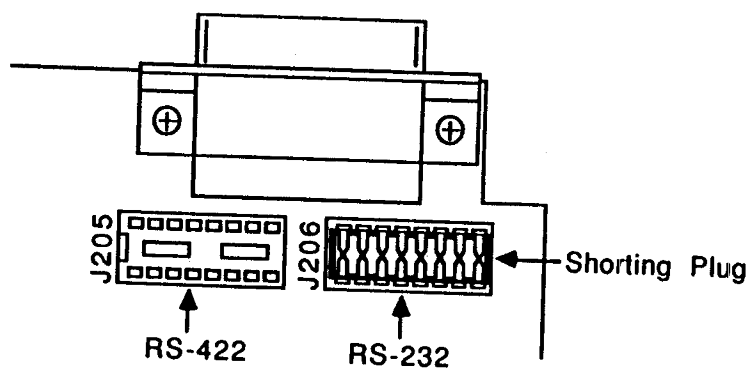
2.8 Serial Interface

The **Serial488A** and **Serial488A/OEM** have the ability to output signal levels that are compatible with either RS-232 or RS-422. An internal DIP shorting plug determines which electrical specification is chosen. If the interface is to be connected to an IBM PC/XT/AT/PS2 or compatible, the RS-232 level should be selected. If it will be connected to a Macintosh 512K/Plus/SE/II, the RS-422 level should be used. For connection to other computers, refer to the manufacturer's manual to determine which levels are supported.

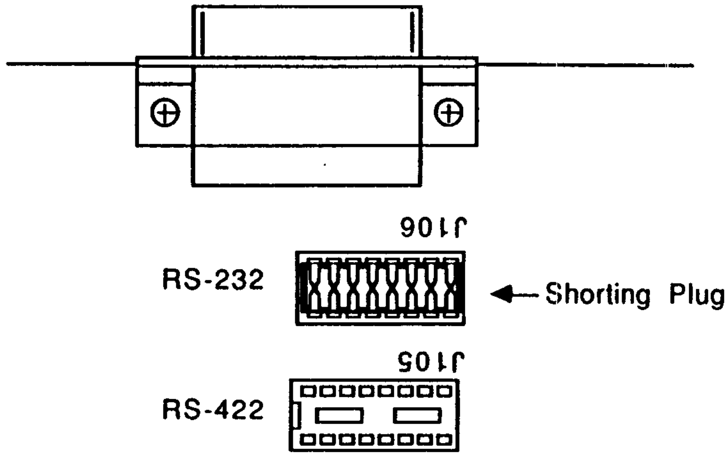
2.8.1 RS-232/RS-422 Signal Level Selection

The **Serial488A**'s and **Serial488A/OEM**'s factory default signal levels are compatible with RS-232. To select RS-422 levels, carefully remove the 8 position shorting plug with a small flat blade screwdriver from it's socket. Install the DIP jumper into the adjacent socket making certain that all of the pins on the shorting plug are inserted correctly. The following diagrams show which socket the jumper must be inserted for the desired operation.

RS-232 Signal Levels Selected - Serial488A



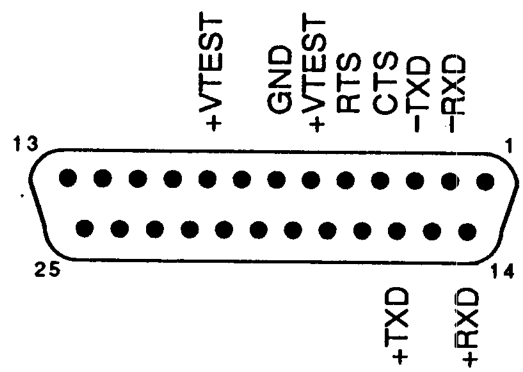
RS-232 Signal Levels Selected - Serial488A/OEM



2.8.2 Serial Signal Descriptions

The **Serial488A** is equipped with a standard DB-25S connector on its rear panel and requires a standard DB-25P mating connector. The **Serial488A**'s connector is configured as DCE type equipment for RS-232 communications, which means the **Serial488A** always transmits data on Pin 3 and receives data on Pin 2. The following list describes the RS-232 and RS-422 signals provided on the **Serial488A**. Note: since the **Serial488/512K** does not support RS-422 communication, the pins labelled +TxD and +RxD are not used. Any reference to RS-422 communication does not apply to the **Serial488/512K**.

Rear View of the Serial488A's Serial Connector



-RxD **Receive Data - Input - Pin 2**
This pin accepts serial data sent by the RS-232 or RS-422 host. The serial data is expected with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is low true.

-TxD **Transmit Data - Output - Pin 3**
This pin transmits serial data to the RS-232 or RS-422 host. The serial data is sent with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is low true.

CTS **Clear To Send - Input - Pin 4**
The **CTS** input is used as a hardware handshake line to prevent the **Serial488A** from transmitting serial data when the RS-232 host is not ready to accept it. When **RTS/CTS** handshake is selected on the internal switches, the **Serial488A** will not transmit data out **-TxD** while this line is un-asserted (low). If the RS-232 host is not capable of driving this line it can be connected to the **Vtest** output (Pin 6) of the **Serial488A**. If **Xon/Xoff** handshake is selected, the **CTS** line is not tested to determine if it can transmit data.

RTS **Request To Send - Output - Pin 5**
The **RTS** output is used as a hardware handshake line to prevent the RS-232/RS-422 host from transmitting serial data if the **Serial488A** is not ready to accept it. When **RTS/CTS** handshake is selected on the internal switches, the **Serial488A** will drive the **RTS** output high when there are greater than 1000 character locations available in its internal buffer. If the number of available locations drops to less than 1000, the **Serial488A** will un-assert (low) this output. If **Xon/Xoff** handshake is selected, the **RTS** line will be permanently driven active high.

Vtest **Test Voltage - Output - Pin 6**
This pin is connected to +5 volts through a 1KΩ resistor. It is also common to Vtest on pin 9.

Gnd **Ground - Pin 7**
This pin sets the ground reference point for the other RS-232 inputs and outputs.

Vtest **Test Voltage - Output - Pin 9**
This pin is connected to 5 volts through a 1KΩ resistor. It is also common to Vtest on pin 6.

+RxD **Receive Data Plus - Input - Pin 14**
This pin accepts serial data sent by the RS-422 host. The serial data is expected with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is high true and only connected to this pin when RS-422 operation is selected. It is 180° out of phase with **-RxD**.

+TxD **Transmit Data Plus - Output - Pin 16**
This pin transmits serial data to the RS-422 host. The serial data is sent with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is high true and only connected to this pin when RS-422 operation is selected. It is 180° out of phase with **-TxD**.

2.8.3 Serial Cable Wiring Diagrams

If a cable was not purchased with the interface, the following diagrams will be helpful in making your own cable. Simple soldering skills and an attention to detail will ensure successful construction.

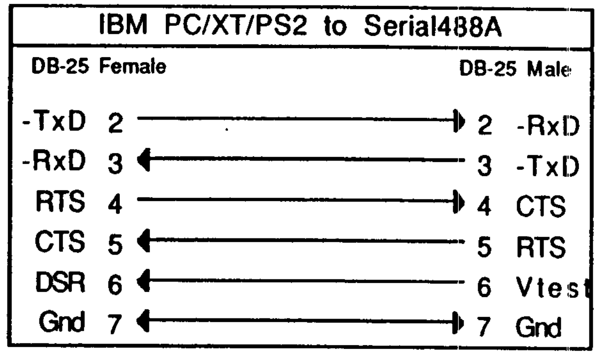
Macintosh to Serial488A Wiring Diagram (RS-422)

Macintosh to Serial488A				
DB-9 Male				DB-25 Male
RTS	6	→	4	CTS
CTS	7	←	5	RTS
-TxD	5	→	2	-RxD
Gnd	3	↔	7	Gnd
-RxD	9	←	3	-Txd
+TxD	4	↔	14	+Rxd
+RxD	8	←	16	+Txd

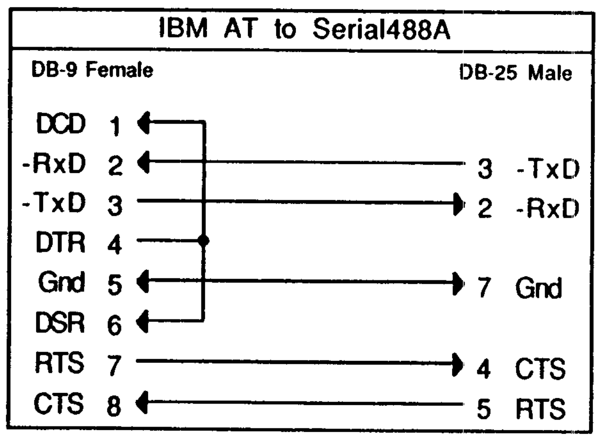
Macintosh Plus/SE/II to Serial488A Wiring Diagram (RS-422)

Macintosh II/SE/Plus to Serial488A				
Mini DIN8 Male				DB-25 Male
RTS	1	→	4	CTS
CTS	2	←	5	RTS
-TxD	3	→	2	-RxD
Gnd	4	↔	7	Gnd
-RxD	5	←	3	-Txd
+TxD	6	↔	14	+Rxd
+RxD	8	←	16	+Txd

IBM PC/XT/PS2 to Serial488A Wiring Diagram (RS-232)



IBM AT to Serial488A Wiring Diagram (RS-232)



Note: Standard AT 9 Pin to 25 Pin adapter cables are not wired as shown above and will not work with the **Serial488A**.

2.9 General Operation

Refer to the following sections for specific operational modes. This sub-section gives a general test of functionality. After setting the power on defaults and reassembling the **Serial488A**, plug the power supply connector into the rear jack on the interface.

CAUTION

Never install the power supply into the interface while it is connected to AC line power. Failure to observe this caution may result in damage to the Serial488A.

WARNING

The power supply provided with the interface is intended for INDOOR USE ONLY. Failure to observe this warning could result in equipment failure, personal injury or death.

After installing the power supply connector into the interface, plug the power supply into AC line power. Place the rear panel power switch in the ON **[1]** position. All the front panel indicators should light momentarily while the **Serial488A** performs an internal ROM and RAM self check. At the end of this self check all indicators except **POWER** should turn off.

If there is an error in the ROM checksum, all of the LEDs will remain on. Flashing LEDs indicates a RAM failure. Should such an error occur, turn the rear panel switch to the OFF **[0]** position and retry the above procedure.

When the **Serial488/512K** is first powered on, it performs a self test which lasts approximately 15 seconds. The front panel LED's will flash while the self test is performed. If the unit is functional, all LED's except power should turn off after the self test is completed. If one or more LED's remains flashing, refer to the Hardware Fault Identification Table to determine the cause of error.

Hardware Fault Identification Table - Serial488/512K

Error	Talk	Listen	Empty	Full	Power
No Error	off	off	on	off	on
No Error--Listen Only	off	on	on	off	on
No Power	off	off	off	off	off
Program Rom	on	on	on	on	on
Ram - U209	blink	off	off	off	on
Ram - U210	blink	off	on	off	on
Ram - U211	blink	off	off	on	on
Ram - U212	blink	off	on	on	on
Ram - U213	blink	on	off	off	on
Ram - U214	blink	on	on	off	on
Ram - U215	blink	on	off	on	on
Ram - U216	blink	on	on	on	on
Ram - U208	off	off	off	blink	on
Ram - U201	off	off	on	blink	on
Ram - U202	off	on	off	blink	on
Ram - U203	off	on	on	blink	on
Ram - U204	on	off	off	blink	on
Ram - U205	on	off	on	blink	on
Ram - U206	on	on	off	blink	on
Ram - U207	on	on	on	blink	on
Logic Error	blink	blink	blink	blink	on

If the front panel indicators do not flash and the **POWER** indicator does not remain lit there may not be any power supplied to the interface. In this event, check the AC line and the rear panel connection of the power supply for proper installation. If the problem is unresolved, refer to the **Service Information** section of this manual.

If proper operation is obtained, connect an interface cable to the rear of the **Serial488A** [25-Pin Sub-D]. Connect the other end to the host's serial port. Except for connecting IEEE bus instruments, the **Serial488A** is installed and ready to use.

