CHAPTER II

Hardware
Chapter II
Hardware

2.0 Hardware

This application receives data from USB controller (USB Device). The USB controller is connected to an analog device called Temperature sensor which detects room temperature. The Temperature sensor sends the data in Analog format to the USB controller. USB controller in turn converts the Analog data into Digital format.

When the USB Controller is connected to computer related drivers are automatically installed on the system. These drivers are helpful in capturing data through the Monitoring Module.

A USB cable has four wires: power, ground, D+ and D-. The signal wires usually operate in differential mode (one is high while the other is low).

There are two types of USB devices: high speed and low speed. There are differences in the type, length and connectors that can be used with the two types. Generally the requirements are stricter for high speed devices. A high speed device can transfer data at a maximum of 12 Mbs while low speed is limited to 1.5 Mbs. The PIC 16C745 can only operate at low speed standard.

The connector at the computer (A End) is a flattened rectangle while at the device (B end) the connector is square with two cut-off corners (necessary to be compliant with the high-speed standard but optional for low speed devices).

Each USB device is able to draw 100mA from the cable which allows a device to be bus powered. A device can negotiate up to 500mA once connected to
the USB cable but this power may not be available. If the power is unavailable the device must stay in a low power setting. Obviously devices can be powered externally as well.

**The Software:**

When a device is connected, the computer (or a hub) detects the device by a pullup resister (see circuit). A pullup resister on the D- wire signals a low speed device, while a high speed device has a pullup on the D+ wire. When detected a series of enumeration steps are started.

The enumeration process assigns an identifier number to the device (there can be a maximum of 127 devices on a network) and also tells the host computer what sort of capabilities a device has (input, output, etc). The device also informs the computer of its name (Vendor, Product, Version and Serial Number).
2.1 Hardware Connections

P0.0 to P0.31

I/O Port 0: Port 0 is a 32-bit I/O port with individual direction controls for each bit. Total of 31 pins of the Port 0 can be used as a general purpose bidirectional digital I/Os while P0.31 is output only pin. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pins P0.24, P0.26 and P0.27 are not available.

P0.0/TXD0/

PWM1

I/O P0.0 — General purpose input/output digital pin (GPIO).
O TXD0 — Transmitter output for UART0.
O PWM1 — Pulse Width Modulator output 1.

P0.1/RXD0/
PWM3/EINT0
I/O P0.1 — General purpose input/output digital pin (GPIO).
I RXD0 — Receiver input for UART0.
O PWM3 — Pulse Width Modulator output 3.
I EINT0 — External interrupt 0 input

P0.2/SCL0/
CAP0.0
I/O P0.2 — General purpose input/output digital pin (GPIO).
I/O SCL0 — I2C0 clock input/output. Open-drain output (for I2C-bus compliance).
I CAP0.0 — Capture input for Timer 0, channel 0.

P0.3/SDA0/
MAT0.0/EINT1
I/O P0.3 — General purpose input/output digital pin (GPIO).
I/O SDA0 — I2C0 data input/output. Open-drain output (for I2C-bus compliance).
O MAT0.0 — Match output for Timer 0, channel 0.
I EINT1 — External interrupt 1 input.

P0.4/SCK0/
CAP0.1/AD0.6
I/O P0.4 — General purpose input/output digital pin (GPIO).
I/O SCK0 — Serial clock for SPI0. SPI clock output from master or input to slave.
I CAP0.1 — Capture input for Timer 0, channel 0.
I AD0.6 — ADC 0, input 6.

