4.0 Osborne 1 Interface Design

4.1 IEEE-488 INTERFACE

The IEEE-488 interface is created using a 6821 PIA. The IEEE-488 implementation, as described in the Osborne 1 User Guide's Appendix, is a subset of the complete IEEE specification. Specifically, no provision has been made for controlling multiple devices on the interface.

4.1.1 IEEE-488 Signal Direction

The Osborne 1 IEEE-488 signal directions are provided here:

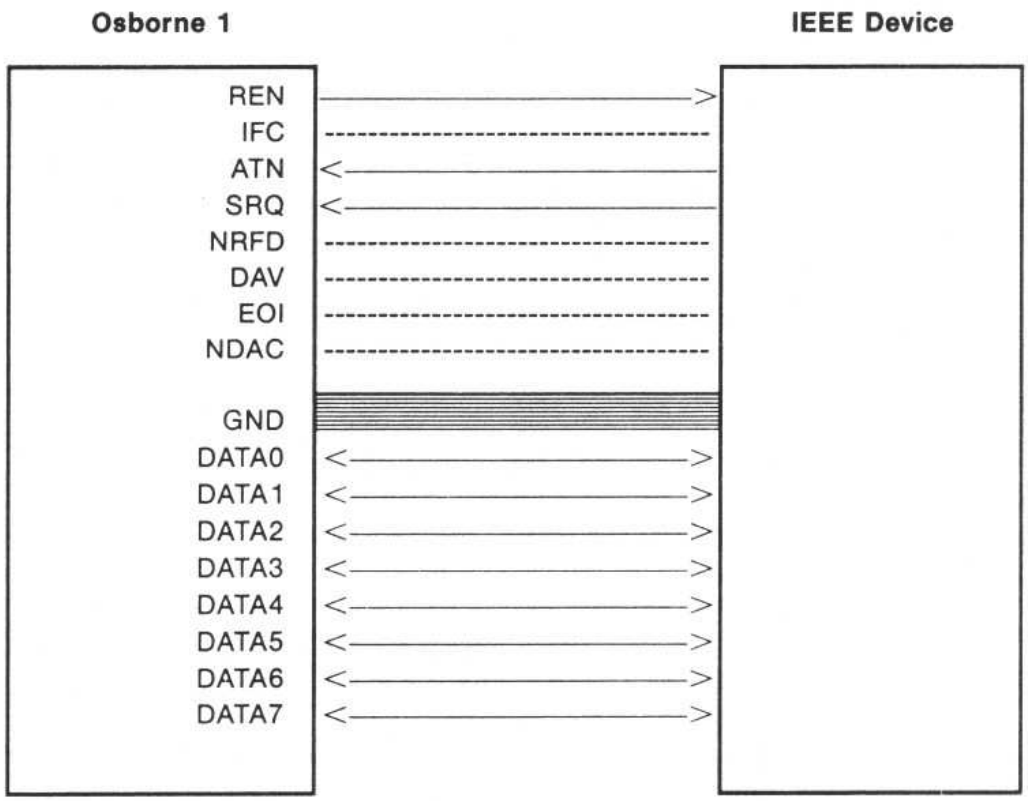


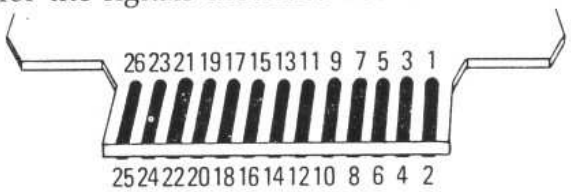
Figure 4.1.1 IEEE-488 Signal Direction

4.1.2 IEEE-488 Pinouts

Any IEEE 488-compatible device can connect to the Osborne 1 through the IEEE connector. Because this port is used for more than just IEEE 488 signals, we've declined to use a standard IEEE connector. The following table shows the pin assignments for both the IEEE standard connector and the Osborne 1 edge connector:

IEEE	OSBORNE	SIGNAL NAME	
1	1	Data bit 1	(DIO1)
2	3	Data bit 2	(DIO2)
3	5	Data bit 3	(DIO3)
4	7	Data bit 4	(DIO4)
5	9	End or Identify	(EOI)
6	11	Data valid	(DAV)
7	13	Not ready for data	(NRFD)
8	15	No data accepted	(NDAC)
9	17	Interface clear	(IFC)
10	19	Service request	(SRQ)
11	21	Attention	(ATN)
12	23	Cable shield + GND	(SHIELD)
13	2	Data bit 5	(DIO5)
14	4	Data bit 6	(DIO6)
15	6	Data bit 7	(DIO7)
16	8	Data bit 8	(DIO8)
17	10	Remote enable	(REN)
18	12	Signal ground	(DAV)
19	14	Signal ground	(NRFD)
20	16	Signal ground	(NDAC)
21	18	Signal ground	(IFC)
22	20	Signal ground	(SRQ)
23	22	Signal ground	(ATN)
24	24	Signal ground	(Logic)

The pinouts for the signals described above are as follows:



26-pin edge connector, looking at front of Osborne 1. Pins 25 and 26 are not used, but provided for compatability.

Figure 4.1.2 IEEE-488 Pinouts

4.1.3 IEEE-488 Jump Vectors

To provide easier access to the routines necessary to use the IEEE-488 interface, the CP/M BIOS jump table has been extended to provide a series of extra jumps specifically for the IEEE-488 programmer. The IEEE-488 routines are offset from the starting address of BIOS as follows:

BIOS + 3FH Control out
BIOS + 42H Status in
BIOS + 45H Go to standby
BIOS + 48H Take control
BIOS + 4BH Output interface message
BIOS + 4EH Output device message
BIOS + 51H Input device message
BIOS + 54H Input parallel poll message

4.1.4 IEEE-488 Communication Protocol

IEEE-488 commands use no RAM other than the stack. Each command routine in BIOS determines status of the port by reading the status of the 6821 PIA chip. The PIA transmits signals in both directions, so to reduce the overhead in determining the current direction the PIA is attempting to communicate, it is always left in one of two modes:

the source handshake mode
or
the acceptor handshake mode

(The PIA specification sheet will be helpful in determining these modes.)

Several of the IEEE commands require that the PIA be in the source handshake mode when called. The PIA is normally in the source handshake mode following the completion of any IEEE- bus information transfer, so this is not a major restriction. For instance, both the Status In and the Parallel Poll commands require that the PIA be in the source mode, which means that you can perform the detection-of-device request using either serial poll or parallel poll only when the interface is idle.

To send data to a device on the IEEE bus, the controller makes the device a LISTENER, assumes the role of TALKER, and sends the data. To receive data from an external device, the controller must first make the device a TALKER and then assume the role of LISTENER. After this, the controller goes on "standby" and allows the two devices to communicate at their own rate.

The controller can regain control asynchronously by setting the ATN signal to true. But if a device-dependent message is true at the same time when ATN becomes true, other devices on the IEEE bus can misinterpret the interrupted

byte as an interface message and produce chaos. Avoid the problem by taking control synchronously. If high-speed transfer of data between devices is not required and the computer can be tied up during the transfer, it is better to make the controller listen to the transfer while discarding the data. This procedure allows the controller to count transfers, look for EOI signals, or "time out" the TALKER before regaining control.

The IEEE commands are detailed in the User's Guide's Appendix, with sample programs included to help decipher how we've put the BIOS jumps into effect for the IEEE bus. A listing of the 6821 registers and instruction set is provided as Appendix B of this manual for those who wish to make direct use of the PIA for controlling the IEEE-488 port.