WARNING

The Serial488A makes its earth ground connection through the serial interface cable. It should only be connected to IEEE bus devices after being first connected to the host. Failure to do so may allow the Serial488A to float to a bus device test voltage. This could result in damage to the interface, personal injury or death.

Controller Operation

3.1 Controller Mode (Serial to IEEE) Operation

The **IEEE Controller** mode allows a serial RS-232 or RS-422 host device to send data to a single IEEE bus peripheral or to multiple peripherals if they occupy the same bus address. Applications include interfacing a listen-only or addressable IEEE printer/plotter to a serial printer port.

Once the **Serial488A** has initialized itself after power-on, it waits for serial input data. When received, it addresses the selected IEEE device to listen with the following bus sequence:

ATN•UNL,MTA,LAG,*ATN

The data received from the serial host is placed into a circular serial input buffer. Simultaneously, characters are removed from that buffer and sent to the IEEE bus device. The serial terminator(s), if present, are not sent. Instead, the IEEE terminators are substituted and sent in their place.

So long as the serial input buffer is not empty, the **Serial488A** will continue to send data from it to the IEEE bus device. If the serial input buffer becomes emptied, the **Serial488A** will command the IEEE bus device to talk if one of the talk back features is enabled. This allows the **Serial488A** to be used as a controller with devices, such as plotters or instruments, that return status and other information to the host computer.

When the **Serial488A** addresses the IEEE bus device to talk it uses the following bus sequence:

ATN•UNL,MLA,TAG,*ATN

The **Serial488A** then accepts data from the IEEE device and returns it to the host until the last selected IEEE terminator is detected. The IEEE bus terminators are replaced by the serial terminators and these are then sent to the serial host.

If the IEEE device has been addressed to talk but does not respond or finish transmission by the time additional characters are received into the circular serial input buffer, the talk sequence will be aborted to allow additional serial information to be sent to the IEEE device.

3.2 Serial and IEEE Terminator Substitution

The **Serial488A** can be configured to provide serial to IEEE 488 and IEEE 488 to serial terminator substitution. This is useful when interfacing a serial host which only issues carriage return [CR] as an output terminator to an IEEE peripheral which expects a carriage return followed by a line feed [CR-LF].

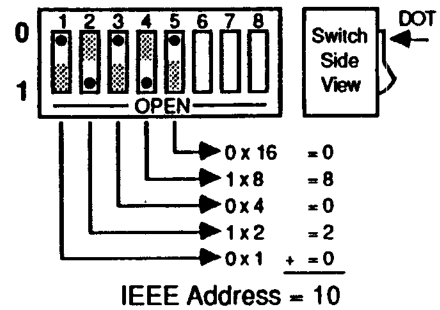
In this previous example, the serial terminator should be selected for CR Only while the IEEE terminator is set for CR-LF. When a serial CR character is received it is discarded and substituted with an IEEE CR followed by an IEEE LF. In the IEEE to serial direction, the IEEE CR is unconditionally discarded. Upon receipt of the IEEE LF a serial CR is substituted.

The **Serial488A** can be made totally data transparent by setting both the serial and IEEE terminators to be CR Only or LF Only. Refer to Section 2 for the proper switch settings for both the IEEE and serial terminators.

3.3 IEEE Address Selection

SW3-1 through SW3-5 select the IEEE bus address of the IEEE peripheral the **Serial488A** will be communicating with. These switches set the address of the IEEE device that will be controlled, not the address of the **Serial488A**. The address of the **Serial488A** is automatically adjusted so that address conflicts will not occur. The address is selected by simple binary weighting with SW3-1 being the least significant bit and SW3-5 the most significant. If address 31 (reserved on the IEEE bus) is selected in the controller mode, address 30 is assigned as the device it will be communicating with. The following figure shows the IEEE address selection of 10.

SW3 View for IEEE Address Selection



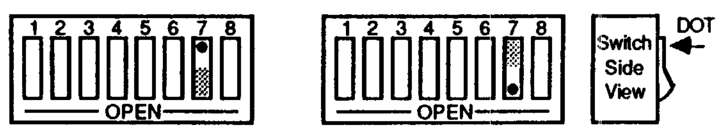
3.4 Talk Back Features

Two different switch selectable talk back features are included to provide bi-directional communication with the IEEE device. Whether either talk back feature should be enabled is dependent on the application.

3.4.1 Talk Back On Terminator

SW1-7 is used to determine whether the interface should address the attached bus device to talk after sending the selected IEEE bus terminator(s). This feature is commonly used to provide bi-directional communication with a single IEEE instrument. Talk back will only occur if there is no serial data to output to the IEEE device. The factory default is **Talk-back On Terminator** enabled.

SW1 View for Talk-Back on Terminator Selection



Talk Back on Terminator Disabled

Talk Back on Terminator Enabled

When the serial input buffer becomes empty, the **Serial488A** checks the last characters sent to the IEEE bus device. If these were the IEEE bus terminators and **Talk-Back on Terminator** is enabled, the IEEE bus device is addressed to talk. Any data received by the **Serial488A** from the bus device is sent to the serial host.

When the last IEEE bus terminator is detected from the IEEE device, the **Serial488A** disables the device from sending additional information by asserting Attention (ATN) on the bus.

If the IEEE device does not responded or finish transmission by the time additional characters are received into the serial input buffer, the talk sequence will be aborted to allow additional serial information to be sent to the IEEE device.

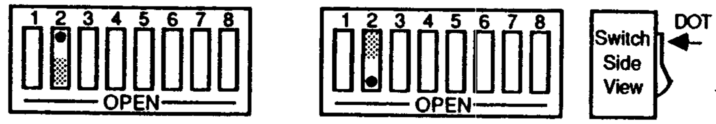
The following is an example of how this feature can be used to communicate with a single IEEE instrument. The program example is written in Basic on an IBM PC or compatible and communicates with a Keithley Model 196 DMM.

```
10   '
20   '      Example Program using Serial488A with
25   '  the Talk Back on Terminator Feature Enabled to
30   '   Communicate with a Keithley Model 196 DMM
40   '
50   ' Open Basic's serial communications port
60   OPEN "COM1: 9600,N,8,2" AS 1
70   ' Set the Model 196 DMM to the 30VDC range
80   PRINT #1,"F0R3X"; ' The ; suppresses terminators
90   ' Request 10 Readings from 196"
100  FOR N = 1 to 10
110     PRINT #1,"" ' Output terminator
120     LINE INPUT #1, A$ ' Get Reading from 196
130     PRINT A$ ' print it on the screen
140  NEXT N
150  END
```

3.4.2 Talk Back On Time Out

SW2-2 selects whether the **Serial488A** should address the attached bus device to talk when the **Serial488A** has no more serial data to send. This feature relies on time and not on terminators. Its use is primarily for simulating a serial plotter from an IEEE 488 [HP-IB] plotter. The factory default is **Talk-back On Time Out** enabled.

SW2 View for Talk-Back on Time-Out Selection



Talk Back on Time Out Disabled

Talk Back on Time Out Enabled

If **Talk-Back on Time-Out** is enabled, then **Serial488A** waits approximately 100 milliseconds after it detects its serial input buffer is empty. If no serial character has been received by the end of this time, the IEEE bus device is addressed to talk. The choice of talk-back modes depends strongly on the type of device and software being used. For most plotter applications the **Talk-back on Time-Out** feature should be enabled.

When the last IEEE bus terminator is detected from the IEEE device, the **Serial488A** disables the device from sending additional information by asserting Attention (ATN) on the bus. If the IEEE device does not respond or finish transmission by the time additional characters are received into the serial input buffer, the talk sequence will be aborted to allow additional serial information to be sent to the IEEE device.

Most IEEE 488 plotters will not respond to the talk address sequence with output data unless there has been a specific device dependent command sent to tell them what to say. If they have not been told what to say, they say nothing.

The following is an example of how this feature can be used to communicate with an IEEE plotter. The program example is written in Basic on an IBM PC or compatible. It turns the PC into a dumb serial terminal. When a key is pressed on the keyboard, the character is transmitted out of the serial (COM1) port. Any serial data which is received from the port is printed on the display.

```
10  '      Dumb Terminal Program for the Serial488A
20  ' This Program allows direct interaction between
30  '   the IBM-PC and an IEEE plotter through the
40  ' Serial488A. The Serial488A must have talk back
50  '            on time out enabled.
60  'Open the serial communications port
70  OPEN "COM1: 9600,n,8,2,cs,ds" AS 1
80  ' Display any data received from the COM1 port
90  IF LOC(1) THEN PRINT INPUT$(LOC(1),1);: GOTO 90
100'  Transmit key presses to the COM1 port and
screen
110 K$=INKEY$
120 PRINT #1,K$; : PRINT K$;
130 GOTO 90 ' Do it again
```

Enter the program into the computer and run it. The example below shows how to test the **Serial488A**'s operation with a Hewlett Packard 7470A plotter. Other IEEE plotters are similar but you should refer to the plotter's programming manual for the proper command syntax. Notice the **Serial488**'s front panel LEDs as you type the plotter commands.

Type the following HPGL output identify command on the keyboard.....

```
OI;
```

The plotter (HP 7470A) should immediately respond with.....

```
7470A
```

By typing the following HPGL command on the keyboard, the plotter should respond by retrieving its pen, drawing a line and returning the pen.

```
SP1;PA1000,1000;PD;PA1000,6000;PU;SP0;
```

3.5 Plotter Applications

To use the **Serial488A** to interface an HP-IB plotter to a serial computer port, you will need the following information about your system.

1. The serial data format that the application (plotting or graphics) program expects the plotter to communicate with. These parameters include baud rate, word length, stop bits, parity and serial control.

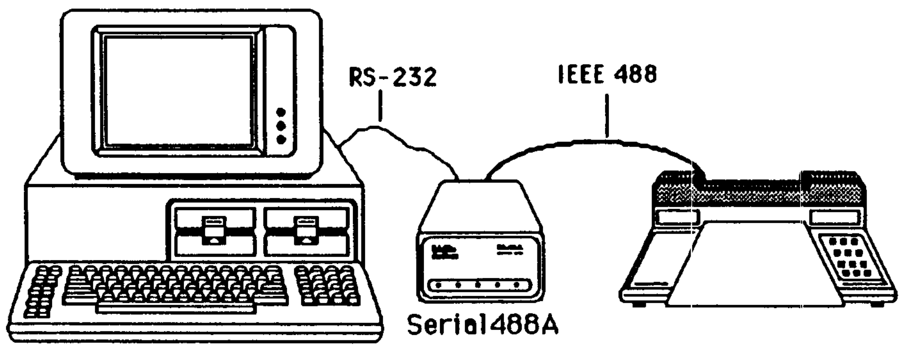
   Some programs allow these parameters to be selected by the user. Other graphics programs depend on the RS-232 version of the plotter defaults. Usually, Hewlett Packard plotters use 9600 baud, 7 data bits, 1 stop bit, even parity and Xon/Xoff serial control. Since these plotters are available with serial interfaces, the operator's manual of your IEEE plotter should contain this information.

2. The IEEE bus address of your plotter. This address is usually set by a DIP switch located on the rear of the plotter. The first five switches set the address which, for Hewlett Packard plotters, is usually address 5. Refer to the plotter's operator's manual for exact information.

Set the **Serial488A**'s internal DIP switches to match the parameters determined above. Other parameters which should be selected include...

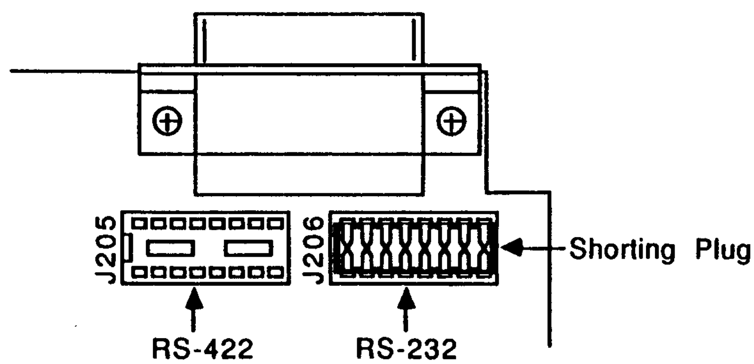
1. Talk Back on Terminator Enabled.
2. Talk Back on Time Out Enabled.
3. Serial Terminators set to CR Only.
4. IEEE Terminators set to CR Only with EOI enabled.