P0.5/MISO0/
MAT0.1/AD0.7
I/O P0.5 — General purpose input/output digital pin (GPIO).
I/O MISO0 — Master In Slave OUT for SPI0. Data input to SPI master or data output from SPI slave.
MAT0.1 — Match output for Timer 0, channel 1.
AD0.7 — ADC 0, input 7.
P0.6/MOSI0/
CAP0.2/AD1.0
I/O P0.6 — General purpose input/output digital pin (GPIO).
I/O MOSI0 — Master Out Slave In for SPI0. Data output from SPI master or data input to SPI slave.
I CAP0.2 — Capture input for Timer 0, channel 2.
I AD1.0 — ADC 1, input 0. Available in LPC2144/46/48 only.
P0.7/SSEL0/
PWM2/EINT2
I/O P0.7 — General purpose input/output digital pin (GPIO).
I SSEL0 — Slave Select for SPI0. Selects the SPI interface as a slave.
O PWM2 — Pulse Width Modulator output 2.
I EINT2 — External interrupt 2 input.
P0.8/TXD1/
PWM4/AD1.1
I/O P0.8 — General purpose input/output digital pin (GPIO).
O TXD1 — Transmitter output for UART1.
O PWM4 — Pulse Width Modulator output 4.
I AD1.1 — ADC 1, input 1. Available in LPC2144/46/48 only.
P0.9/RXD1/
PWM6/EINT3
I/O P0.9 — General purpose input/output digital pin (GPIO).
I RXD1 — Receiver input for UART1.
O PWM6 — Pulse Width Modulator output 6.
I EINT3 — External interrupt 3 input.
P0.10/RTS1/
CAP1.0/AD1.2

I/O P0.10 — General purpose input/output digital pin (GPIO).
O RTS1 — Request to Send output for UART1. LPC2144/46/48 only.
I CAP1.0 — Capture input for Timer 1, channel 0.
I AD1.2 — ADC 1, input 2. Available in LPC2144/46/48 only.

P0.11/CTS1/
CAP1.1/SCL1

I/O P0.11 — General purpose input/output digital pin (GPIO).
I CTS1 — Clear to Send input for UART1. Available in LPC2144/46/48 only.
I CAP1.1 — Capture input for Timer 1, channel 1.
I/O SCL1 — I2C1 clock input/output. Open-drain output (for I2C-bus compliance)

P0.12/DSR1/
MAT1.0/AD1.3

I/O P0.12 — General purpose input/output digital pin (GPIO).
I DSR1 — Data Set Ready input for UART1. Available in LPC2144/46/48 only.
O MAT1.0 — Match output for Timer 1, channel 0.
I AD1.3 — ADC input 3. Available in LPC2144/46/48 only.

P0.13/DTR1/
MAT1.1/AD1.4

I/O P0.13 — General purpose input/output digital pin (GPIO).
O DTR1 — Data Terminal Ready output for UART1. LPC2144/46/48 only.
O MAT1.1 — Match output for Timer 1, channel 1.
I AD1.4 — ADC input 4. Available in LPC2144/46/48 only.

P0.14/DCD1/
EINT1/SDA1
I/O P0.14 — General purpose input/output digital pin (GPIO).
IDCD1 — Data Carrier Detect input for UART1. LPC2144/46/48 only.
IEINT1 — External interrupt 1 input.
I/O SDA1 — I2C1 data input/output. Open-drain output (for I2C-bus compliance)
Note: LOW on this pin while RESET is LOW forces on-chip boot loader to
take over control of the part after reset.

P0.15/RI1/
EINT2/AD1.5
Pin 45 I/O P0.15 — General purpose input/output digital pin (GPIO).
I RI1 — Ring Indicator input for UART1. Available in LPC2144/46/48 only.
IEINT2 — External interrupt 2 input.
I AD1.5 — ADC 1, input 5. Available in LPC2144/46/48 only.

P0.16/EINT0/
MAT0.2/CAP0.2
I/O P0.16 — General purpose input/output digital pin (GPIO).
IEINT0 — External interrupt 0 input.
O MAT0.2 — Match output for Timer 0, channel 2.
I CAP0.2 — Capture input for Timer 0, channel 2.

P0.17/CAP1.2/
SCK1/MAT1.2
I/O P0.17 — General purpose input/output digital pin (GPIO).
I CAP1.2 — Capture input for Timer 1, channel 2.
I/O SCK1 — Serial Clock for SSP. Clock output from master or input to slave.
O MAT1.2 — Match output for Timer 1, channel 2.

