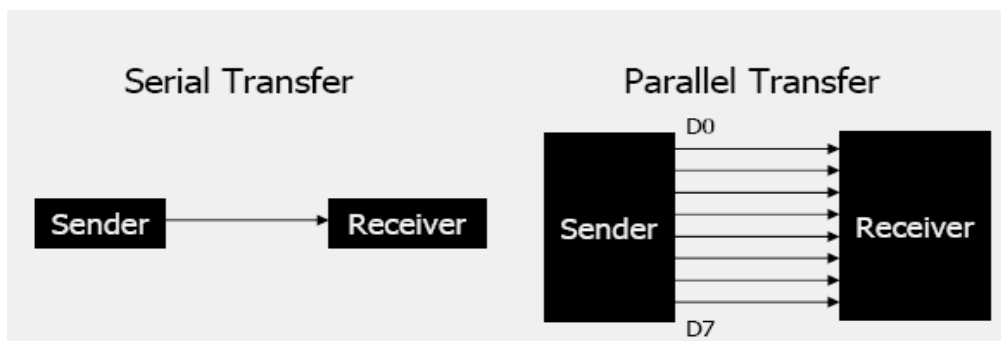


## 8051 Serial Communication

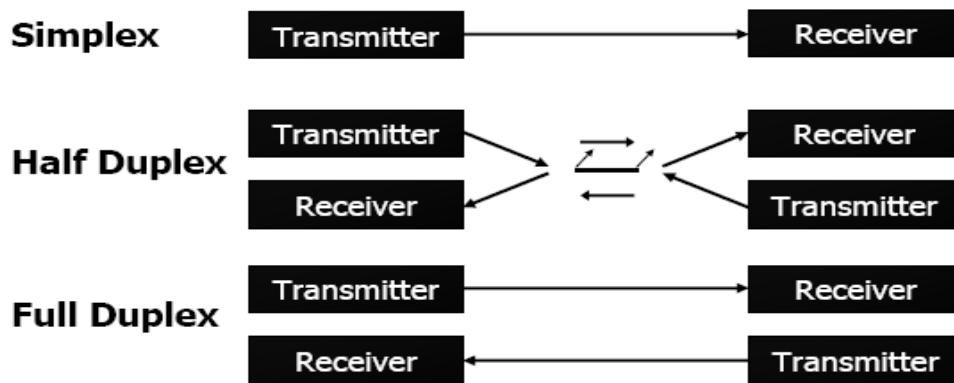
When a microcontroller communicates with the outside world, it provides the data in byte-sized chunks. In some cases, such as printers, the information is simply taken from the 8-bit data bus and presented to the 8-bit data bus of the printer. This can work only if the cable is not too long, since long cables diminish and even distort signals. Furthermore, an 8-bit data path is expensive. For these reasons, serial communication is used for transferring data between two systems located at distances of hundreds of feet to millions of miles apart. Fig(1) shows serial versus parallel data transfers. The fact that serial communication uses a single data line instead of the 8-bit data line of parallel communication. For serial data communication to work, the byte of data must be converted to serial bits using a parallel-in-serial-out shift register, then it can be transmitted over a single data line. At the receiving end there must be a serial-in-parallel-out shift register to receive the serial data and pack them into a byte.

When the distance is short, the digital signal can be transferred as it is on a simple wire and requires no modulation. However, for long-distance data transfers, serial data communication requires a modem to modulate (convert to 0s and 1s to audio tones) and demodulate (converting from audio tones to 0s and 1s).

Serial data communication uses two methods, asynchronous and synchronous. The synchronous method transfers a block of data(characters) at a time, while the asynchronous method transfers a single byte at a time. There are special IC's made by many manufacturers for serial data communications. These chips are commonly referred to as UART (universal asynchronous receiver transmitter) and USART (universal synchronous asynchronous receiver transmitter).