An IBM PC based Graphics System

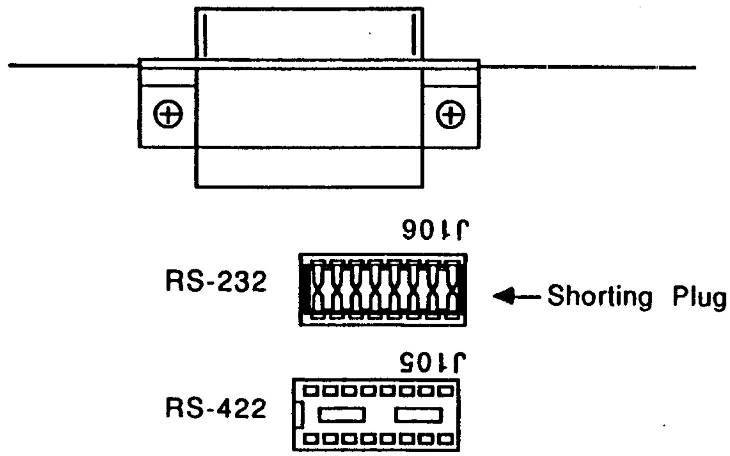


The following shows the **Serial488A**'s internal switch settings required to use a Hewlett Packard 7580A plotter with AutoCad™ from AutoDesk on an IBM PC or compatible. Because PC and compatibles output RS-232 levels, the shorting DIP jumper should be set to the RS-232 position (J206).

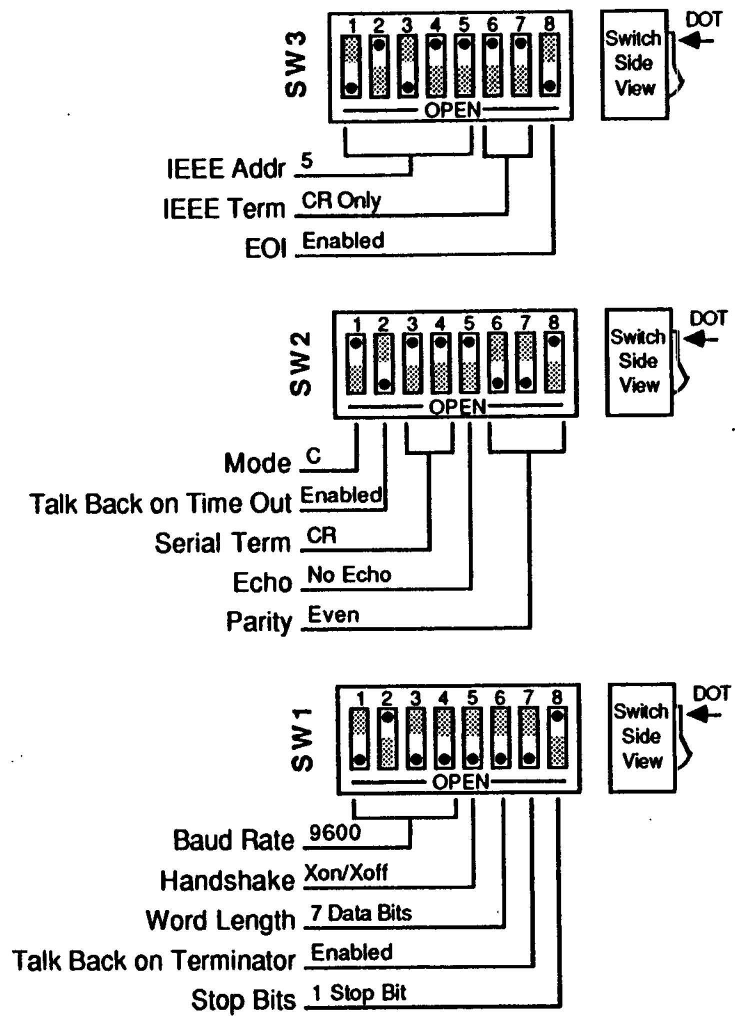
Selecting RS-232 Signal Levels - Serial488A



Selecting RS-232 Signal Levels - Serial488A/OEM



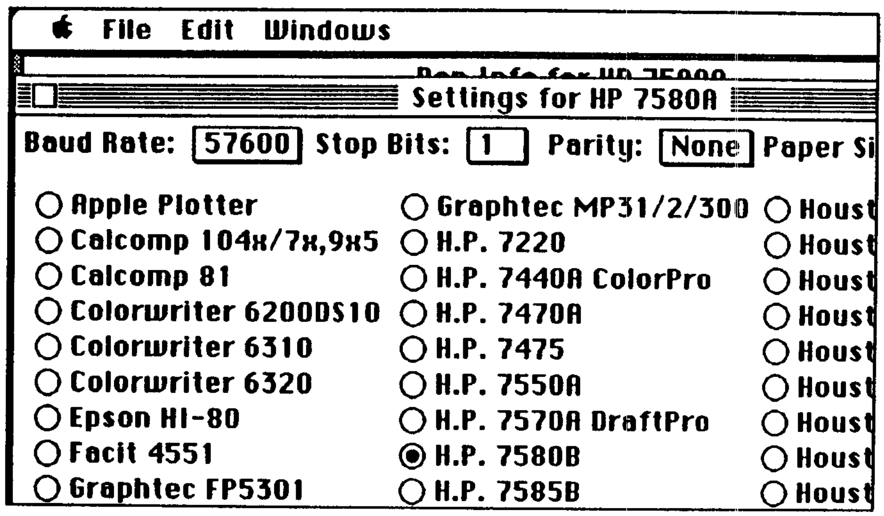
Serial488A Settings For Use With HP 7580A Plotter on an IBM PC



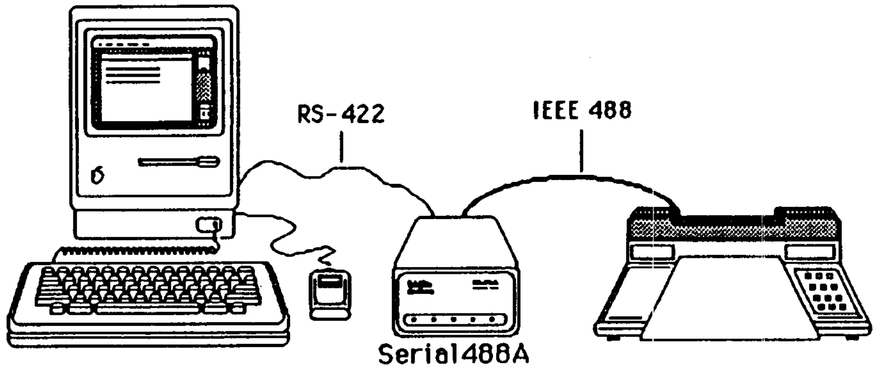
When using the **Serial488A** with plotting programs on the Macintosh™ computer with graphic drivers such as MacPlot™, some serial data format parameters are user modifiable. The following is a partial MacPlot configuration screen which allows selection of baud rate, stop bits and parity. With this driver, the word length is fixed to 7 data bits with Xon/Xoff serial control. These non-modifiable defaults are plotter dependent. Refer to the plotter or driver manual for the defaults of the specific plotter.

For this example, 57600 baud with one stop bit and no parity has been chosen for the serial data format.

MacPlot™ Configuration Screen

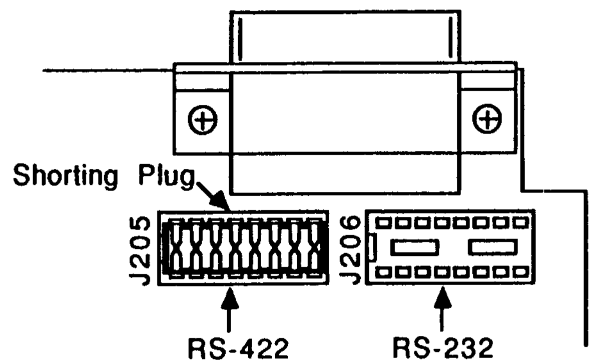


A Macintosh based Graphics System

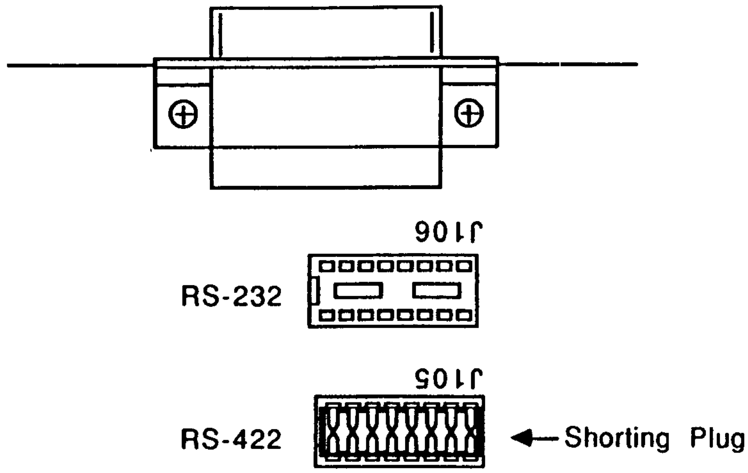


The Macintosh computer outputs RS-422 levels. Because of this, the internal DIP shorting jumper is set to the RS-422 position (J205).

Selecting RS-422 Signal Levels - Serial488A

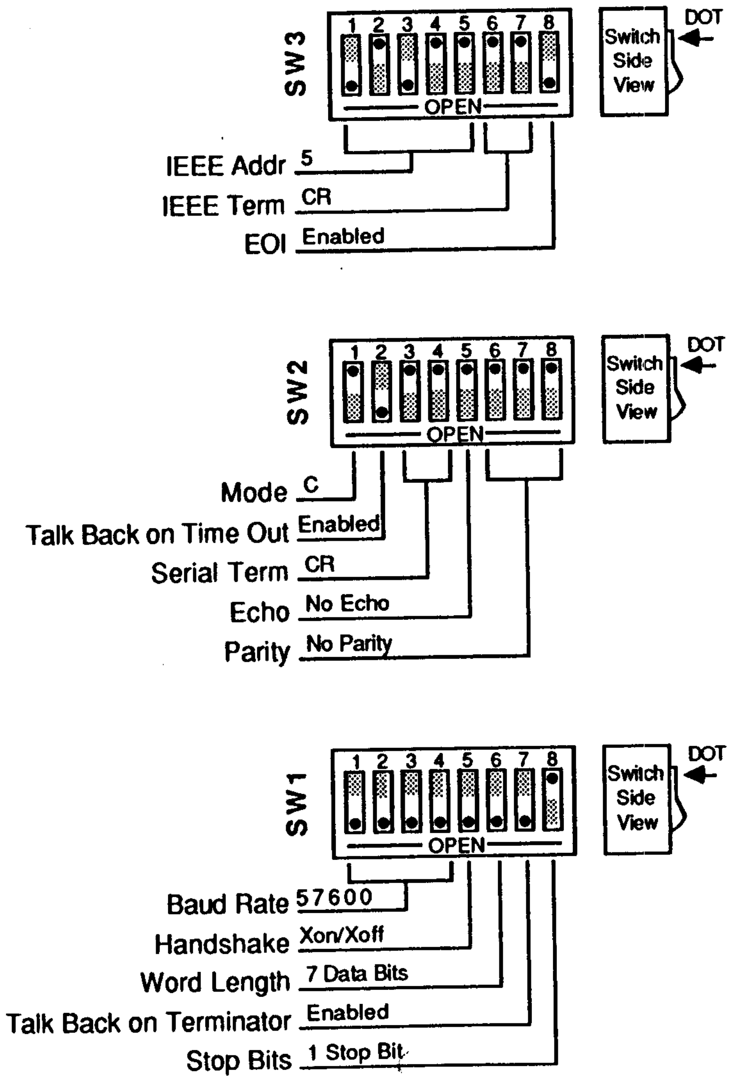


Selecting RS-422 Signal Levels - Serial488A/OEM



The following illustrates the **Serial488A**'s internal switch settings for use with MacPlot utilizing the previously described format.