P0.18/CAP1.3/
MISO1/MAT1.3
I/O P0.18 — General purpose input/output digital pin (GPIO).
ICAP1.3 — Capture input for Timer 1, channel 3.
I/O MISO1 — Master In Slave Out for SSP. Data input to SPI master or data output from SSP slave.
O MAT1.3 — Match output for Timer 1, channel 3.

P0.19/MAT1.2/
MOSI1/CAP1.2
54 [1] I/O P0.19 — General purpose input/output digital pin (GPIO).
O MAT1.2 — Match output for Timer 1, channel 2.
I/O MOSI1 — Master Out Slave In for SSP. Data output from SSP master or data input to SSP slave.
I CAP1.2 — Capture input for Timer 1, channel 2.

P0.20/MAT1.3/
SSEL1/EINT3
55 [2] I/O P0.20 — General purpose input/output digital pin (GPIO).
O MAT1.3 — Match output for Timer 1, channel 3.
I SSEL1 — Slave Select for SSP. Selects the SSP interface as a slave.
I EINT3 — External interrupt 3 input.

P0.21/PWM5/
AD1.6/CAP1.3
1 [4] I/O P0.21 — General purpose input/output digital pin (GPIO).
O PWM5 — Pulse Width Modulator output 5.
I AD1.6 — ADC 1, input 6. Available in LPC2144/46/48 only.
I CAP1.3 — Capture input for Timer 1, channel 3.

P0.22/AD1.7/
CAP0.0/MAT0.0
2 [4] I/O P0.22 — General purpose input/output digital pin (GPIO).
I AD1.7 — ADC 1, input 7. Available in LPC2144/46/48 only.
I CAP0.0 — Capture input for Timer 0, channel 0.
O MAT0.0 — Match output for Timer 0, channel 0.

**P0.23/VBUS**

58 [1] I/O P0.23 — General purpose input/output digital pin (GPIO).
I VBUS — Indicates the presence of USB bus power.
Note: This signal must be HIGH for USB reset to occur.

**P0.25/AD0.4/ AOUT**

9 [5] I/O P0.25 — General purpose input/output digital pin (GPIO).
I AD0.4 — ADC 0, input 4.
O AOUT — DAC output. Available in LPC2142/44/46/48 only.

**P0.28/AD0.1/ CAP0.2/MAT0.2**

13 [4] I/O P0.28 — General purpose input/output digital pin (GPIO).
I AD0.1 — ADC 0, input 1.
I CAP0.2 — Capture input for Timer 0, channel 2.
O MAT0.2 — Match output for Timer 0, channel 2.

**P0.29/AD0.2/ CAP0.3/MAT0.3**

14 [4] I/O P0.29 — General purpose input/output digital pin (GPIO).
I AD0.2 — ADC 0, input 2.
I CAP0.3 — Capture input for Timer 0, Channel 3.
O MAT0.3 — Match output for Timer 0, channel 3.

**P0.30/AD0.3/ EINT3/CAP0.0**

15 [4] I/O P0.30 — General purpose input/output digital pin (GPIO).
I AD0.3 — ADC 0, input 3.
I EINT3 — External interrupt 3 input.
I CAP0.0 — Capture input for Timer 0, channel 0.

P0.31/UP_LED/

CONNECT
17 [6] O P0.31 — General purpose output only digital pin (GPO).
O UP_LED — USB GoodLink LED indicator. It is LOW when device is configured (non-control endpoints enabled). It is HIGH when the device is not configured or during global suspend.
O CONNECT — Signal used to switch an external 1.5 kW resistor under the software control. Used with the SoftConnect USB feature.
Important: This is an digital output only pin. This pin MUST NOT be externally pulled LOW when RESET pin is LOW or the JTAG port will be disabled.

P1.0 to P1.31
I/O Port 1: Port 1 is a 32-bit bidirectional I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 0 through 15 of port 1 are not available.

P1.16/
TRACEPKT0
O TRACEPKT0 — Trace Packet, bit 0. Standard I/O port with internal pull-up.

P1.17/
TRACEPKT1
O TRACEPKT1 — Trace Packet, bit 1. Standard I/O port with internal pull-up.
P1.18/
TRACEPKT2
O TRACEPKT2 — Trace Packet, bit 2. Standard I/O port with internal pull-up.

