PC Interfacing

Fourth Level

Lecture Three

Serial (RS232) Port Interface

Goals:
Up-on completing this lecture, the student should be able to:

1- Identify the main concept of serial data transmission
2- Use the serial transmission parameters according to the desired QoS
3- Use the concept of Null-Modem in the designs with the right coding

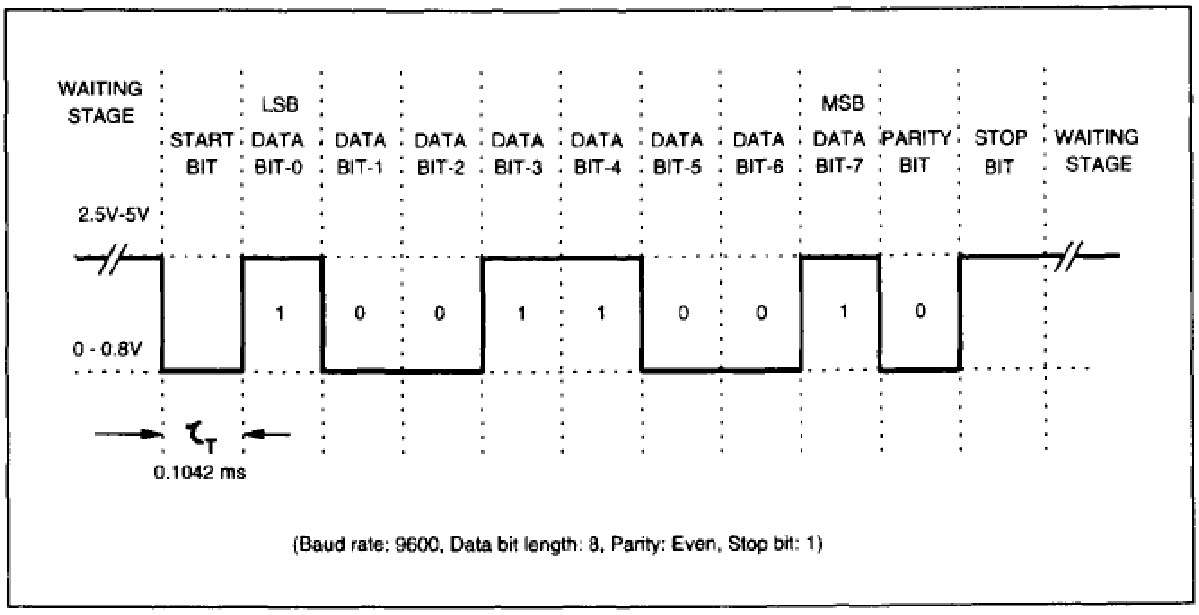
1. RS232 serial interface:-

The RS232 serial interface is an industrial standard bi-directional data communication interface. For computers, it is used for connecting printers, modems, mice, etc. The communication distance is 20 meters.

Unlike a parallel I/O port, which consists of a number of data lines and each time transmits a byte; the serial data transmission requires only one line. A byte is transmitted bit by bit. This reduces data lines between devices. It reduces the rate of data transfer too; maximum data rates may be up to 20 kbps.

2. Serial data transmission:-

A serial data format includes four parts: a start bit (1 bit), serial data bits (5, 6, 7 or 8 bits), parity check bit (1 bit) and stop bits (1 or 1.5 bit). Figure 1.6 shows a typical serial data format. When no data is sent, the data line is at logic high. This is called the waiting stage.



(Baud rate: 9600, Data bit length: 8, Parity: Even, Stop bit: 1)

Fig. 1.6: the format of a serial data transmission produced by the UARTs

The beginning of a data transmission is indicated by pulling the line to the logic low state for 1 bit time. This bit is the start bit. The data bits are then sent out one after another. The number of the data bits can be 6, 7 or 8. Following the data bits comes the parity bit which is used to check transmission errors occurred during the data transmission. The parity check can be ODD, EVEN or NONE. The odd and even parities indicate that the total number of ones ('1') in the transmitted serial data is an odd number or an even number. It is only reliable to detect single-bit errors. Errors occurred to several bits cannot be detected. The last bits are the stop bits, which pull the data line to the high state for at least 1 bit time to indicate the end of the data transmission. The number of the stop bits can be 1, 1.5 and 2 bits.