Serial488A Settings For Use With HP 7580A Plotter on a Macintosh



After configuration, turn on the plotter and the **Serial488A**. The **Serial488A**'s front panel LEDs should all light momentarily while it performs an internal ROM and RAM test. All LEDs should go out except for the Power and Talk LED. The Talk LED indicates that the **Serial488** has detected the plotter on the IEEE bus and has addressed it to listen.

When the serial host begins to send the **Serial488** data, the Receive LED will flash. If it does not, this indicates that the interface is not receiving data from the serial host. Verify the cables are connected properly and the serial cable wiring. Verify the serial data format, word length, stop bits and parity.

3.6 Printer Applications

Most of the information given for plotter applications applies to applications for interfacing IEEE 488 printers to a serial host. Some high end printers have a secondary command setting which must be disabled for the **Serial488A** to control them. The **Serial488** does not use secondary commands to control IEEE peripherals, such as printers or plotters. Refer to the printer's instruction manual if the is a question as to whether the printer requires secondary commands.

Peripheral Operation

4.1 Peripheral Mode Operation

This mode of operation is useful in interfacing a serial device, such as a serial printer, plotter or instrument, to an IEEE controller. Data which is sent by the IEEE controller to the **Serial488A** is buffered and transmitted out its serial port. Data received from the serial device is buffered by the **Serial488A** until read by the IEEE controller. The **Serial488A** and the **Serial488A/OEM** can buffer approximately 32,000 bytes of data from both the IEEE input and the serial input. The **Serial488/512K** can buffer 512,000 bytes of data from both the IEEE input and the serial input.

The **Serial488A** will refuse to accept more data from the IEEE controller when its buffer memory is full. It does this by preventing completion of the bus handshaking sequences. It will also request that additional serial data not be sent by negating its Request To Send (**RTS**) output or by transmitting the **Xoff** ASCII character. The serial handshake used is dependent on the handshake selection (Refer to **Section 2**).

4.2 Serial and IEEE Input Buffers

Memory in the **Serial488A** is dynamically allocated for the serial input and IEEE input buffers. This allows for the most efficient partitioning of memory for any given application.

At power on, or device clear, each buffer is allocated a 128 byte mini-buffer or queue. When the serial input [or IEEE input] requires more buffer space, additional queues are allocated. When a queue is empty, it is released from the input buffers so that it may be re-allocated when, and where, required.

There are approximately 250 available queues in the **Serial488A** and the **Serial488A/OEM** for a total of 32,000 bytes of buffer (character) space. Queues are continually allocated and released as required by the serial and IEEE input. Of the 250 available queues, 240 are issued without regard to controlling the receipt of additional serial or IEEE input data.

When the serial input buffer requests one of the last 10 queues (1280 character locations left), it signals the serial host that it should stop sending data. This is accomplished by either un-asserting RTS or issuing "Xoff", depending on which serial handshake control has been switch selected. When more than 10 queues become available, it asserts RTS or issues "Xon".

The IEEE bus input signals that the IEEE input (or serial output) buffer is full when the number of queues available drops below 10 (1280 character locations left). When the number of available queues drops to 4 or less (512 character locations left), the IEEE interface of the **Serial488A** stops accepting data from the bus. This bus hold-off will only occur until additional queues (greater than 4) become available. At that time it will resume accepting bus data.

4.3 IEEE Data Transfers

The following methods may be used by the IEEE controller when sending data to the **Serial488A**:

4.3.1 Blind Bus Data Transfers

If the IEEE controller does not mind waiting an indefinite time for data space in the buffer to become available, the data can simply be sent to the **Serial488A**. This is referred to as blind data transfers because the IEEE controller is blind as to whether or not the **Serial488A** is capable of accepting data. In this case, the bus controller's output data transfer will be held off by the **Serial488A** if it is unable to buffer the data. It will resume accepting IEEE input data when memory becomes available. This type of control might be appropriate in a single user environment.

To illustrate how this would appear, let's assume the **Serial488A** is connected to a serial device which will accept data at 1200 baud or 110 bytes per second. The IEEE bus controller is capable of sending data to the **Serial488A** at a rate of 5000 bytes per second. The data would be transferred on the bus at 5000 characters per second for slightly over six seconds, filling over 31,000 locations. At that time, the IEEE input would hold off additional data transfers until 128 characters are sent out the serial port at rate of 110 characters per second. This 110 cps would then become the average bus data acceptance rate of the **Serial488A**.

If the controller is set to detect a data time-out error, then it will do so if the **Serial488A** holds off IEEE input data transfers for too long. The error can be used to alert the operator to the problem, such as a printer out of paper, so that it can be corrected. If the controller then restarts transmission exactly where it left off, no data will be lost.

If data is requested by the controller and no serial input data is available in the **Serial488A**, the bus will hang until serial data is received. If no serial data is received it will hang forever or until the controller times out.

4.3.2 Controlled Bus Data Transfers

If the controller must avoid waiting for the serial device, it can 'serial poll' the **Serial488A**. Serial poll is a method by which the controller can inquire the internal status of the interface without disturbing any data being transferred, slowing data transfers or locking up the bus. You should refer to the programming manual of your controller to determine the method of performing serial polls.

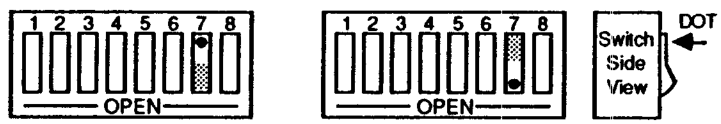
When serial polled, the **Serial488A** provides eight bits of status information to the controller. The most significant bit [DIO8] of the **Serial488A**'s serial poll byte is set to a logic "1" when the IEEE input buffer is NOT EMPTY. The term NOT EMPTY is used to signify that not all of the previous data sent to the interface has been transmitted to the serial device. If it is NOT EMPTY, the controller may avoid sending any more data to the **Serial488A**. If this bit is a logic "0", then the serial device has accepted all previous data and the IEEE controller may send more.

Another bit [DIO4] of the Serial Poll byte is used to indicate additional information concerning the IEEE input buffer. This bit is set to a logic "1" when there is 1280 or less locations in the buffer for data. It is cleared, set to a logic "0", when there is greater than 1280 locations available. This bit is referred to as the IEEE input buffer FULL bit.

When serial data is received, DIO5 of the Serial Poll byte is set, '1', to indicate to the IEEE controller that the serial input buffer is NOT EMPTY. If set, it indicates that at least one character is available in the serial input buffer to be read by the IEEE controller. Once all of the serial input data is read by the IEEE controller this bit is reset.

The **Serial488A** can generate a request for service on the bus when it receives the last serial terminator. To enable this feature, the **Peripheral SRQ** switch, located on the internal switch bank of SW1, must be enabled. When enabled, the **Serial488A** will assert the IEEE bus **SRQ** line and set serial poll status bits DIO7 and DIO3 when the last serial terminator is detected. The IEEE controller must perform a serial poll on the interface to clear the **SRQ**. If the **Peripheral SRQ** switch is in the disabled position, there will still be an indication in the serial poll status byte that the last serial input terminator was received, but **Serial488A** will not generate a service request (SRQ).

SW1 View For Selecting SRQ on Last Terminator



SRQ on Last Terminator Disabled

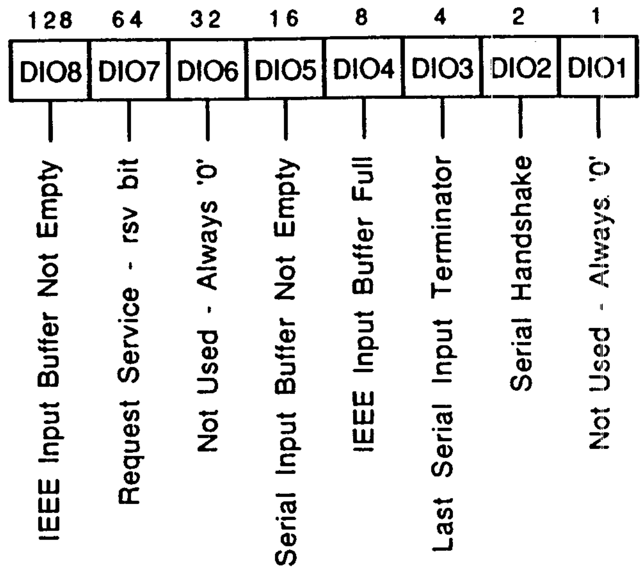
SRQ on Last Terminator Enabled

4.4 Serial Poll Status Byte Register

The following shows and describes the serial poll status information provided by the **Serial488A**.

DIO8 IEEE Input Buffer NOT Empty
This bit is set when the IEEE input buffer contains one or more data bytes which have not been sent out the serial port. It is cleared, set to "0", when the buffer is empty.

Serial Poll Status Byte



DIO7 rsv
This bit is defined by the IEEE 488 Specification and is used to indicate to the bus controller that the **Serial488A** is the bus device that requested service. It is cleared when the interface is serial polled by the controller.

DIO6 Not Defined - Always "0"

DIO5 Serial Input Buffer NOT EMPTY
This bit is set when the serial input buffer contains one or more data bytes which have not been sent out the IEEE bus. It is cleared, set to "0", when the buffer is empty.

DIO4 IEEE Input Buffer Full
When this bit is set, it indicates that the **Serial488A** may hold off the controller on subsequent data transfers. The interface may continue to accept an additional 512 characters but this is dependent on the serial input buffer size.

DIO3 Received Last Serial Terminator
This bit is set [1] when the **Serial488A** detects the last serial terminator at its serial input. It remains set as long as there is at least one serial terminator in the serial input buffer. If the **Peripheral SRQ** feature is enabled, the **Serial488A** will issue a request for service by asserting the **SRQ** line and also set the *rsv* bit [DIO7]. The *rsv* bit is cleared, along with the **SRQ** line, when serial polled by the controller. If there are additional serial terminators in the serial input buffer, **Serial488A** will reassert the **SRQ** line and *rsv* bit when the last IEEE 488 bus terminator is sent to the IEEE 488 controller.

DIO2 Serial Handshake
This bit indicates the present state of the serial handshake. If it is set to "1", the serial device connected to the **Serial488A** is capable of accepting serial data. If "0", the RTS line is unasserted, if configured for hardware handshake, or the "Xoff" character has been received, if configured for Xon/Xoff software handshake.

DIO1 Not Used - Always "0"

4.5 Use of Serial and Bus Terminators

The **Serial488A** can be configured to provide RS-232 to IEEE 488 and IEEE 488 to RS-232 terminator substitution. This is useful when interfacing a serial device, which only issues carriage return [CR] as an output terminator, to an IEEE controller, which expects a carriage return followed by a line feed [CR-LF].

In the previous example, the serial terminator should be selected for CR Only with the IEEE terminator set to CR-LF. When a serial CR character is received it is discarded and substituted with an IEEE CR followed by an IEEE LF. In the IEEE to serial direction, the IEEE CR is unconditionally discarded. Upon receipt of the IEEE LF, a serial CR is substituted.

The **Serial488A** can be made totally data transparent by setting both the serial and IEEE terminators to be CR Only or LF Only. The choice of appropriate terminators may be determined by inspection of the serial device and IEEE controller's instruction manuals. For selection of the **Serial488A**'s serial and bus terminators you should refer to **Section 2** of this manual.

4.6 IEEE 488 Bus Implementation

The **Serial488A** implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. The bus uniline and multiline commands that the **Serial488A** does not support or respond to include:

Remote Enable (REN)
Go to Local (GTL)
Group Execute Trigger (GET)
Local Lockout (LLO)
Take Control (TCT)
Parallel Poll (PP)
Parallel Poll Configure (PPC)
Parallel Poll Unconfigure (PPU)
Parallel Poll Disable (PPD)

4.6.1 My Talk Address (MTA)

When the **Serial488A** is addressed to talk, it retrieves data from the serial input buffer and outputs it to the IEEE 488 bus. It substitutes the selected IEEE bus terminators for the received serial terminators. The **Serial488A** will continue to output serial input buffer data as long as the IEEE controller allows.