4.1.5 IEEE-488 As A Parallel Port

The IEEE-488 can also be used as a standard parallel port, and software has been added to the BIOS section of CP/M so that users of a Centronics-compatible printer may use their printer as the list device under CP/M. By setting the CP/M IOBYTE equal to BAT:, UR1:, UP1:, or LPT:, the IEEE-488 port is reconfigured by BIOS to be a simple 8-bit parallel input/output port with the following pinouts:

Osborne IEEE Edge Connector		Centronics-Compatible Connector
pin 1	data 0	2
2	data 4	6
3	data 1	3
4	data 5	7
5	data 2	4
6	data 6	8
7	data 3	5
8	data 7	9
11	out strobe	1
12	ground	19
15	busy	11
16	ground	29
19	select	13

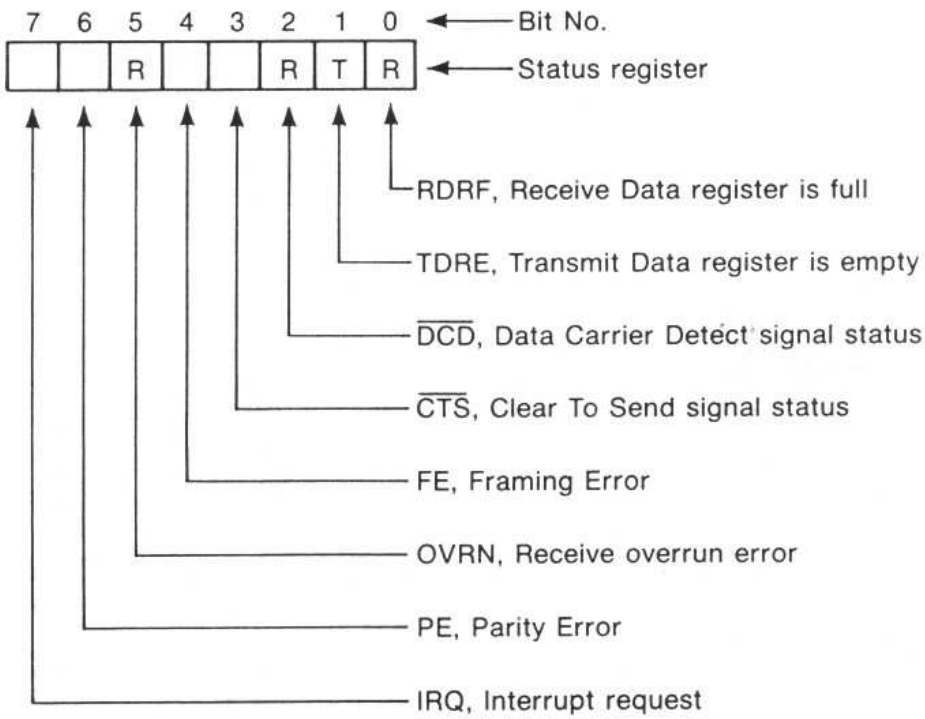
Figure 4.1.5 IEEE-488 Parallel Interface

To use a parallel printer connected to the IEEE interface, you must employ the SETUP program to configure the appropriate protocol for the printer. You need do this only once for each diskette. Alternatively, if you have two printers hooked up, or wish to change the printer being used from within a program, you can reset the IOBYTE as described later.

4.2 SERIAL RS232 INTERFACE

The serial port is configured as a RS-232C-compatible port, though certain of the RS-232C signals are held at +5 volts since they are not needed to control the Osborne 1. A 6850 ACIA chip controls the serial port.

The RS232 status port address is located at 2A00H and the data port at 2A01H in the shadow mode. RS232 status bit assignments are detailed in the following diagram:



NOTE: See pages 9-59 and 9-60 in Volume 2 of An Introduction To Micro computers by Adam Osborne/McGraw-Hill or the 6850 Data sheet for a complete description.

Figure 4.2 RS-232 Status Bit Assignments

4.2.1 RS232 Signal Direction

The serial port is configured as a DTE device. The following signals apply:

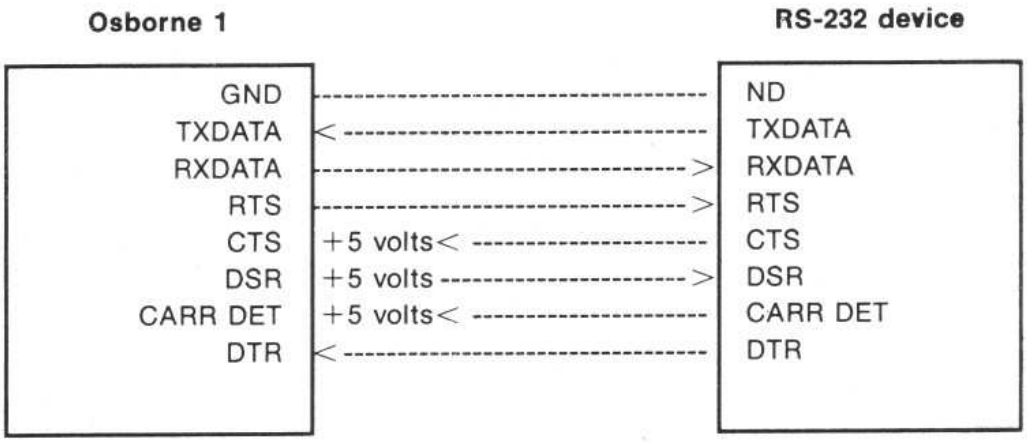
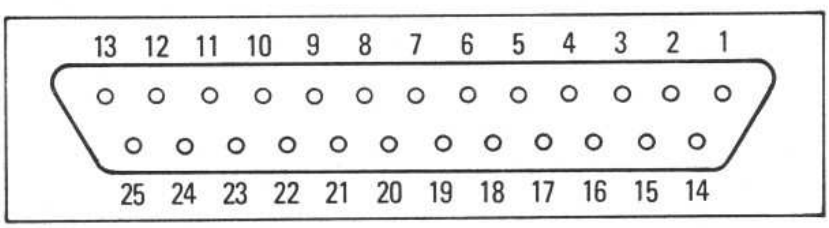


Figure 4.2.1 RS-232 Signal Direction

4.2.2 RS232 Pinouts

Below are the pin assignments for the RS-232 serial interface:

DB-25S	RS-232	Pin Definition
1	AA	Frame ground (optional)
2	BA	Transmitted data (low=1)
3	BB	Received data (low=1)
4	CA	Request to send (high or no connection enables)
5	CB	Clear to send (always high on OCC 1)
6	CC	Data set ready (always high on OCC 1)
7	AB	Signal ground
8	CF	Received line signal detected (always high)
20	CD	Data terminal ready (high or no connection enables)



9,10,11,12,13,14,15,16,17,18,19,21,22,23,24,25 no connections

Figure 4.2.2 RS-232 Serial Pinouts

4.3 MODEM

A close look at the circuitry in the Osborne 1 schematics will show that the modem and RS-232 interfaces are basically one and the same. In addition to the serial port, TTL-level signals may be directly input into the 6850 ACIA using the modem port connection. To read to and from the modem or serial port use the CP/M IOBYTE function.

4.3.1 Modem Signal Direction

The following signals apply to the Modem Port:

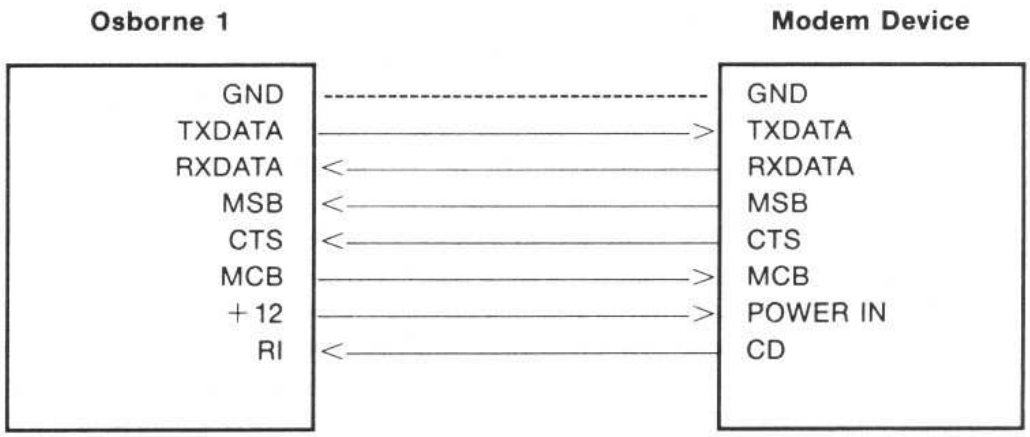


Figure 4.3.1 Modem Signal Direction

4.3.2 Modem Pin Connections

The pin connections on the modem port are as follows (all use the standard numbering of the DE-9P connector).