The rate at which the data bits are sent is measured by the baud rate. The standard baud rates for an RS232 serial port are 110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19200. Knowing the baud rate, the number of bytes to be transmitted per second can be calculated. For example, if a serial data has 8 data bits, no parity check and 1 stop bit, the total length of serial data bits is 10. The transfer rate for characters is the baud rate divided by 10. A baud rate of 9600 will transfer 960 characters per second. What is the effects baud-rate on error rate? (H.W.)

A specially designed electronic device which generates and receives the asynchronous serial data is called the Universal Asynchronous Receiver/Transmitter (UART). The serial data transmission format is generated by the transmitting UART.

The Universal Asynchronous Receiver / Transmitter (UART) controller is the key component of the serial communications of a computer. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART reassembles the bits into complete byte.

The UARTS have a TTL voltage level. In order to achieve a long distance communication, the TTL voltage level is converted to higher voltage level (logic 0 = -12 to - 3V, logic 1 = +3V to +12V). This is achieved by using dedicated RS232 drivers/receivers. All drivers/receivers have an inverting action.

UART is responsible for sending and receiving a sequence of bits. At the output of a UART these bits are usually represented by logic level voltages. These bits can become RS232.

RS232 specifies voltage levels. Notice that some of these voltage levels are negative, and they can also reach ±15V.

A microcontroller UART cannot generate such voltages levels by itself. This is done with help of an additional component: RS232 line driver. A classic example of an RS232 line driver is MAX232. This IC has a charge pump, which generates ±10V from +5V.