4.6.2 **My Listen Address** (MLA)

When the **Serial488A** is addressed to listen, it accepts data from the active talker and outputs this data through the serial interface. It substitutes the selected serial terminators for the received IEEE bus terminators.

4.6.3 **Device Clear** (DCL and SDC)

Device Clear resets the **Serial488A**'s IEEE input and serial input buffers. Any pending data and Service Requests (SRQ), including the information they convey, are lost.

4.6.4 **Interface Clear** (IFC)

IFC places the **Serial488A** in the Talker/Listener Idle State. It clears any pending requests for service (SRQ). The condition which caused the SRQ remains unmodified.

4.6.5 **Serial Poll Enable** (SPE)

When Serial Poll Enabled, the **Serial488A** sets itself to respond to a serial poll with its serial poll status byte if addressed to talk. When the serial poll byte is accepted by the controller, any pending SRQs are cleared. The **Serial488A** will continue to try to output its serial poll response until it is 'Serial Poll Disabled' by the controller.

4.6.6 **Serial Poll Disable** (SPD)

Disables the **Serial488A** from responding to serial polls by the controller.

4.6.7 **Unlisten** (UNL)

UNL places the **Serial488A** in the Listener Idle State.

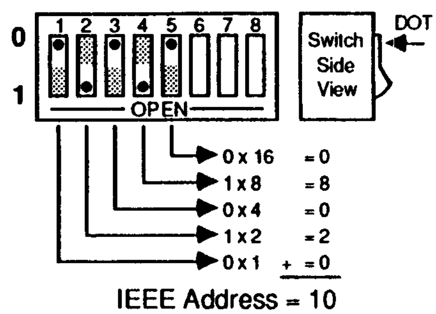
4.6.8 **Untalk** (UNT)

UNT places the **Serial488A** in the Talker Idle State.

4.7 IEEE Address Selection

SW3-1 through SW3-5 select the IEEE bus address of the **Serial488A** when in the **IEEE Peripheral** mode. The address is selected by simple binary weighting with SW3-1 being the least significant bit and SW3-5 the most significant. The following figure shows the IEEE address of the **Serial488A** set to 10.

SW3 View for IEEE Address Selection



4.7.1 Listen Only Mode

Listen Only is a special type of **Peripheral** operation. In the **Listen Only** mode the **Serial488A** accepts all data transmitted on the bus and transfers it out its serial port. The **Serial488A** is set to **Listen Only** mode by setting its address to 31 (switches SW3-1 through SW3-5 all open).

4.8 IEEE to Serial Applications

The following program uses a **Serial488A** as an interface to a serial instrument or host computer. The IEEE controller is an IBM PC running GWBasic with the IOtech Personal488™ controller package. Communications are provided under direct interaction from the keyboard.

In this program example, key presses are detected and sent via the IEEE bus to the **Serial488A**. The character is then sent to the serial device. Any incomming serial characters are buffered by the **Serial488A**. The **Serial488A** is polled by the controller for any data in the serial input buffer. When data is detected, it is read by the controller one character at a time and printed on the PC's screen. The IEEE address of the **Serial488A** is 10.

```
10  ' Open Driver488 Files and initialize
20  OPEN "\DEV\IEEEOUT" FOR OUTPUT AS 1
30  IOCTRL #1,"BREAK"
40  PRINT #1,"RESET"
50  OPEN "\DEV\IEEEIN" FOR INPUT AS 2
60  ' Look for PC Key Press
70  K$ = INKEY$
80  IF K$="" THEN GOTO 110
90  ' Output Key Press to Serial488A
100 PRINT #1,"OUTPUT 10;";K$;
110 ' Test for Serial data
120 PRINT #1,"SPOLL 10" : INPUT #2,SPOLL
130 IF NOT (SPOLL AND 16) THEN GOTO 70
140 ' Enter One Byte From Serial488A and print it
150  PRINT #1,"ENTER 10 #1"  :  S$  =  INPUT$(1,1)  :  PRINT
S$;
160 GOTO 120 ' Try for more
```

IEEE 488 Primer

5.1 HISTORY

The **IEEE 488** bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978. The **Serial488A** conforms to this most recent revision designated **IEEE 488-1978**.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the **IEEE 488** interface (sometimes called the General Purpose Interface Bus or **GPIB**) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the **IEEE 488** standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

5.2 GENERAL STRUCTURE

The main purpose of the **GPIB** is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the **System Controller** which is, at power on, the **Active Controller**.

The **System Controller** is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the **Active Talker**. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called **Addressing to Talk**. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the **GPIB**, this device becomes the **Active Controller**.

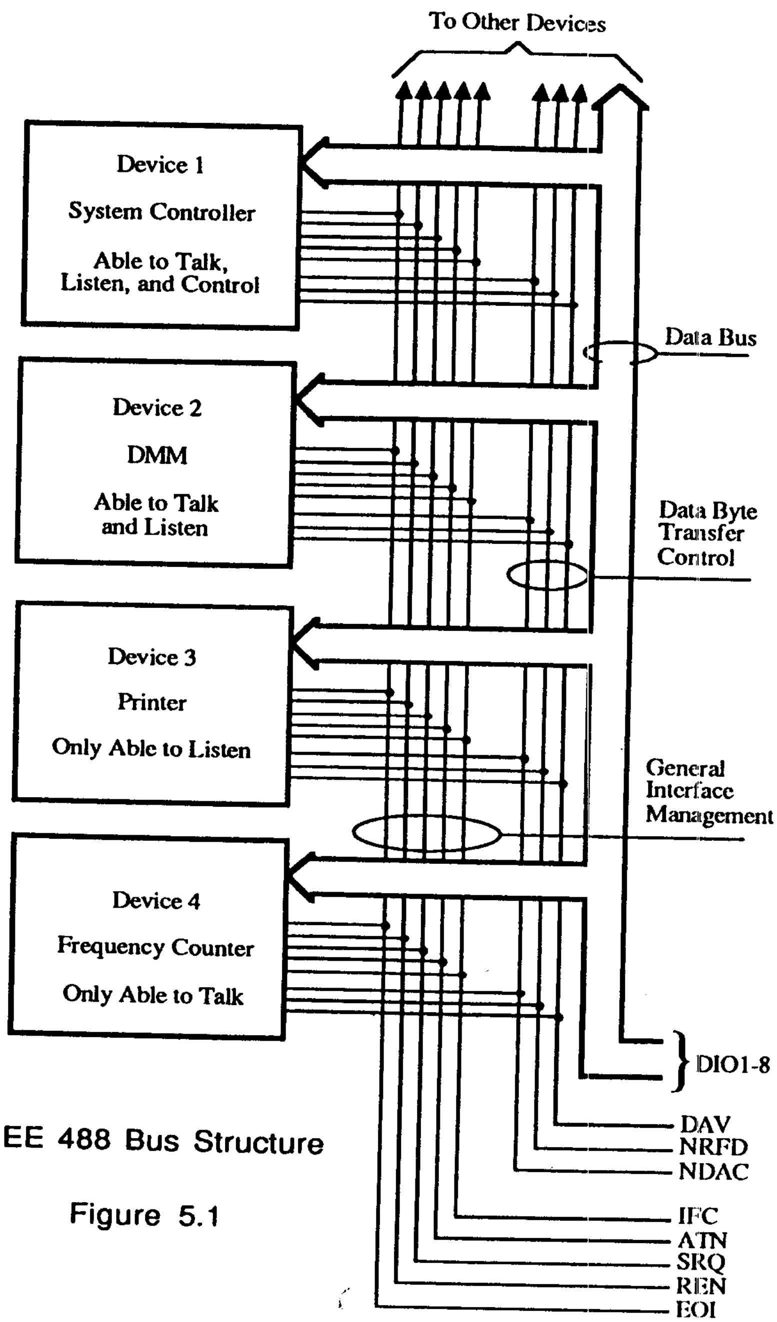
At a committee meeting, everyone present usually listens. This is not the case with the **GPIB**. The **Active Controller** selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being **Addressed to Listen**. This device is then referred to as an **Active Listener**. Devices which are to ignore the data message are instructed to **Unlisten**.

The reason some devices are instructed to **Unlisten** is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (**Unlisten**) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The **GPIB** transfers information in a similar way. This method of data transfer is called **handshaking**. More on this later.

For data transfer on the **IEEE 488**, the **Active Controller** must...

a) **Unlisten** all devices to protect against eavesdroppers.
b) Designate who will **talk** by **addressing** a device to **talk**.
c) Designate all the devices who are to **listen** by **addressing** those devices to **listen**.
d) Indicate to all devices that the data transfer can take place.



IEEE 488 Bus Structure

Figure 5.1

5.3 SEND IT TO MY ADDRESS

In the previous discussion, the terms **Addressed to Talk** and **Addressed to Listen** were used. These terms require some clarification.

The **IEEE 488** standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with **universal (multiline)** commands from the **Active Controller**. These commands include **My Listen Address** (MLA), **My Talk Address** (MTA), **Talk Address Group** (TAG), and **Listen Address Group** (LAG).

5.4 BUS MANAGEMENT LINES

Five hardware lines on the **GPIB** are used for bus management. Signals on these lines are often referred to as **uniline** (single line) commands. The signals are active low, i.e. a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

5.4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the **Active Listeners**. The **Active Controller** is the only bus device that has control of this line.

5.4.2 Interface Clear (IFC)

The **IFC** line is used only by the **System Controller**. It is used to place all bus devices in a known state. Although device configurations vary, the **IFC** command usually places the devices in the Talk and Listen Idle states (neither **Active Talker** nor **Active Listener**).

5.4.3 Remote Enable (REN)

When the **System Controller** sends the **REN** command, bus devices will respond to remote operation. Generally, the **REN** command should be issued before any bus programming is attempted. Only the **System Controller** has control of the **Remote Enable** line.

5.4.4 End or Identify (EOI)

The **EOI** line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts **EOI** during the transfer of the last data byte. The **EOI** signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The **Active Controller** also uses **EOI** to perform a **Parallel Poll** by simultaneously asserting **EOI** and **ATN**.

5.4.5 Service Request (SRQ)

When a device desires the immediate attention of the **Active Controller** it asserts **SRQ**. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a **Serial Poll** or a **Parallel Poll**.

5.5 HANDSHAKE LINES

The **GPIB** uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other **IEEE 488** lines, are active low.

5.5.1 Data Valid (DAV)

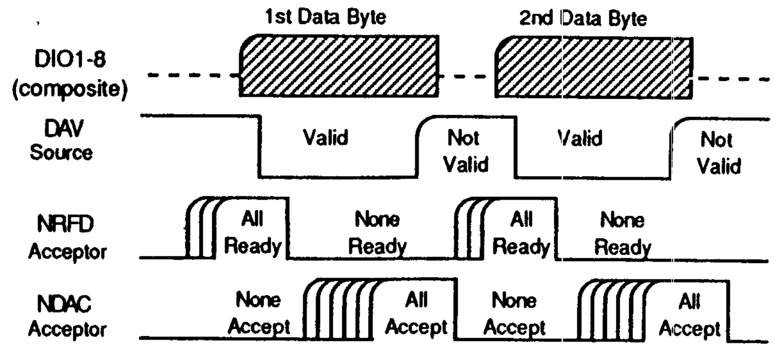
The **DAV** line is controlled by the **Talker**. The **Talker** verifies that **NDAC** is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The **Talker** then outputs data on the bus and waits until **NRFD** is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When **NRFD** and **NDAC** are in the proper state, the **Talker** asserts **DAV** (active low) to indicate that the data on the bus is valid.

5.5.2 Not Ready for Data (NRFD)

This line is used by the **Listeners** to inform the **Talker** when they are ready to accept new data. The **Talker** must wait for each **Listener** to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

5.5.3 Not Data Accepted (NDAC)

The **NDAC** line is also controlled by the **Listeners**. This line indicates to the **Talker** that each device addressed to listen has accepted the information. Each device releases **NDAC** (high) at its own rate, but the **NDAC** will not go high until the slowest Listener has accepted the data byte.