P1.19/
TRACEPKT3
O TRACEPKT3 — Trace Packet, bit 3. Standard I/O port with internal pull-up.

P1.20/
TRACESYNC
O TRACESYNC — Trace Synchronization. Standard I/O port with internal pull-up.
Note: LOW on this pin while RESET is LOW enables pins P1.25:16 to operate as Trace port after reset.

P1.21/
PIPESTAT0
O PIPESTAT0 — Pipeline Status, bit 0. Standard I/O port with internal pull-up.

P1.22/
PIPESTAT1
O PIPESTAT1 — Pipeline Status, bit 1. Standard I/O port with internal pull-up.

P1.23/
PIPESTAT2
O PIPESTAT2 — Pipeline Status, bit 2. Standard I/O port with internal pull-up.

P1.24/
TRACECLK
O TRACECLK — Trace Clock. Standard I/O port with internal pull-up.

P1.25/EXTIN0
I EXTIN0 — External Trigger Input. Standard I/O with internal pull-up.

P1.26/RTCK
I/O RTCK — Returned Test Clock output. Extra signal added to the JTAG port.
Assists debugger synchronization when processor frequency varies.
Bidirectional pin with internal pull-up.
Note: LOW on RTCK while RESET is LOW enables pins P1.31:26 to operate as Debug port after reset.

P1.27/TDO
O TDO — Test Data out for JTAG interface.

P1.28/TDI
I TDI — Test Data in for JTAG interface.

P1.29/TCK
I TCK — Test Clock for JTAG interface.
P1.30/TMS
ITMS — Test Mode Select for JTAG interface.

P1.31/TRST
ITRST — Test Reset for JTAG interface.

D+
10 [7] I/O USB bidirectional D+ line.

D-

RESET
57 [8] I External reset input: A LOW on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.

XTAL1
62 [9] I Input to the oscillator circuit and internal clock generator circuits.

XTAL2

RTXC1
3 I Input to the RTC oscillator circuit.

RTXC2

VSS
I Ground: 0 V reference.

**VSSA**

59 I Analog ground: 0 V reference. This should nominally be the same voltage as VSS, but should be isolated to minimize noise and error.

**VDD**

23, 43, 51 I 3.3 V power supply: This is the power supply voltage for the core and I/O ports.

**VDDA**

71 Analog 3.3 V power supply: This should be nominally the same voltage as VDD but should be isolated to minimize noise and error. This voltage is only used to power the on-chip ADC(s) and DAC.

**VREF**

63 I A/D converter reference voltage: This should be nominally less than or equal to the VDD voltage but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC(s) and DAC.

**VBAT**

49 I RTC power supply voltage: 3.3 V on this pin supplies the power to the RTC.

1. 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control.

2. 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control. If configured for an input function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns.

3. Open-drain 5 V tolerant digital I/O I2C-bus 400 kHz specification compatible pad. It requires external pull-up to provide an output.
functionality.

4. 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog input function. If configured for an input function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns. When configured as an ADC input, digital section of the pad is disabled.

5. 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog output function. When configured as the DAC output, digital section of the pad is disabled.

6. 5 V tolerant pad with built-in pull-up resistor providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control. The pull-up resistor's value typically ranges from 60 kW to 300 kW.

7. Pad is designed in accordance with the Universal Serial Bus (USB) specification, revision 2.0 (Full-speed and Low-speed mode only).

8. 5 V tolerant pad providing digital input (with TTL levels and hysteresis) function only.

9. Pad provides special analog functionality.

2.2.1 LM35 (Electrical Connections)

- Here is a commonly used circuit. For connections refer to the picture above.

- In this circuit, parameter values commonly used are:
  - \( V_c = 4 \) to 30v
  - 5v or 12 v are typical values used.
  - \( R_a = V_c / 10^6 \)
  - Actually, it can range from 80 kW to 600 KW, but most just use 80 kW.
Here is a photo of the LM 35 wired on a circuit board.