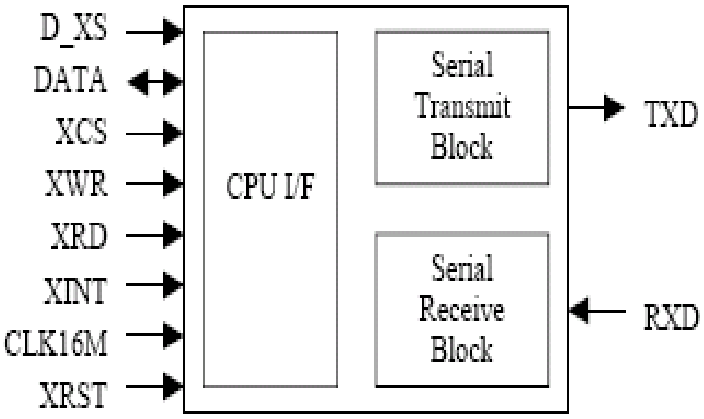
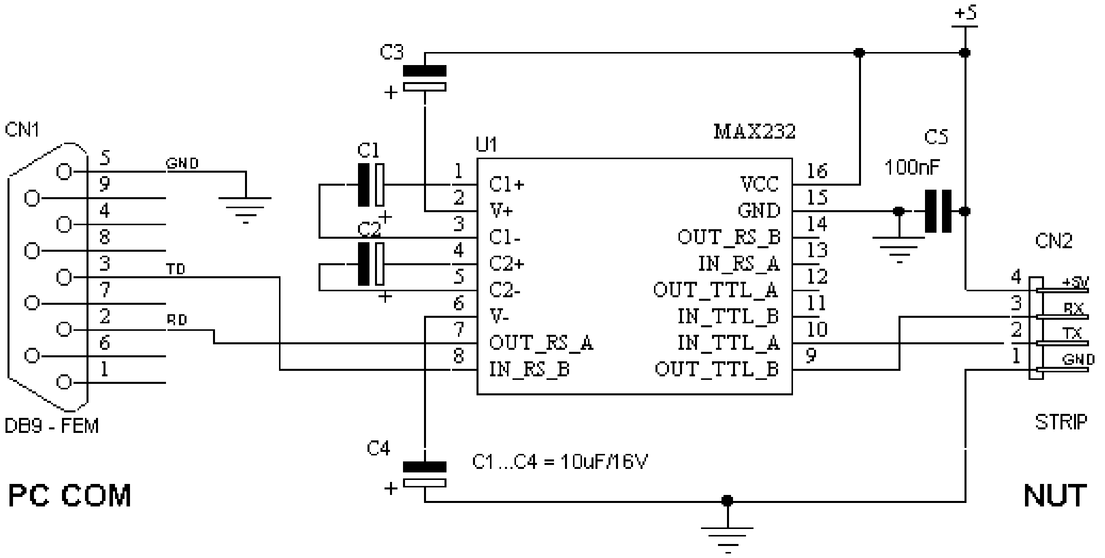
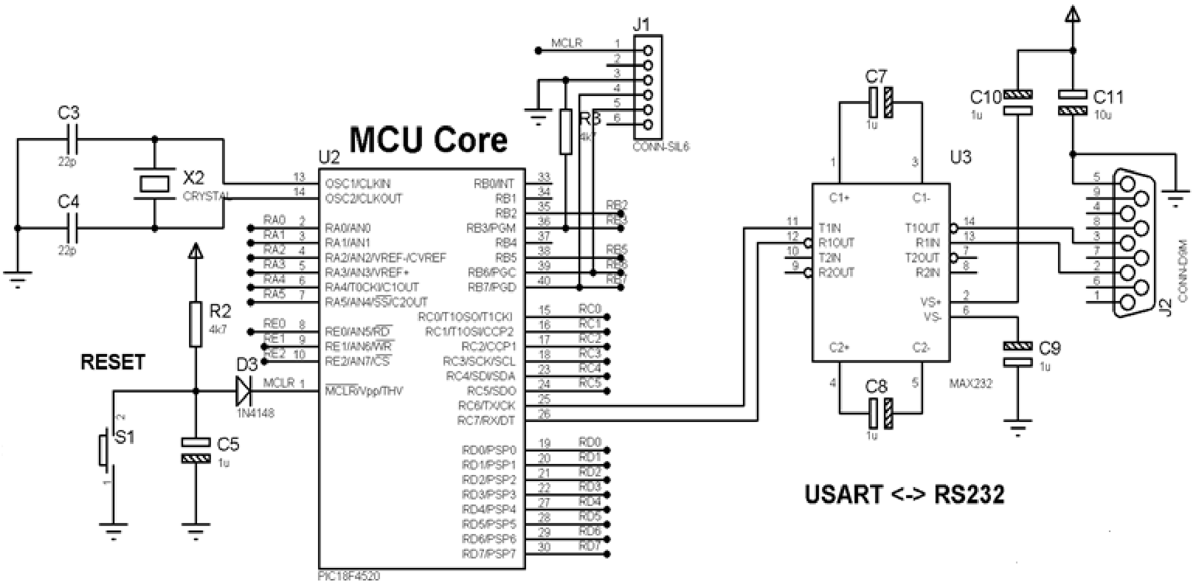