DE-9P Osborne Modem Definition:

1 **GND** – Signal ground
2 **TXD** – Transmitted data — TTL logic, 1=high
3 – Not used
4 **MSB** – Modem status bit— open collector, 50ua sink=inactive
5 **CTS** – Clear to send
6 **RXD** – Receive data — bipolar input, -0.5v-10v=1
7 **+12v** – Connected to power supply through 22 ohms
8 **MCB** – Modem control bit — TTL, low suppresses output
9 **RI** – Ring indicator — TTL, high-to-low sets flag

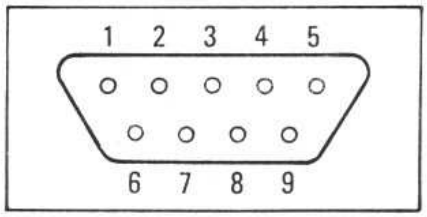


Figure 4.3.2 Modem Pinouts

NOTE: all signals are TTL-level and thus this interface is extremely vulnerable to damage through misuse.

Early versions of the main PC board have a reversal of the modem signals described in this document; the original design called for a female modem socket. Consult the User's Guide which accompanied your Osborne 1 for the applicable modem pinouts. A special cable is required with the Osborne Modem to compensate for the configuration of these earlier connectors.

If you have a modem that uses an RS-232C connector, you may have to use an external adapter box to properly institute all the modem functions. Connecting a modem without using an external adaptor may damage your Osborne 1, as pins 4 and 5 are open collectors and are sensitive to signal-edge transitions. If pin 4 is not connected to the modem, make sure that nothing is connected to pin 4 at the Osborne end; otherwise, adjacent signals may be received inadvertently.

4.3.3 Modem Status

You can determine the output status of the modem port by using the BIOS call LISTST located at 0E12D hex. A value of 0FF hex indicates that the list device is ready; 00 indicates busy.

To find the input status of the modem port, you must first switch to bank 2 of memory and then look at memory location 2A00 hex. To change the status of the modem or serial device directly, you use the same memory location and write a special "control" byte as dictated in the 6850 specification sheet. Memory location 2A01 hex in bank 2 is the data buffer: you read information from external devices by moving the byte to one of the CPU internal registers, you send information to the external device by moving data from the CPU register to the memory location.

4.4 BAUD RATE

Baud rates for the serial and modem ports is software-selectable between 300 or 1200 (use the SETUP program to change the baud rate from 1200 that BIOS assumes to 300). If necessary, the baud rate may be increased (on Revision level E boards and latter) from the 300/1200 baud normally used on the Osborne 1 to 600/2400 baud. To Switch baud rates remove the two-pin jumper from the position "J1" (See Fig. 2.3) on the logic board. Earlier versions require soldering and cutting of traces.

An even faster Baud rate of 1200/2400 or 2400/9600 can be attained with the addition of a few routing wires. The Osborne 1 cannot handle terminal functions above 2400 baud due to the limitation of system calls. However, communication to an external terminal, printer or another computer can be maintained at these higher rates of transmittion.

Note: We are making the procedure for increasing Baud rate available because we are convinced that certain parties are using it successfuly. Osborne Computer Corporation does not officially support these higher Baud rates because the 6850 support driver is potentially unreliable at high speeds. Also, be forwarned that any tampering within the computer will void your waranty.

Here is the procedure for increasing the Baud rate to either 1200/2400 or 2400/9600. Refer to the Figure 2.3 of the main logic board and the illustration below the instructions for more details:

1. Remove LS161/163 from C3 and repace it with a 16 pin socket.
2. Bend up pins 2, 3, 7, and 10.
3. Route a wire between pins 3 and 4, another between pins 4 and 7, and another wire between pins 7 and 10.
4. Place the modified pack in the socket. The modification so far will provide 1200/4800 Baud capability.
5. For 2400/9600 Baud you must connect an extra wire from pin 2 to the middle contact of J2 as illustrated in Figure 4.4.

