## Serial Port

The microcontroller AT89C51 has an inbuilt UART for carrying out serial communication. The serial communication is done in the asynchronous mode. A serial port, like other PC ports, is a physical interface to establish data transfer between computer and an external hardware or device. This transfer, through serial port, takes place bit by bit.

IBM introduced the **DB-9 RS-232** version of serial I/O standard, which is most widely used in PCs and several devices. In RS232, high and low bits are represented by flowing voltage ranges:

| Bit | Voltage Range (in V) | |
|---|---|---|
| 0 | +3 | +25 |
| 1 | -25 | -3 |

The range -3V to +3V is undefined. The TTL standards came a long time after the RS232 standard was set. Due to this reason RS232 voltage levels are not compatible with TTL logic. Therefore, while connecting an RS232 to microcontroller system, a voltage converter is required. This converter converts the microcontroller output level to the RS232 voltage levels, and vice versa. IC MAX232, also known as line driver, is very commonly used for this purpose.

The simplest connection between a PC and microcontroller requires a minimum of three pins, RxD (receiver, pin2), TxD (transmitter, pin3) and ground (pin5) of the serial port of computer.
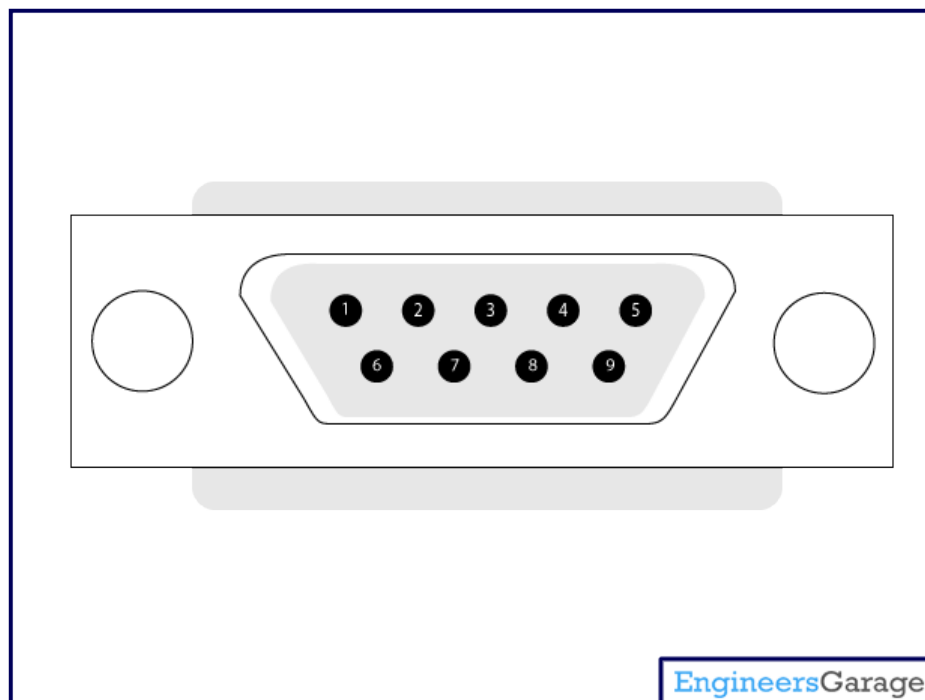
TxD pin of serial port connects to RxD pin of controller via MAX232. And similarly, RxD pin of serial port connects to the TxD pin of controller through MAX232.