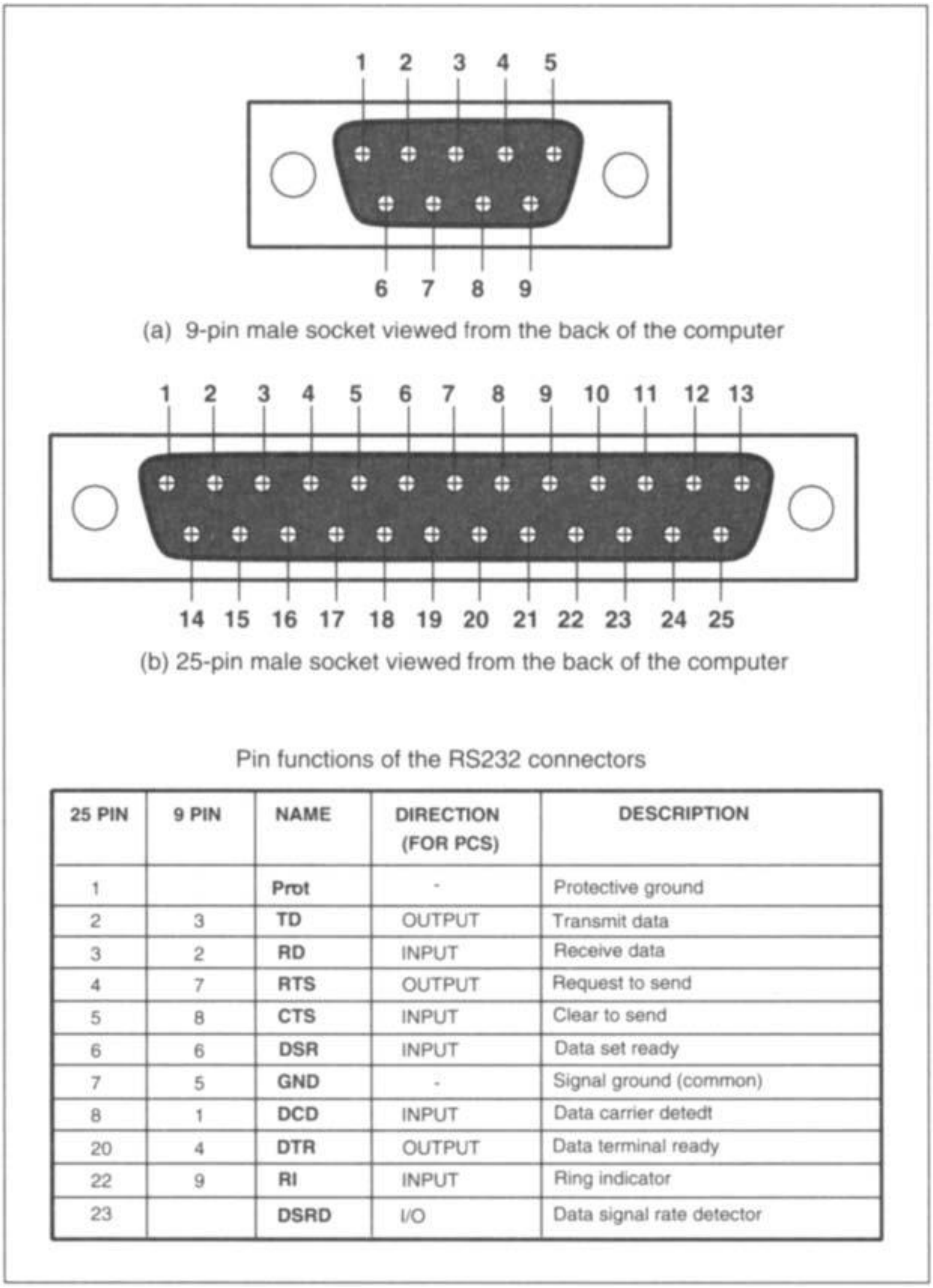
Figure 1. Basic UART block diagram.





3. RS232 port connector and connections:-

A standard RS232 interface is a 25-pin interface housed in a 25-pin or a 9-pin D-type male connector. Figure 1.7 gives the pin-out and functions of connectors.



(a) 9-pin male socket viewed from the back of the computer

(b) 25-pin male socket viewed from the back of the computer

Pin functions of the RS232 connectors

25 PIN	9 PIN	NAME	DIRECTION (FOR PCS)	DESCRIPTION
1		Prot	-	Protective ground
2	3	TD	OUTPUT	Transmit data
3	2	RD	INPUT	Receive data
4	7	RTS	OUTPUT	Request to send
5	8	CTS	INPUT	Clear to send
6	6	DSR	INPUT	Data set ready
7	5	GND	-	Signal ground (common)
8	1	DCD	INPUT	Data carrier detedt
20	4	DTR	OUTPUT	Data terminal ready
22	9	RI	INPUT	Ring indicator
23		DSRD	I/O	Data signal rate detector