IEEE Bus Handshaking

5.6 DATA LINES

The **GPIB** provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of **DIO1** through **DIO8** instead of the binary designation of **D0** to **D7**. The data lines are bidirectional and are active low.

5.7 MULTILINE COMMANDS

Multiline (bus) commands are sent by the **Active Controller** over the data bus with **ATN** asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

5.7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

5.7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 001 while the 5 least significant bits are the address of the device being told to listen.

5.7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

5.7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 010 while the 5 least significant bits are the address of the device being told to talk.

5.7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

5.7.6 Local Lockout (LLO)

Issuing the LLO command prevents manual control of the instrument's functions. ($11)

5.7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a pre-defined or power up state. ($14)

5.7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a pre-defined or power up state. ($04)

5.7.9 Serial Poll Disable (SPD)

The **SPD** command disables all devices from sending their Serial Poll status byte. ($19)

5.7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after **SPE** is sent and **ATN** is unasserted. ($18)

5.7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

5.7.12 Take Control (TCT)

This command passes bus control responsibilities from the current **Controller** to another device which has the ability to control. ($09)

5.7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

5.7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a **Parallel Poll** as to which data bit they are to assert in response to a **Parallel Poll**. ($05)

5.7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a **Parallel Poll**. ($15)

5.8 MORE ON SERVICE REQUESTS

Most of the commands covered, both uniline and multiline, are the responsibility of the **Active Controller** to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, **SRQ** is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the **Controller** that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert **SRQ**. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might **SRQ** if its output has current limited. This is a powerful bus feature that removes the burden from the **System Controller** to periodically inquire, "Are you done yet?". Instead, the **Controller** says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since **SRQ** is a single line command, there is no way for the **Controller** to determine which device requested the service without additional information. This information is provided by the multiline commands for **Serial Poll** and **Parallel Poll**.

5.8.1 Serial Poll

Suppose the **Controller** receives a service request. For this example, let's assume there are several devices which could assert **SRQ**. The **Controller** issues an **SPE** (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the **Controller** that it was the device that asserted SRQ. Often times the other bits will indicate why the device wanted service. This **Serial Polling** sequence, and any resulting action, is under control of the software designer.

5.8.2 Parallel Poll

The **Parallel Poll** is another way the **Controller** can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When **SRQ** is asserted, the **Controller** (under user's software) conducts a **Parallel Poll**. The **Controller** must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a **Serial Poll** might be used to determine the why.

Of the two polling types, the **Serial Poll** is the most popular due to its ability to determine the who and why. In addition, most devices support **Serial Poll** only.

Service Information

6.1 Factory Service

If problems are encountered in using the **Serial488A** you should first telephone the factory. Many problems can be resolved by discussing the problems with our applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

6.2 Theory of Operation

The Heart of the **Serial488A** is a 6809 microprocessor [U101] supported by 8K bytes of firmware EPROM [U102 (2764)] and 32K bytes of static RAM [U103 (58256)]. A Versatile Interface Adapter [U104 (65B22)] is used to generate real-time interrupts for the firmware operating system.

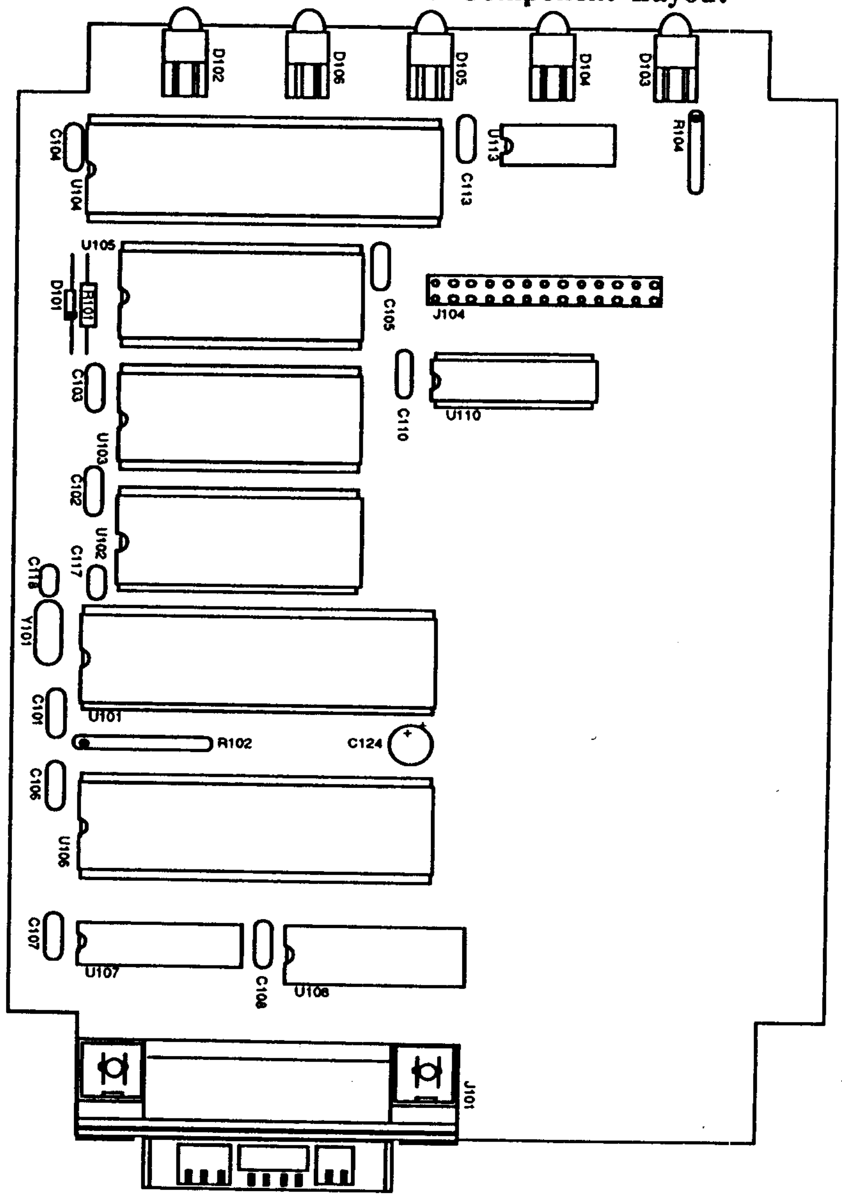
The front panel annunciators are also driven by U104 through an inverter [U113 (74LS04)]. The IEEE 488 bus interface is accomplished by a TMS9914A [U106] controller with drivers U107 and U108. The serial interface is provide by the UART [6551 (U105)]. If RS-232 levels are chosen, they are provided by the RS-232 transceiver (U209). If RS-422 levels are selected, the differential driver [26LS30 (U207)] and receiver [26LS33 (U208)] are used.

The internal DIP switches [SW1, SW2 and SW3] are read via 74HCT244 tri-state buffers [U201, U202 and U203]. Power is supplied by an external unregulated 9 volt wall mount supply. Regulation to the required +5 volts is provided by U206 [7805].

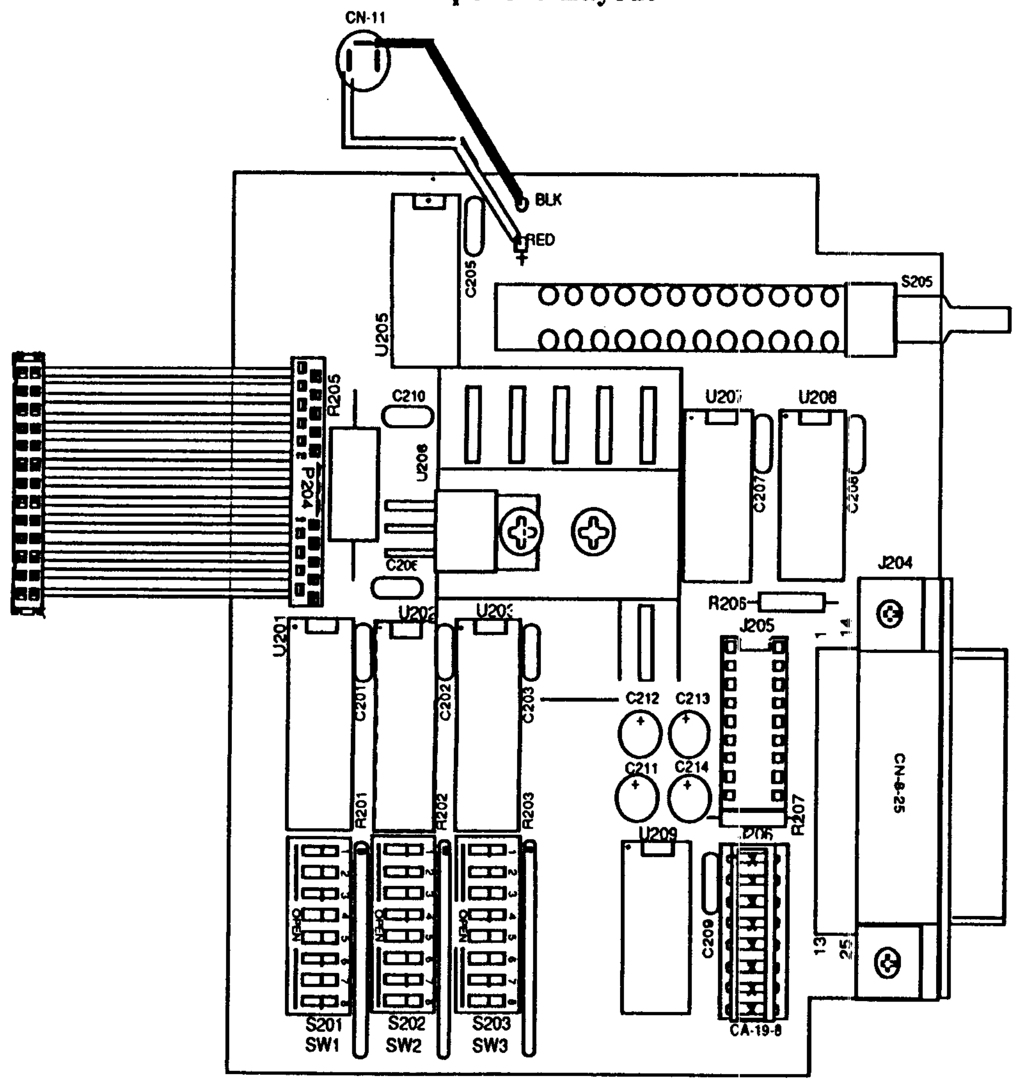
Decoding of the microprocessor address space is accomplished with a Programmable Logic Array [U110 (16L8)]. The Memory space allocation is

Address	Device	Part Number	Function
$0000-$7FFF	U103	58258	Static RAM
$A000-$A007	U106	9914A	IEEE Controller
$A800-$A807	U105	6551	UART
$B000-$B00F	U104	65B22	VIA
$B800	U201	74HCT244	SW1 (S201)
$B801	U202	74HCT244	SW2 (S202)
$B802	U203	74HCT244	SW3 (S203)
$E000-$FFFF	U102	2764	Programmed EPROM