MAX232 has two sets of **line drivers** for transferring and receiving data. The line drivers used for transmission are called T1 and T2, where as the line drivers for receiver are designated as R1 and R2. The connection of MAX232 with computer and the controller is shown in the circuit diagram.
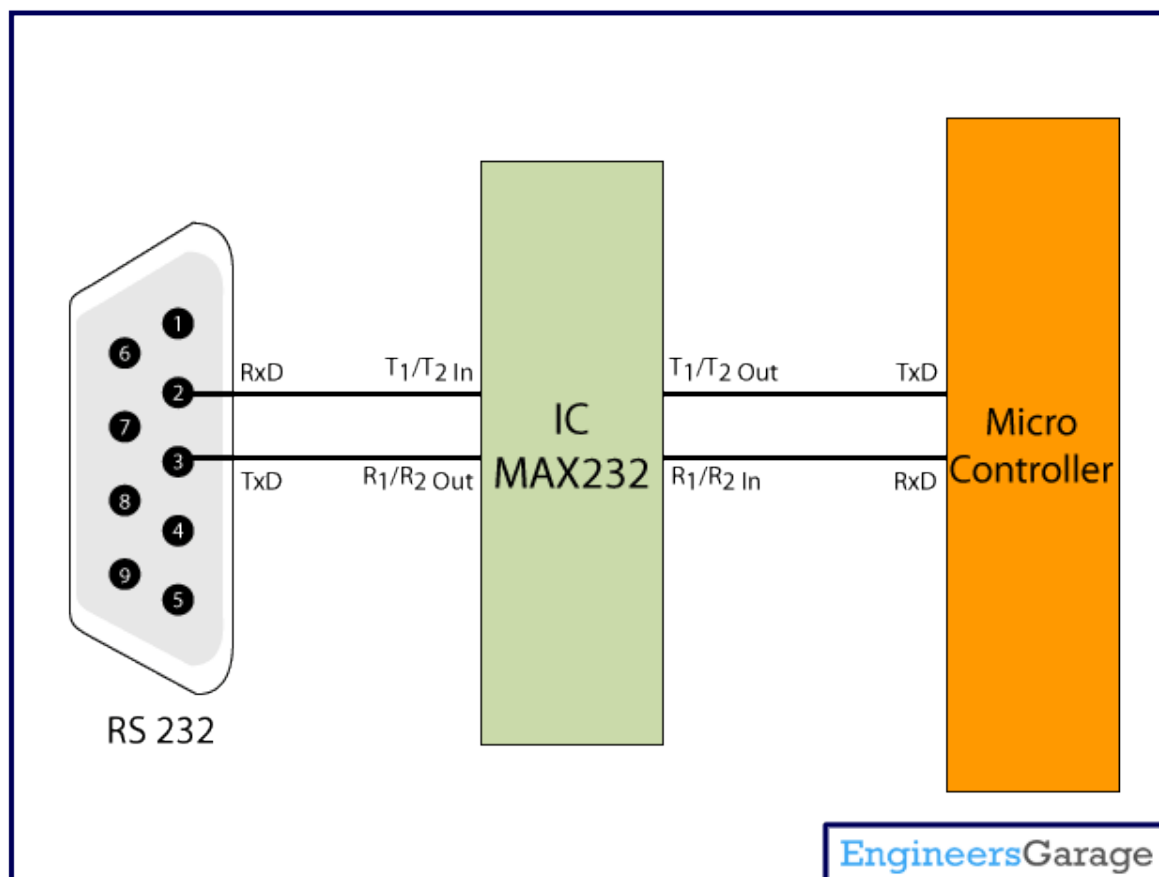


An important parameter considered while interfacing serial port is the Baud rate which is the speed at which data is transmitted serially. It is defined as number of bits transmitted or received per second. It is generally expressed in bps (bits per second). AT89C51 microcontroller can be set to transfer and receive serial data at different baud rates using software instructions. Timer1 is used to set the baud rate of serial communication for the microcontroller. For this purpose, Timer1 is used in mode2 which is an 8-bit auto reload mode. (Refer Timer programming with 8051) To get baud rates compatible with the PC, TH1 should be loaded with the values as shown:

| Baud Rate (bps) | TH1 (Hex value) |
| --- | --- |
| 9600 | FD |
| 4800 | FA |
| 2400 | F4 |
| 1200 | E8 |

In this project baud rate 9600bps is used.

For serial communication AT89C51 has registers **SBUF** and **SCON** (Serial control register). SBUF is an 8-bit register. For transmitting a data byte serially, it needs to be placed in the SBUF register. Similarly whenever a data byte is received serially, it comes in the SBUF register, *i.e.*, SBUF register should be read to receive the serial byte.

SCON register is used to set the mode of serial communication. The project uses Mode1, in which the data length is of 8 bits and there is a start and a stop bit. The SCON register is bit addressable register. The following table shows the configuration of each bit.

**SCON (Serial Control) Register**

| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
|-----|-----|-----|-----|-----|-----|-----|-----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

**SM0**     **SM1**

| | | |
|---|---|---|
| 0 | 0 | Serial mode 0 |
| 0 | 1 | Serial mode 1, 8-bit data, 1 start bit, 1 stop bit |
| 1 | 0 | Serial mode 2 |
| 1 | 1 | Serial mode 3 |

**TI (transmit interrupt)** is an important flag bit in the SCON register. The controller raises the TI flag when the 8-bit character is transferred. This indicates that the next byte can be transferred now. The TI bit is raised at the beginning of the stop bit.

**RI (receive interrupt)** is also a flag bit of the SCON register. On receiving the serial data, the microcontroller skips the start and stop bits, and puts the byte is SBUF register. The RI flag bit is then raised to indicate that the byte has been received and should be picked up.

## Hyper Terminal

Hyper Terminal, a Windows XP application, can be used to receive or transmit serial data through RS232. To open Hyper Terminal, go to Start Menu, select all programs, go to Accessories, click on Communications and select Hyper Terminal.

To start a new connection, go to File menu and click on new connection. The connection window opens up. Give a name to your connection and select 1$^{st}$icon and click on OK. Connection property window opens here. Select Bit rate as 9600bps, Data bits 8, Parity as none, Stop bit 1, Flow control none and click OK. Now the serial data can be read on hyper terminal.

In program, Timer1 is used with auto reload setting. The baud rate is fixed to 9600bps by loading TH1 to 0xFD. The value 0x50 is loaded in the SCON register. This will initialize the serial port in Mode1. The program continuously receives a character (say 'a') from the serial port of the computer and transmits it back.