Fig. 1.7: pin-out and function of the RS232 connectors and computers

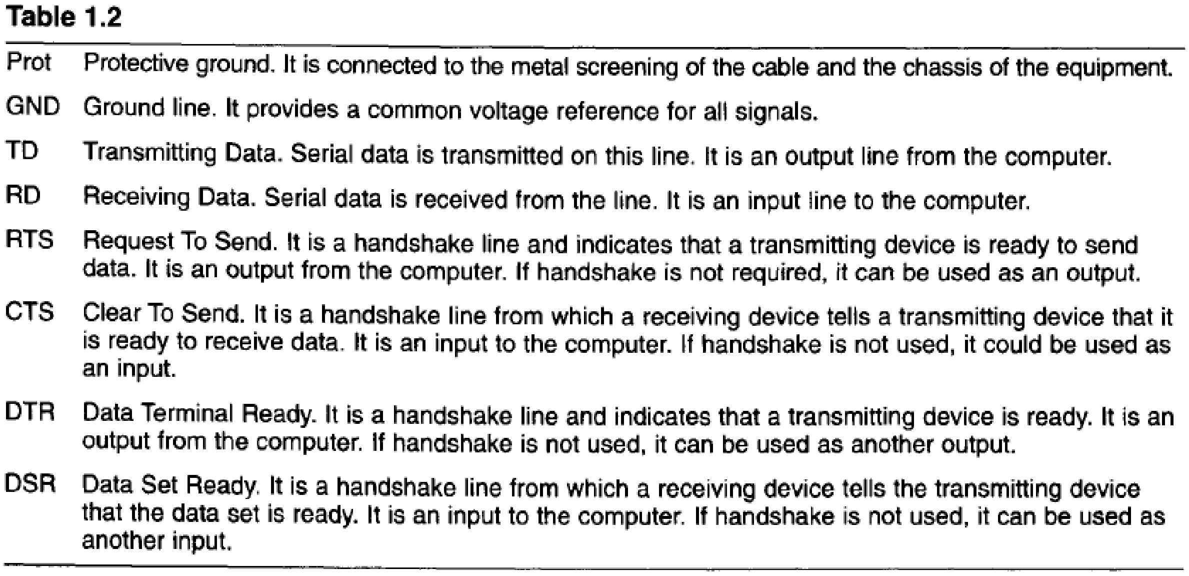
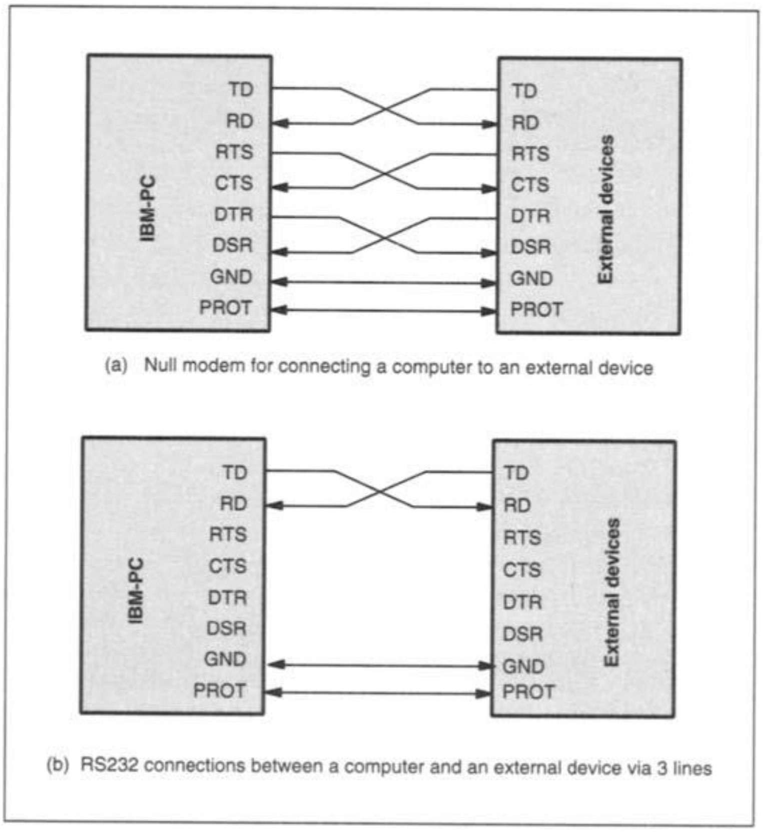