Fig(1)

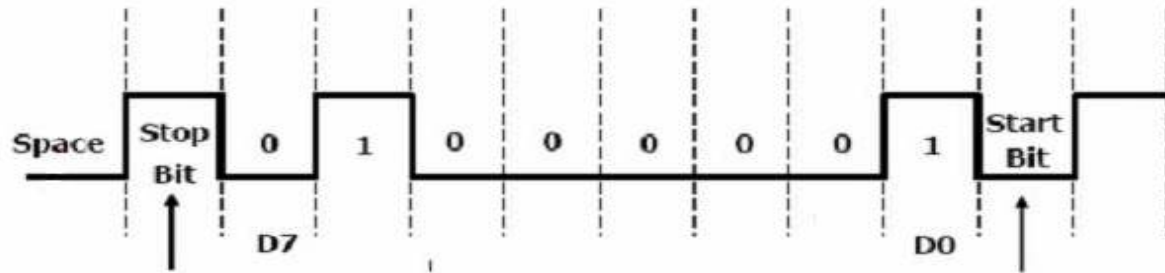


In data transmission if the data can be transmitted and received, it is a duplex transmission. This is in contrast to simplex transmission such as with printers, in which the computer only sends data. Duplex transmissions can be half or full duplex, depending on whether or not the data transfer can be simultaneous. If data is transmitted one way at a time, it is referred to as half duplex. If the data can go both ways at the same time, it is full duplex. Full duplex requires two wire conductors for the data lines, one for transmission and one for reception, in order to transfer and receive data simultaneously.

### **Asynchronous Serial Communication.**

The data coming in at the receiving end of the data line in a serial data transfer is all 0s and 1s, it is difficult to make sense of the data unless the sender and receiver agree on a set of rules, a protocol, on how the data is packed, how many bits constitute a character, and when the data begins and ends.

Asynchronous serial data communication is widely used for character oriented transmissions. In this method, each character is placed between start and stop bits. This is called framing. In data framing for asynchronous communications, the data such as ASCII characters, are packed between a start bit and stop bit. The start bit is always one bit, but the stop bit can be one or two bits. The start bit is always a 0 (low) and the stop bit is 1 (high). Look at below figure in which the ASCII character "A" ( 8-bit binary 0100 0001) is framed between the start bit and a stop bit. Notice that the LSB is sent out first.



Figure(2)

### Data Transfer Rate

The rate of data transfer in serial data communication is stated in bps ( bits per second). Another widely used terminology for bps is baud rate. The baud rate is the modem terminology and is defined as the number of signal changes per second.

### RS 232 standards

RS 232 is an interfacing standard and is the most widely used serial I/O interfacing standard. Since the standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible. For this reason, to connect any RS232 to a microcontroller system we must use voltage converters such as MAX232 to convert the TTL logic levels to RS232 voltage levels, and vice versa. MAX232 IC's are commonly referred to as line drivers.

### RS232 pins

RS232 cable commonly referred to as the DB-25 connector has 25 pins. Since not all the pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard, which uses 9 pins only, as shown in Table 1. The DB-9 pins are shown in Fig(3).

Current terminology classifies data communication equipment as DTE (data terminal equipment) or DCE (data communication equipment). DTE refers to terminals and computers that send and receive data, while DCE refers to communication equipment, such as modems, that responsible

for transferring the data. The simplest connection between a PC and microcontroller requires a minimum of three pins, TxD, RxD and ground as shown in fig(4).

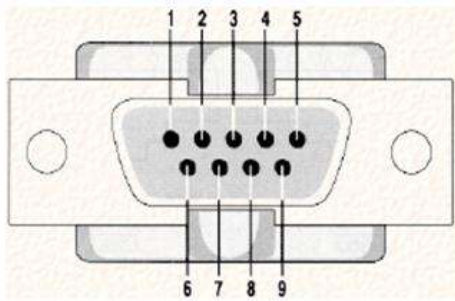
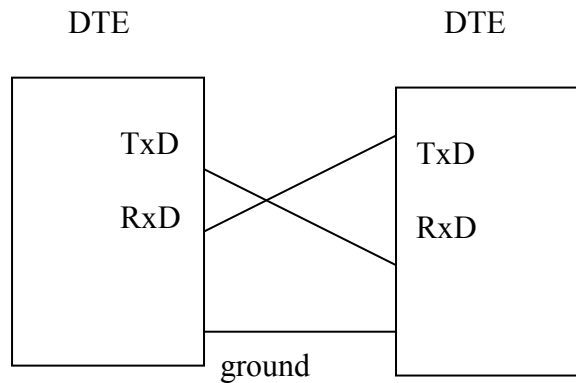


Fig (3)



Fig(4)

### RS232 handshaking signals

To ensure fast and reliable data transmission between two devices, the data transfer must be coordinated. Many of the pins of the RS232 connector are used for handshaking signals. Their descriptions are given below.