- The white wire in the photo goes to the power supply.
- Both the resistor and the black wire go to ground.
- The output voltage is measured from the middle pin to ground.

What Can Expect When Use An LM35?

- It will need to use a voltmeter to sense Vout.
- The output voltage is converted to temperature by a simple conversion factor.
- The sensor has a sensitivity of 10mV / °C.
- Use a conversion factor that is the reciprocal, that is 100 °C/V.
- The general equation used to convert output voltage to temperature is:
• Temperature (°C) = Vout * (100 °C/V)
• So if Vout is 1V, then, Temperature = 100 °C
• The output voltage varies linearly with temperature.

2.2.2 Interfacing – 16 Character x 2Line LCD

Description.

This is the first interfacing example for the Parallel Port. To start with simple example is given. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

2.2.3 Schematic
2.2.4 Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector/open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program. The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, left the power supply has been left. A bench power supply set can be used to 5v or a onboard +5 regulator can be used. Remember a few de-coupling capacitors, especially if there is any trouble with the circuit operation.
The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780. To test this circuit has been used a Powertip PC-1602F and an old Philips LTN211F-10 which was extracted from a Poker Machine! The diagram to the right, shows the pin numbers for these devices. When viewed from the front, the left pin is pin 14 and the right pin is pin 1.

2.3 Serial Communication

One of the 8051s many powerful features is its integrated UART, otherwise known as a serial port. The fact that the 8051 has an integrated serial port means that you may very easily read and write values to the serial port. If it were not for the integrated serial port, writing a byte to a serial line would be a rather tedious process requiring turning on and off one of the I/O lines in rapid succession to properly "clock out" each individual bit, including start bits, stop bits, and parity bits. All IBM PC and compatible computers are typically equipped with two serial ports and one parallel port. Although these two types of ports are used for communicating with external devices, they work in different ways. In fact, two-way (full duplex) communications is possible with only three separate wires - one to send, one to Communicating by Bits

Once the start bit has been sent, the transmitter sends the actual data bits. There may either be 5, 6, 7, or 8 data bits, depending on the number that have been selected. Both receiver and the transmitter must agree on the number of data bits, as well as the baud rate. Almost all devices transmit data using either 7 or 8 bits
Notice that when only 7 data bits are employed, you cannot send ASCII values greater than 127. Likewise, using 5 bits limits the highest possible value to 31. After the data has been transmitted, a stop bit is sent. A stop bit has a value of 1 - or a mark state - and it can be detected correctly even if the previous data bit also had a value of 1. This is accomplished by the stop bit’s duration. Stop bits can be 1, 1.5, or 2 bit periods in length.

**The Parity Bit**

Besides the synchronization provided by the use of start and stop bits, an additional bit called a parity bit may optionally be transmitted along with the data. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose either even parity, odd parity, mark parity, space parity or none at all. When even or odd parity is being used, the number of marks (logical 1 bits) in each data byte are counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bits just sent is even or odd. Mark parity means that the parity bit is always set to the mark signal condition and likewise space parity always sends the parity bit in the space signal condition. Since these two parity options serve no useful purpose whatsoever, they are almost never used.

**2.3.1 RS-232C**

RS-232 stands for Recommend Standard number 232 and C is the latest revision of the standard. The serial ports on most computers use a subset of the RS-232C standard. The full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins.

MAX 232 Logic Signal Voltage
Serial RS-232 (V.24) communication works with voltages (-15V ... -3V for high [sic]) and +3V ... +15V for low [sic]) which are not compatible with normal computer logic voltages. On the other hand, classic TTL computer logic operates between 0V ... +5V (roughly 0V ... +0.8V for low, +2V ... +5V for high). Modern low-power logic operates in the range of 0V ... +3.3V or even lower.

So, the maximum RS-232 signal levels are far too high for computer logic electronics, and the negative RS-232 voltage for high can't be grokked at all by computer logic. Therefore, to receive serial data from an RS-232 interface the voltage has to be reduced, and the low and high voltage level inverted. In the other direction (sending data from some logic over RS-232) the low logic voltage has to be "bumped up", and a negative voltage has to be generated, too.