Table 1.2

Prot	Protective ground. It is connected to the metal screening of the cable and the chassis of the equipment.
GND	Ground line. It provides a common voltage reference for all signals.
TD	Transmitting Data. Serial data is transmitted on this line. It is an output line from the computer.
RD	Receiving Data. Serial data is received from the line. It is an input line to the computer.
RTS	Request To Send. It is a handshake line and indicates that a transmitting device is ready to send data. It is an output from the computer. If handshake is not required, it can be used as an output.
CTS	Clear To Send. It is a handshake line from which a receiving device tells a transmitting device that it is ready to receive data. It is an input to the computer. If handshake is not used, it could be used as an input.
DTR	Data Terminal Ready. It is a handshake line and indicates that a transmitting device is ready. It is an output from the computer. If handshake is not used, it can be used as another output.
DSR	Data Set Ready. It is a handshake line from which a receiving device tells the transmitting device that the data set is ready. It is an input to the computer. If handshake is not used, it can be used as another input.

Two types of RS232 link between a computer and an external device are shown in Figure 1.8. The arrows show the direction of data flow. Figure 1.8(a) is known as the null modem. Figure 1.8(b) shows a connection using only three lines. One line is for transmitting data and the other for receiving data. The connection is arranged so that the transmitting line of the first device is connected to the receiving line of the second device.



(a) Null modem for connecting a computer to an external device

(b) RS232 connections between a computer and an external device via 3 lines

Fig. 1.8: RS232 connections between a PC and an external device

4. Internal hardware organization:-

A computer can have up to four RS232 interfaces installed. They are labeled COM1 to COM4. Each COM port is associated with a UART inside the computer.

RS232 driver/receivers

The RS232 output control signals (-RTS and -DTR) and input status signals (-CTS,-DSR) are processed by the UART in an inverted form. The serial data signal S_{IN} and S_{OUT} are in a non-inverting form. The UART produces the TT'L/CMOs voltage levels only. RS232 line drivers/receivers are connected between the UART and the RS232 connector. The drivers convert the TTL voltage to the RS232 voltage level and the receivers convert the RS232 level to the TTL level. All the drivers/receivers have inverting action. The logic structure of the RS232 port is shown in Figure 1.10.