For those who wish to control the 6850 ACIA directly, Appendix C contains a listing of the registers and instructions the 6850 chip utilizes.

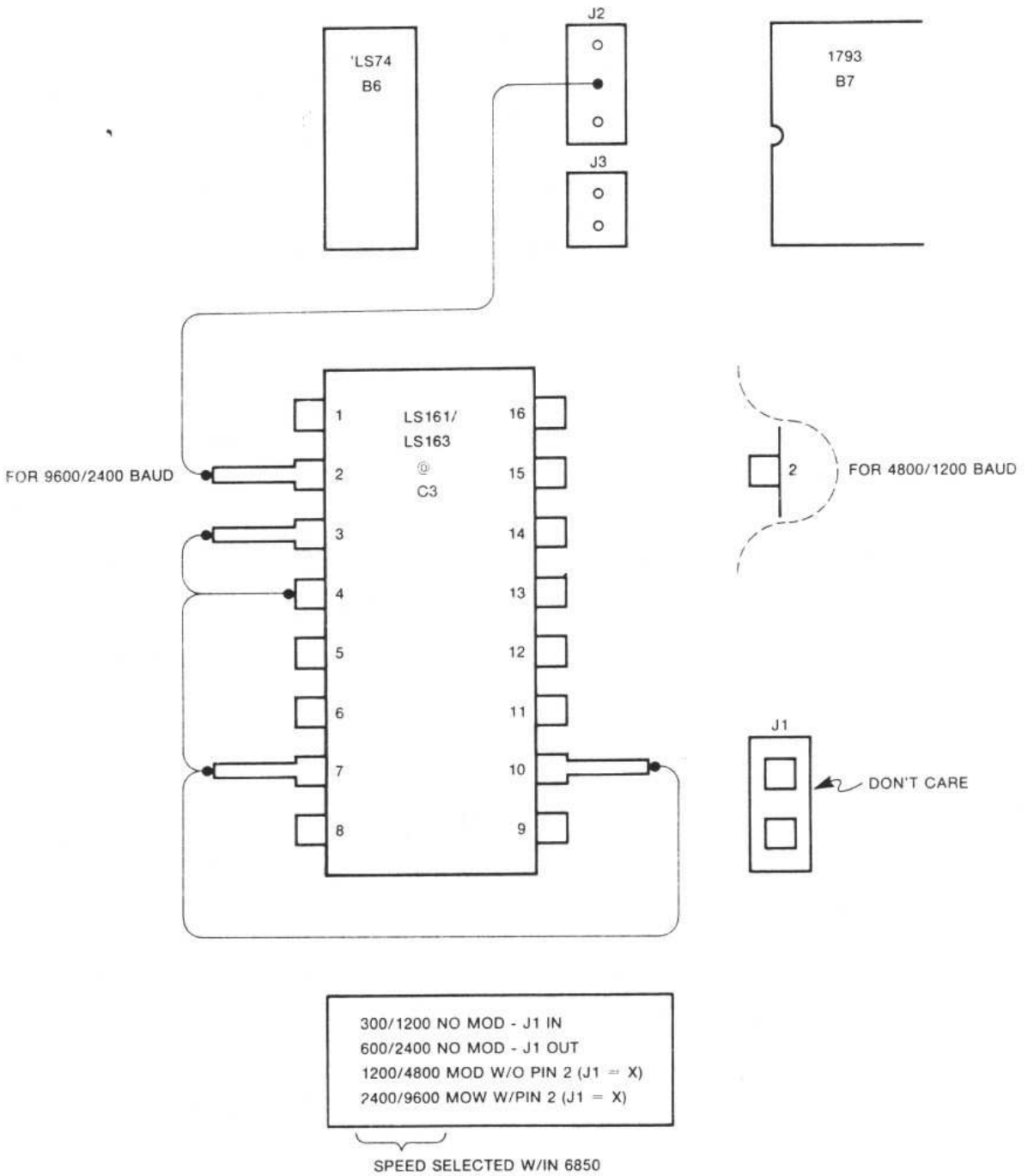


Figure 4.4 Baud Rate Hardware Modification