**DTR(Data Terminal Ready):** When a terminal is turned on, after going through a self-test, it sends out signal DTR to indicate that it is ready for communication.

**DSR (Data Set Ready):** When DCE (modem) is turned on and has gone through the self-test, it asserts DSR to indicate that it is ready to communicate.

**RTS (Request To Send):** When the DTE device ( such as PC) has a byte to transmit, it asserts RTS to signal the modem that it has a byte of data to transmit. RTS is an active-low output from the DTE and an input to the modem.

**CTS (Clear To Send):** In response to RTS, when the modem has room for storing the data it is to receive, it sends out signal CTS to the DTE to indicate that it can receive the data now.

| Pin | Description                |
|-----|----------------------------|
| 1   | Data carrier detect (-DCD) |
| 2   | Received data (RxD)        |
| 3   | Transmitted data (TxD)     |
| 4   | Data terminal ready (DTR)  |
| 5   | Signal ground (GND)        |
| 6   | Data set ready (-DSR)      |
| 7   | Request to send (-RTS)     |
| 8   | Clear to send (-CTS)       |
| 9   | Ring indicator (RI)        |

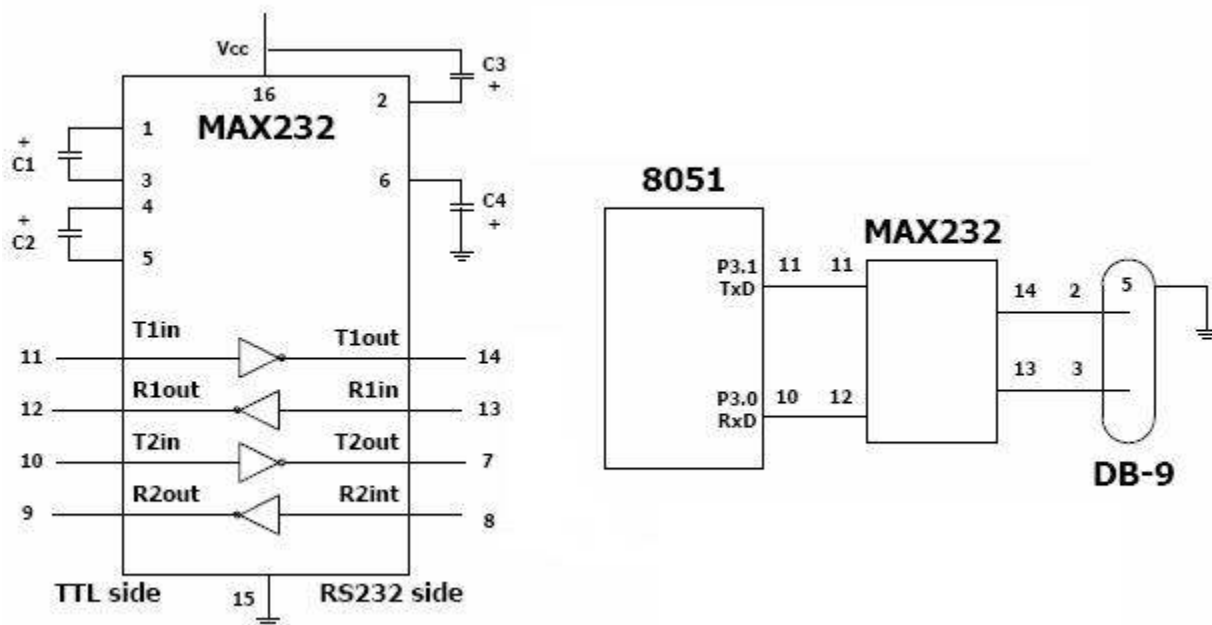
**DCD ( Data Carrier Detect):** The modem asserts signal DCD to inform the DTE that a valid carrier has been detected and that contact between it and the other modem is established. Therefore , DCD is an output from the modem and an input to the PC.

**RI ( Ring Indicator):** It indicates that the telephone is ringing. However if the PC is in charge of answering the phone, this signal can be used.

### MAX232

Since the RS232 is not compatible with today’s microprocessors and microcontrollers, we need a voltage converter(line driver) to convert the RS232’s signals to TTL voltage levels that will be acceptable to the 8051’s TxD and RxD pins. One example of such converter is MAX232. It converts from RS232 voltage levels to TTL voltage levels and vice versa. One advantage of the MAX232 is that it uses a +5 V power source which is the same as the source voltage for the 8051.