6.3 Serial488A Mother Board Component Layout



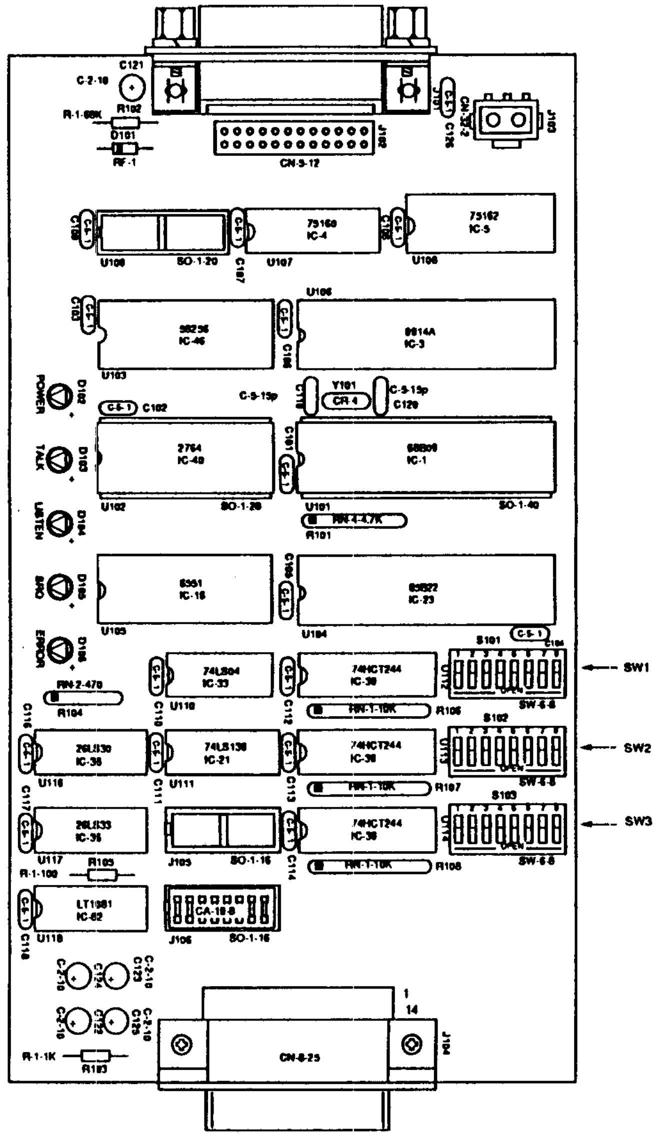
6.4 Serial488A I/O Board Component Layout



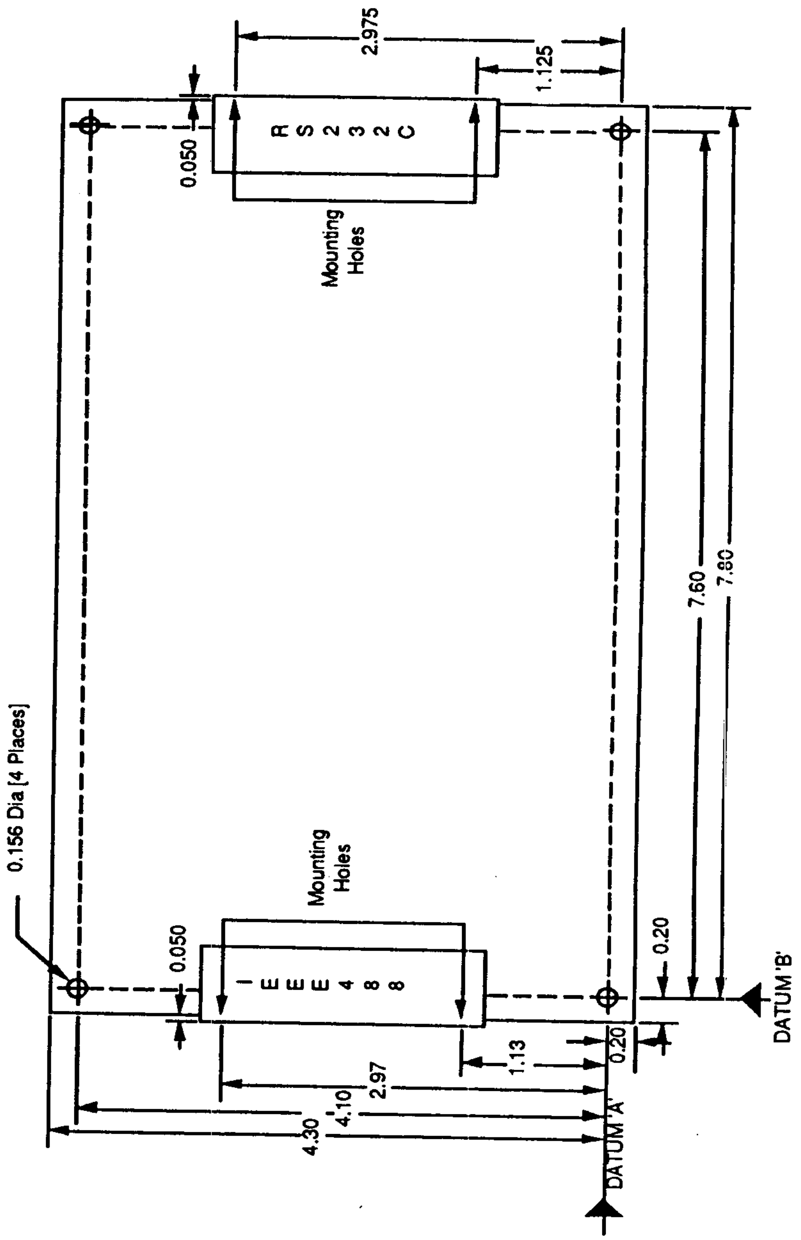
6.5 Replaceable Parts List - Serial488A

Schematic	Part Number	Description
C101-C108	C-5-.1	Ceramic, 25v
C110,C113	C-5-.1	Ceramic, 25v
C117,C118	C-5-15p	Ceramic, 25v
C124	C-2-10	Electrolytic, 25v
C211-C214	C-2-10	Electrolytic - 25v
C201-C203	C-5-.1	Ceramic, 25v
C205-C209	C-5-.1	Ceramic, 25v
C210	C-5-1	Ceramic, 25v
D101	RF-1	Small Signal Diode
D102-D106	DD-2	Red PC Mount
J101	CN-2	IEEE 488 Connector
J104	CN-5-13	13 x 2 0.1" Header
J201	CN-11	9 Volt Power Jack
J204	CN-8-25	PC Mount Female DB-25
J205	CA-19-8	8 Pos. DIP Jumper
P204	CA-6	26 Conductor Ribbon Assembly
R101	R-1-68K	68KΩ, 1/4w carbon
R102	RN-4-4.7K	4.7KΩ x 7 SIP
R104	RN-2-470	470Ω x 5 SIP
R201-R203	RN-1-10K	10KΩ x 9 SIP
R205	R-2-39	39Ω, 1w carbon
R206	R-1-100	100Ω, 1/4w carbon
R207	R-1-1K	1KΩ, 1/4w carbon
S201-S203	SW-6-8	8 Pole DIP
S205	SW-11	8 Pole DT Push Push
U101	IC-1	MC68B09P Microprocessor
U102	Serial488A-600	Programmed EPROM
U103	IC-78	84256-15 32K x 8 CMOS SRAM
U104	IC-23	65B22 Versitile Interface Adapter
U105	IC-16	R6551AP UART
U106	IC-3	TMS9914ANL IEEE Controller
U107	IC-4	SN75160BN IEEE Driver
U108	IC-5	SN75162BN IEEE Driver
U110	Macro488-601	Programming Equation - 16L8 PAL
U113	IC-33	74LS04 Hex Inverter
U201-U203	IC-39	74HCT244 Octal Buffer
U205	IC-21	SN74LS139 Dual Decoder
U206	IC-30	LM7805CT Regulator - +5v
U207	IC-38	26LS30 RS-423 Driver
U208	IC-36	26LS33 RS-422 Receiver
U209	IC-82	LT1081 RS-232 Transceiver
Y101	CR-4	7.3728 MHz Crystal

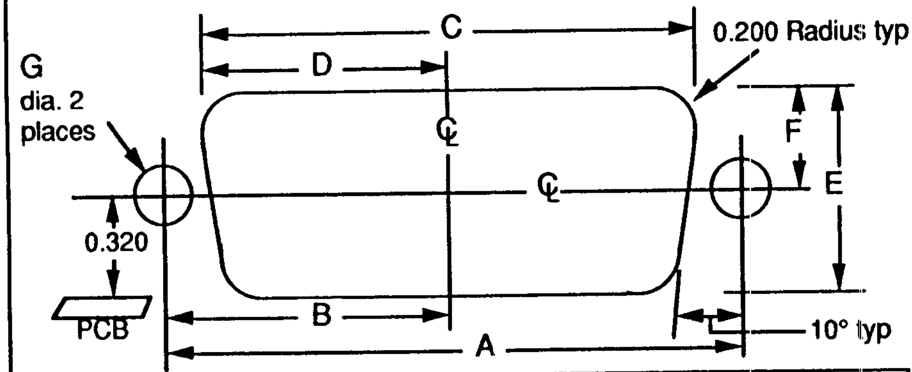
6.6 Serial488A/OEM Component Layout



6.7 Serial488A/OEM Mechanical Dimensions



IEEE Connector Mounting & Hole Punch Out



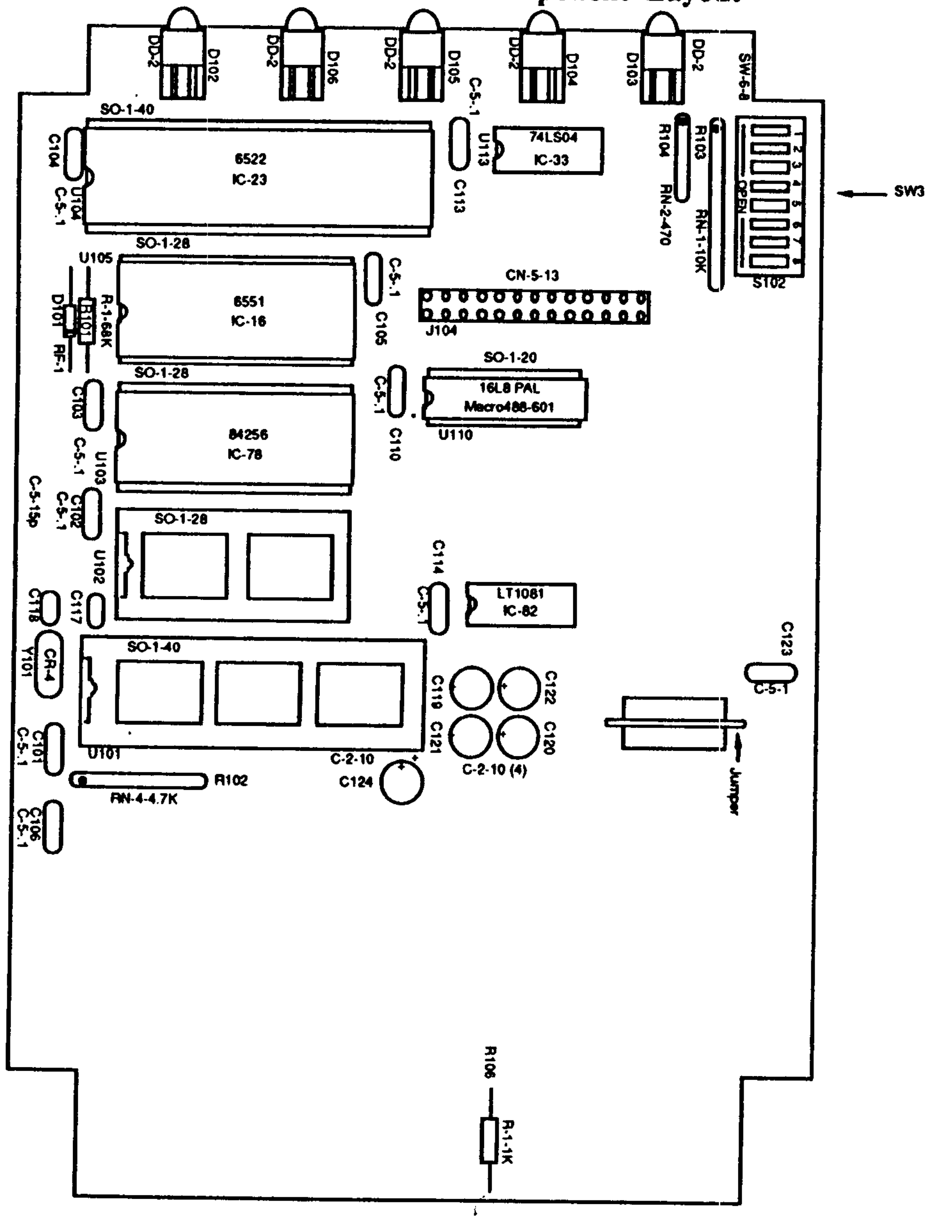
A	B	C	D	E	F	G
1.842	0.921	1.578	0.789	0.640	0.320	0.194