<table>
<thead>
<tr>
<th>RS-232</th>
<th>TTL</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15V ... -3V</td>
<td>+2V ... +5V</td>
<td>high</td>
</tr>
<tr>
<td>+3V ... +15V</td>
<td>0V ... +0.8V</td>
<td>low</td>
</tr>
</tbody>
</table>

The MAX232 & MAX232A

Here is a diagram of the internals of the MAX232A. It shows a double charge pump voltage doubler and a +10v to -10v voltage inverter. The voltages output are used to generate the RS-232 compliant signals. The MAX232A has provisions for two serial ports on the same physical package. Most people only connect one of them.

You can get a data sheet for the MAX232 and friends from Maxim.
MAX232 internals

The MAX232 from Maxim was the first IC which in one package contains the necessary drivers (two) and receivers (also two), to adapt the RS-232 signal voltage levels to TTL logic. It just needs one voltage (+5V) and generates the necessary RS-232 voltage levels (approx. -10V and +10V) internally. This greatly simplified the design of circuitry. Circuitry designers no longer need to design and build a power supply with three voltages (e.g. -12V, +5V, and +12V), but could just provide one +5V power supply, e.g. with the help of a simple 7805 voltage converter. MAX232A only needs external capacitors 1/10th the capacity of what the original MAX232 needs.

It should be noted that the MAX232(A) is just a driver/receiver. It does not generate the necessary RS-232 sequence of marks and spaces with the right timing, it does not decode the RS-232 signal, it does not provide a serial/parallel conversion. All it does is to convert signal voltage levels. Generating serial data with the right timing and decoding serial data has to be done by additional circuitry, or one of these small micro controllers getting more and more popular. A similar IC, the MAX3232 is nowadays available for low-power 3V logic.
## MAX232(A) DIP Package Pin Layout

<table>
<thead>
<tr>
<th>Nbr</th>
<th>Name</th>
<th>Purpose</th>
<th>Signal Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C1+</td>
<td>+ connector for capacitor C1</td>
<td>capacitor should stand at least 16V</td>
</tr>
<tr>
<td>2</td>
<td>V+</td>
<td>output of voltage pump</td>
<td>+10V</td>
</tr>
<tr>
<td>3</td>
<td>C1-</td>
<td>- connector for capacitor C1</td>
<td>capacitor should stand at least 16V</td>
</tr>
<tr>
<td>4</td>
<td>C2+</td>
<td>+ connector for capacitor C2</td>
<td>capacitor should stand at least 16V</td>
</tr>
<tr>
<td>5</td>
<td>C2-</td>
<td>- connector for capacitor C2</td>
<td>capacitor should stand at least 16V</td>
</tr>
<tr>
<td>6</td>
<td>V-</td>
<td>output of voltage pump / inverter</td>
<td>-10V</td>
</tr>
<tr>
<td>7</td>
<td>T2out</td>
<td>Driver 2 output</td>
<td>RS-232</td>
</tr>
<tr>
<td>8</td>
<td>R2in</td>
<td>Receiver 2 input</td>
<td>RS-232</td>
</tr>
<tr>
<td>9</td>
<td>R2out</td>
<td>Receiver 2 output</td>
<td>TTL</td>
</tr>
<tr>
<td>10</td>
<td>T2in</td>
<td>Driver 2 input</td>
<td>TTL</td>
</tr>
<tr>
<td>11</td>
<td>T1in</td>
<td>Driver 1 input</td>
<td>TTL</td>
</tr>
<tr>
<td>12</td>
<td>R1out</td>
<td>Receiver 1 output</td>
<td>TTL</td>
</tr>
<tr>
<td>13</td>
<td>R1in</td>
<td>Receiver 1 input</td>
<td>RS-232</td>
</tr>
<tr>
<td>14</td>
<td>T1out</td>
<td>Driver 1 output</td>
<td>RS-232</td>
</tr>
<tr>
<td>15</td>
<td>GND</td>
<td>Ground</td>
<td>0V</td>
</tr>
<tr>
<td>16</td>
<td>VCC</td>
<td>Power supply</td>
<td>+5V</td>
</tr>
</tbody>
</table>