The MAX232 has two sets of line drivers for transferring and receiving data as shown in fig(5). The line drivers used for TxD are called T1 and T2, while the line drivers for RxD are designated as R1 and R2. MAX232 requires four capacitors ranging from 1 to 22 μF. The most widely used value for these capacitors is 22 μF.



Fig(5)

## **8051 SERIAL PORT PROGRAMMING**

### **Baud rate in the 8051**

The 8051 transfers and receives data serially at many different baud rates. The baud rate in the 8051 is programmable. This is done with the help of Timer 1. The relationship between the crystal frequency and the baud rate in the 8051 is shown below.

The 8051 divides the crystal frequency by 12 to get the machine cycle frequency. In the case XTAL = 11.0592 MHz, the machine cycle frequency is 921.6 KHz ( $11.0592\text{MHz}/12 = 921.6 \text{ KHz}$ ). The 8051's serial communication UART circuitry divides the machine cycle frequency of 921.6kHz by 32 once more before it is used by Timer 1 to set the baud rate. Therefore, 921.6 kHz divided by 32 gives 28,800 Hz.

### **SBUF REGISTER**

SBUF is an 8-bit register used solely for serial communication in the 8051. For a byte of data to be transferred via the TxD line, it must be placed in the SBUF register. Similarly, SBUF holds the byte of data when it is received by the RxD line. SBUF can be accessed like any other register in the 8051.

### **SCON (Serial Control) REGISTER**

The SCON register is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things.

### **Programming the 8051 to transfer data serially**

In programming the 8051 to transfer character bytes serially, the following steps must be taken.

1. The TMOD register is loaded with the value 20H, indicating the use of Timer 1 in mode 2 to set the baud rate.
2. The TH1 is loaded with one of the values in Table (2) to set the baud rate for serial data transfer.
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8-bit data is framed with start and stop bits.
4. TR1 is set to 1 to start Timer 1.

5. TI is cleared by the “CLR TI” instruction .
6. The character byte to be transferred serially is written into the SBUF register.
7. The TI flag bit is monitored with the use of the instruction “JNB TI, xx” to see if the character has been transferred completely.
8. To transfer the next character, go to Step 5.

**Write a program for the 8051 to transfer letter “A” serially at 4800 baud, continuously.**

**Solution:**

```

MOV    TMOD,#20H    ;timer 1,mode 2(auto reload)
MOV    TH1,#-6      ;4800 baud rate
MOV    SCON,#50H    ;8-bit, 1 stop, REN enabled
SETB   TR1          ;start timer 1
AGAIN: MOV  SBUF,#"A" ;letter "A" to transfer
HERE:  JNB  TI,HERE  ;wait for the last bit
       CLR  TI       ;clear TI for next char
       SJMP AGAIN    ;keep sending A

```

**Programming the 8051 to receive data serially**

In programming the 8051 to transfer character bytes serially, the following steps must be taken.

1. The TMOD register is loaded with the value 20H, indicating the use of Timer 1 in mode 2 to set the baud rate.
2. TH1 is loaded with one of the values in Table(2) to set the baud rate.
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where 8-bit data is framed with start and stop bits and receive enable is turned on.
4. TR1 is set to 1 to start Timer 1.
5. RI is cleared with the “CLR RI” instruction.
6. The RI flag bit is monitored with the use of the instruction “JNB RI, xx” to see if an entire character has been received yet.
7. When RI is raised, SBUF has the byte. Its contents are moved into safe place.
8. To receive the next character, go to Step 5.

Write a program for the 8051 to receive bytes of data serially, and put them in P1, set the baud rate at 4800, 8-bit data, and 1 stop bit

**Solution:**

```
MOV    TMOD,#20H    ;timer 1,mode 2(auto reload)
MOV    TH1,#-6      ;4800 baud rate
MOV    SCON,#50H    ;8-bit, 1 stop, REN enabled
SETB   TR1          ;start timer 1
HERE:  JNB    RI,HERE ;wait for char to come in
MOV    A,SBUF        ;saving incoming byte in A
MOV    P1,A         ;send to port 1
CLR    RI           ;get ready to receive next
                          ;byte
SJMP   HERE         ;keep getting data
```