Note: Does not imply orientation

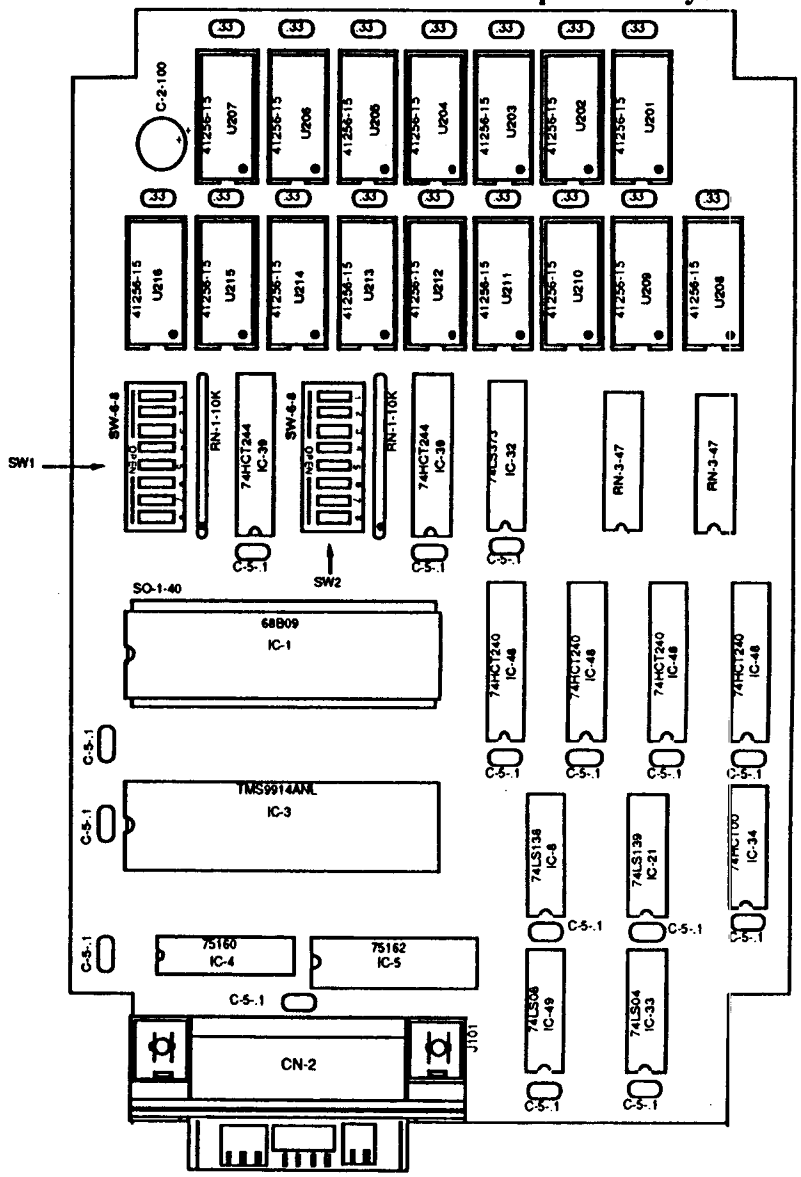
6.8 Replaceable Parts List - Serial488A/OEM

Schematic	Part Number	Description
C101-C114	C-5-.1	Ceramic, 25v
C116-C118	C-5-.1	Ceramic, 25v
C119,C120	C-5-15p	Ceramic, 25v
C121-C125	C-2-10	Electrolytic - 25v
C126	C-5-1	Ceramic, 25v
D101	RF-1	Small Signal Diode
D102-D106	DD-2	Red PC Mount
J101	CN-2	IEEE 488 Connector
J102	CN-5-12	12 x 2 0.1" Header
J103	CN-32-2	2 Position power connector
J104	CN-8-25	PC Mount Female DB-25
J105	CA-19-8	8 Pos. DIP Jumper
R101	RN-4-4.7K	4.7KΩ x 7 SIP
R102	R-1-68K	68KΩ, 1/4w carbon
R103	R-1-1K	1KΩ, 1/4w carbon
R104	RN-2-470	470Ω x 5 SIP
R105	R-1-100	100Ω, 1/4w carbon
R106-R108	RN-1-10K	10KΩ x 9 SIP
S101-S103	SW-6-8	8 Pole DIP
U101	IC-1	MC68B09P Microprocessor
U102	Serial488A-600	Programmed EPROM
U103	IC-78	84256-15 32K x 8 CMOS SRAM
U104	IC-23	65B22 Versitile Interface Adapter
U105	IC-16	R6551AP UART
U106	IC-3	TMS9914ANL IEEE Controller
U107	IC-4	SN75160BN IEEE Driver
U108	IC-5	SN75162BN IEEE Driver
U109	Macro488-601	Programming Equation - 16L8 PAL
U110	IC-33	74LS04 Hex Inverter
U111	IC-21	SN74LS139 Dual Decoder
U112-U114	IC-39	74HCT244 Octal Buffer
U116	IC-38	26LS30 RS-423 Driver
U117	IC-36	26LS33 RS-422 Receiver
U118	IC-82	LT1081 RS-232 Transceiver
Y101	CR-4	7.3728 MHz Crystal

6.9 Serial488/512K Motherboard Component Layout



6.10 Serial488/512K I/O Board Component Layout



6.11 Replaceable Parts List - Serial488/512K

Schematic	Part Number	Description
C101-C105	C-5-.1	0.1µF, 25v ceramic
C110	C-5-.1	0.1µF, 25v ceramic
C113,C114	C-5-.1	0.1µF, 25v ceramic
C117,C118	C-5-15p	15pF, ceramic
C119-C122	C-2-10	10µF, 25v electrolytic
C123	C-5-1	1µF, 25v ceramic
C124	C-2-10	10µF, 25v electrolytic
C201-C216	C-5-.33	0.33µF, 25v ceramic
C217-C232	C-5-.1	0.1µF, 25v ceramic
C233	C-2-100	100µF, 25v electrolytic
D101	RF-1	1N914 diode
D102-D106	DD-1	LED, Dialight #550-2406
J101	CN-11	Power Connector SWCR #712A
J102	CN-19-25	25 Pin Sub-D Female
J103	CN-2	IEEE Connector
J202-J203	CA-20	20 Circuit Jumper Cable
R101	R-1-68K	68KΩ, 1/4w, 10% carbon
R102	RN-4-4.7K	4.7KΩ x 7 SIP Network
R103	RN-1-10K	10KΩ X 9 SIP Network
R104	RN-2-470	470Ω X 5 SIP Network
R106	R-1-1K	1KΩ, 1/4w, 10% carbon
R201-R202	RN-3-47	47Ω X 8 DIP Network
R204,R205	RN-1-10K	10KΩ x 9 SIP Network
S101	SW-8	Power Switch
S102	SW-6-8	8 pole DIP switch
S201,S202	SW-6-8	8 pole DIP switch
U102	IC-10	2764-35 EPROM
U103	IC-78	84256 CMOS RAM
U104	IC-23	6522 VIA
U105	IC-16	6551 UART
U110	Macro488-601	Programmed 16L8 PAL
U113	IC-33	74LS04
U114	IC-82	LT1081

Schematic	Part Number	Description
U201-U216	IC-46	256K DRAM 150NS
U217	IC-50	74LS373 Octal Tri-State D
U218 ,U219	IC-14	74HCT244 Octal Buffer
U220-U223	IC-48	74HCT240 Inverting Buffer
U224	IC-1	MC68B09 Microprocessor
U225	IC-34	74HCTOO Quad NAND
U226	IC-21	74LS139 Dual Decoder
U227	IC-8	74LS138 Decoder
U228	IC-3	TMS9914ANL
U229	IC-33	74LS04 Hex Inverter
U230	IC-49	74LS08 Quad AND Gate
U231	IC-5	75162 IEEE Driver
U232	IC-4	75160 IEEE Driver
Y101	CR-5	8.000 MHz Crystal
	TR-4	Power Supply ; 115V AC
	TR-4E	Power Supply ; 220V AC

Sample Programs

```
10    REM ***   DUMB TERMINAL PROGRAM FOR THE Serial488A
20    REM ***             Running under IBM Basica
30    REM ***   This Program allows direct interaction between the
40    REM ***   IBM-PC and an IEEE bus device through the Serial488A.
50    REM ***   The Serial488A must be configured as the IEEE bus
60    REM ***   controller and have talk back on terminator enabled.
70    REM ***
80    REM ***                      IOtech, Inc.
90    REM ***
100       CLS
110   ' Open the serial communications port and set the serial parameters
120       OPEN "COM1: 9600,n,8,2,cs,ds" AS 1
130   ' Display any characters received from the COM1 port
140       IF LOC(1) THEN PRINT INPUT$(LOC(1),1);
150   ' Transmit any key presses from keyboard to the COM1 port and to the screen
160   K$=INKEY$
170   PRINT #1,K$; : PRINT K$;
180   GOTO 140 ' Do it again
```

$00 0 NUL	$10 16 DLE	$20 32 SP 00	$30 48 0 16	$40 64 @ 00	$50 80 P 16	$60 96 ` SCG	$70 112 p SCG
$01 1 SOH GTL	$11 17 DC1 LLO	$21 33 ! 01	$31 49 1 17	$41 65 A 01	$51 81 Q 17	$61 97 a SCG	$71 113 q SCG
$02 2 STX	$12 18 DC2	$22 34 " 02	$32 50 2 18	$42 66 B 02	$52 82 R 18	$62 98 b SCG	$72 114 r SCG
$03 3 ETX	$13 19 DC3	$23 35 # 03	$33 51 3 19	$43 67 C 03	$53 83 S 19	$63 99 c SCG	$73 115 s SCG
$04 4 EOT SDC	$14 20 DC4 DCL	$24 36 $ 04	$34 52 4 20	$44 68 D 04	$54 84 T 20	$64 100 d SCG	$74 116 t SCG
$05 5 ENQ PPC	$15 21 NAK PPU	$25 37 % 05	$35 53 5 21	$45 69 E 05	$55 85 U 21	$65 101 e SCG	$75 117 u SCG
$06 6 ACK	$16 22 SYN	$26 38 & 06	$36 54 6 22	$46 70 F 06	$56 86 V 22	$66 102 f SCG	$76 118 v SCG
$07 7 BEL	$17 23 ETB	$27 39 ' 07	$37 55 7 23	$47 71 G 07	$57 87 W 23	$67 103 g SCG	$77 119 w SCG
$08 8 BS GET	$18 24 CAN SPE	$28 40 (08	$38 56 8 24	$48 72 H 08	$58 88 X 24	$68 104 h SCG	$78 120 x SCG
$09 9 HT TCT	$19 25 EM SPD	$29 41) 09	$39 57 9 25	$49 73 I 09	$59 89 Y 25	$69 105 i SCG	$79 121 y SCG
$0A 10 LF	$1A 26 SUB	$2A 42 * 10	$3A 58 : 26	$4A 74 J 10	$5A 90 Z 26	$6A 106 j SCG	$7A 122 z SCG
$0B 11 VT	$1B 27 ESC	$2B 43 + 11	$3B 59 ; 27	$4B 75 K 11	$5B 91 [27	$6B 107 k SCG	$7B 123 { SCG
$0C 12 FF	$1C 28 FS	$2C 44 , 12	$3C 60 < 28	$4C 76 L 12	$5C 92 \ 28	$6C 108 l SCG	$7C 124 | SCG
$0D 13 CR	$1D 29 GS	$2D 45 - 13	$3D 61 = 29	$4D 77 M 13	$5D 93] 29	$6D 109 m SCG	$7D 125 } SCG
$0E 14 SO	$1E 30 RS	$2E 46 . 14	$3E 62 > 30	$4E 78 N 14	$5E 94 ^ 30	$6E 110 n SCG	$7E 126 ~ SCG
$0F 15 SI	$1F 31 US	$2F 47 / 15	$3F 63 ? UNL	$4F 79 O 15	$5F 95 _ UNT	$6F 111 o SCG	$7F 127 DEL SCG
ACG	UCG	LAG	LAG	TAG	TAG	SCG	SCG

ACG = Addressed Command Group
UCG = Universal Command Group
LAG = Listen Address Group
TAG = Talk Address Group
SCG = Secondary Command Group