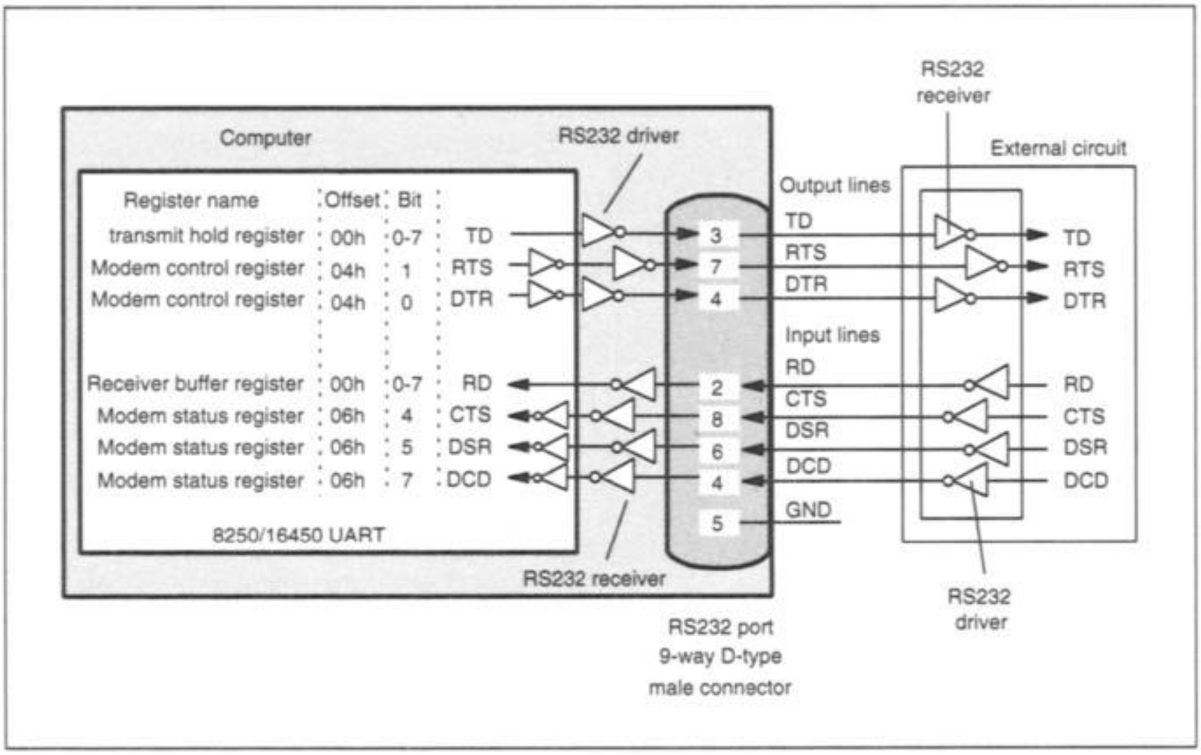


Figure 1.10 Logic structure of the RS232 port

Base addresses of COM ports

The base addresses of COM1 to COM4 are summarized below.

COMI: 3F8h

COM2: 2F8h

COM3: 3E8h

COM4: 2E8h

When a computer is switched on or reset, the BIOS checks all possible RS232 addresses. If it finds an installed one, it writes the base address (a 2-byte word) into specific memory locations. By reading these locations, the base address can be obtained. The memory locations for COM1 to COM4 are listed below.

COM1: 0000:0400h - 0000:0401h

COM2: 0000:0402h - 0000:0403h

COM3: 0000:0404h - 0000:0405h

COM4: 0000:0406h - 0000:0407h

Another useful one-byte memory location is 0000:4011h. It stores the total number of COMs installed. The information is contained in bit 3, bit 2 and bit 1 of the byte.

bit 3=0, bit 2=0, bit 1=0 → no COM port installed

bit 3=0, bit 2=0, bit 1=1→ one COM port installed

bit 3=0, bit 2=1, bit 1=0 → two COM ports installed

bit 3=0, bit 2=1, bit 1=1 → three COM ports installed

bit 3=1, bit 2=0, bit 1=0 → four COM ports installed

Software control

a-How to obtain the base address of COM port

- **-QBASIC**
- **-TP6**
- **-WINDOWS**

b-How to transmit and receive serial data

- **-Printer commands &BIOS interrupt calls**

As the bit-rate increases, the duration of the frame decreases. Hence the total energy of the bit goes down. While its probability of being hit by interference goes up. Choosing the right settings can counter-combat that.
For example, Parity bit and 5 data bits with 110 symbols/second provide the best error-tolerance that the UART can provide.

While, no carry bit with 8 data bits and 19200 symbols/second provides the fastest rate possible.

Summary
1- The serial interface uses a different concept of cascading the data to be sent.
2- There are many settings which can be manipulated to counter-act the noise levels vs the required transmission rate
3- The null modem is a simple, yet effective configuration for data inferface.
4- There is a difference between UART and RS232 although both support the same frame format, however they differ in the used voltage.

Quesions:
1- What is the principle of operation of the serial port?
2- What is the difference between RS232 and UART?
3- What setting(s) give the best immunity against jamming?
4- What setting(s) give the highest througput?
5- What is the null modem?
6- Design a circuit to send the data of a 10 bit A/D using the serial port, write the supporting code as well.