MAX232(A) DIP Package

```
+-----+-----+-----+-----+
| 16  | VCC |
| 1   | C1+ |
| 2   | V+  |
| 3   | C1- |
| 4   | C2+ |
| 5   | C2- |
| 6   | V-  |
+-----+-----+-----+-----+
```

```
T2out => T2in
R2in => R2out
T1in => T1out
R1in => R1out
```
The MAX232(A) has two receivers (converts from RS-232 to TTL voltage levels) and two drivers (converts from TTL logic to RS-232 voltage levels). This means only two of the RS-232 signals can be converted in each direction. The old MC1488/1498 combo provided four drivers and receivers.

Typically a pair of a driver/receiver of the MAX232 is used for TX and RX and the second one for CTS and RTS.

There are not enough drivers/receivers in the MAX232 to also connect the DTR, DSR, and DCD signals. Usually these signals can be omitted when e.g. communicating with a PC's serial interface. If the DTE really requires these signals either a second MAX232 is needed, or some other IC from the MAX232 family can be used (if it can be found in consumer electronic shops at all). An alternative for DTR/DSR is also given below.

Maxim's data sheet explains the MAX232 family in great detail, including the pin configuration and how to connect such an IC to external circuitry. This information can be used as-is in own design to get a working RS-232 interface. Maxim's data just misses one critical piece of information: How exactly to connect the RS-232 signals to the IC. So here is one possible example:

MAX232 to RS232 DB9 Connection as a DCE

<table>
<thead>
<tr>
<th>MAX232 Pin Nbr.</th>
<th>MAX232 Pin Name</th>
<th>Signal</th>
<th>Voltage</th>
<th>DB9 Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>T2out</td>
<td>CTS</td>
<td>RS-232</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>R2in</td>
<td>RTS</td>
<td>RS-232</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>R2out</td>
<td>RTS</td>
<td>TTL</td>
<td>n/a</td>
</tr>
<tr>
<td>10</td>
<td>T2in</td>
<td>CTS</td>
<td>TTL</td>
<td>n/a</td>
</tr>
<tr>
<td>11</td>
<td>T1in</td>
<td>TX</td>
<td>TTL</td>
<td>n/a</td>
</tr>
<tr>
<td>12</td>
<td>R1out</td>
<td>RX</td>
<td>TTL</td>
<td>n/a</td>
</tr>
<tr>
<td>13</td>
<td>R1in</td>
<td>RX</td>
<td>RS-232</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>T1out</td>
<td>TX</td>
<td>RS-232</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>GND</td>
<td>GND</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
In addition one can directly wire DTR (DB9 pin 4) to DSR (DB9 pin 6) without going through any circuitry. This gives automatic (brain dead) DSR acknowledgement of an incoming DTR signal.

Sometimes pin 6 of the MAX232 is hard wired to DCD (DB9 pin 1). This is not recommended. Pin 6 is the raw output of the voltage pump and inverter for the -10V voltage. Drawing currents from the pin leads to a rapid breakdown of the voltage, and as a consequence to a breakdown of the output voltage of the two RS-232 drivers. It is better to use software which doesn't care about DCD, but does hardware-handshaking via CTS/RTS only.

The circuitry is completed by connecting five capacitors to the IC as it follows. The MAX232 needs 1.0µF capacitors, the MAX232A needs 0.1µF capacitors. MAX232 clones show similar differences. It is recommended to consult the corresponding data sheet. At least 16V capacitor types should be used. If electrolytic or tantalic capacitors are used, the polarity has to be observed. The first pin as listed in the following table is always where the plus pole of the capacitor should be connected to.

**MAX232(A) external Capacitors**

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>+ Pin</th>
<th>- Pin</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>2</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>GND</td>
<td>6</td>
<td>This looks non-intuitive, but because pin 6 is on -10V, GND gets the + connector, and not the -</td>
</tr>
<tr>
<td>C5</td>
<td>16</td>
<td>GND</td>
<td></td>
</tr>
</tbody>
</table>