REVIEW QUESTIONS.

8.Contrast and compare serial versus parallel communication.

Solution:

Serial Communication	Parallel communication
1. Sending or receiving data bit by bit is called serial communication.	1. Sending or receiving data in groups of bits at a time is called parallel communication.
2.Speed of data transfer is slow comparably.	2. Speed of data transfer is fast comparably .
3.Single line is used between two communicating devices.	3.Multiple lines are used between two communicating devices.
4.As single line is used circuitry cost is less.	4.As multiple lines are used circuitry cost is more.
5.Used for long distance applications.	5.Used for short distance applications.
6.The bandwidth of serial wires is much higher.	6.The bandwidth of parallel wires is much lower.

9.Explain serial communication protocol.

Solution:

Protocol is a set of rules agreed by the sender and receiver on how the data is packed, how many

bits constitute a character and also when the data begins and ends.

There are any serial comunication protocols such as UART/USART,CAN,USB,I2C etc.

UART protocol is used in 8051 microcontroller(Physical lines such as Rxd and Txd are used to identify the UART protocol).

UART stands for Univeral asynchronous Reciever-Transmitter and is mostly used for character-oriented transmissions.

1. Each character is placed in between start and stop bits, this is called data framing.

2.· The start bit is always one bit, but the stop bit can be one or two bits.

3.· The start bit is always a 0 (low) and the stop bit(s) is 1 (high).

Advantages:

1.The clock is not shared between transmitter and reciever.

2.There is tolerance to clock frequency hence data transfer occurs with more accuracy.

3.This type of data framing provides self synchronisation.

4.It helps in maintaining data integrity by including the parity bit in the data frame.

5.UART chips allow programming of the parity bit for odd-,even- and no parity options.

Disadvantage:

1.For transferring a character of 8 bits minimum of 10 bits is required(if 1 stop bit and 1 start bit is used) This gives 25% overhead, i.e. each 8-bit character with an extra 2 bits.

Pins Used:

Rxd Pin10(P3.0)

Txd Pin11(P3.1)

10. Explain the RS232 standard and its connection to 8051.

Solution:

RS232 is standard that is used to interface DTE(data transfer equipment) and DCE(data communication equipment).The DTE refers to the terminals and computers that send and receive data. The DCE refers to communication equipment such as modems and controllers. It is commonly used for transferring and receiving the serial data between two devices. It supports both synchronous and asynchronous data transmissions.The RS232 standard for communication was set by the Electronics

Industries Association in 1960. RS232 cable is used to identify the difference of two signal levels between logic 1 and logic 0.

Mechanical component:

The most commonly used connector is the DB25 connector. However since not all of the pins were used for PC cables, IBM came up with the DB9 version of the serial I/O standard.

Voltage component:

Logic 1 is represented by -3 to-25 volts. The Logic 0 is represented by 3 to 25 volts. Hence leaving 3 to -3 voltage undefined. The outputs are not TTL compatible.

Maximum cable length:50ft.

Maximum Baud rate:20KBPS.

Functional Component:

PIN NUMBER	NAME	FUNCTION
20	DTR (data terminal ready)	When terminal is turned on, it sends out signal DTR to indicate that it is ready for communication.
6	DSR (data set ready)	When DCE is turned on and has gone through the self-test, it assert DSR to indicate that it is ready to communicate.
4	RTS (request to send)	When the DTE device has byte to transmit, it assert RTS to signal the modem that it has a byte of data to transmit.

5	CTS (clear to send)	When the modem has room for storing the data it is to receive, it sends out signal CTS to DTE to indicate that it can receive the data now.
8	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.
7	-	Signal ground.
1	Gnd(ground)	Protective ground.
2	Txd	Transmit data to DCE
3	Rxd	Transmit data from DCE

8051 connection with RS 232

8051 outputs are TTL compatible whereas RS232 outputs are not TTL compatible .So some level convertors are used to convert RS232 levels to TTL level and vice-versa.

A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa. 8051 has two pins that are used specifically for transferring and receiving data serially. These two pins are called TxDand RxDand are part of the port 3 group (P3.0 and P3.1). These pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible .

11.Describe the Serial Communication features of the 8051.

Solution:

1.Full duplex communication.

2.Recieve buffered(i.e reading data from buffer and recieving serial data from serial line can take place simultaneously upto one byte)

3.**SBUF** is an 8-bit register used solely for serial communication in 8051. For a byte data to be

transferred via the TxD line, it must be placed in the SBUF register. The moment a byte is

written into SBUF, it is framed with the start and stop bits and transferred serially via the TxD

line. SBUF holds the byte of data when it is received by 8051 RxD line. When the bits are

received serially via RxD, the 8051 deframes it by eliminating the stop and start bits, making a

byte out of the data received, and then placing it in SBUF.

4.**SCON** is a bit addressable register used to set the mode in which serial communication takes

place in 8051. It has SM0, SM1, SM2, REN, TB8, RB8, TI, RI.

SM0, SM1: Serial Mode control Bits. They determine the framing of data by specifying the

number of bits per character, and the start and stop bits. Serial Mode 1 ensure 8 bit data

transmission and 1 start and stop bit.

SM2: Multiprocessor mode control bit, logic 1 enables Multi processor mode and 0 for normal

mode.

REN: Enables Serial reception. If set, it enables the reception otherwise the reception is

disabled.

TI: It is known as the Transmit Interrupt flag which is set by hardware to indicate the end of a

transmission. It has to be cleared by the software.

RI: It is known as Receive Interrupt flag which is set by hardware to indicate the end of a reception. It has to be cleared by the software.

5.Baud Rate is defined as number of bits transmitted or received per second and usually expressed in Bits per second bps. For mode 0 and mode 2 the baud rate is determined by means of 1/12, 1/32 or 1/64 of crystal frequency whereas for mode 1 and 3 it is determined by means of timer 1.Baud rate is programmable.

6.Four different modes of operation.

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift Register	1/12 of oscillator frequency.
0	1	1	8-bit UART	variable.
1	0	2	9-bit UART	1/32 of oscillator frequency -or- 1/64 of oscillator frequency.
1	1	3	9-bit UART	variable.

12.Write an assembly program to transfer letter ‘Y’ serially at 9600 baud continuously and also send a

letter ‘N’ to Port 0 which is connected to the display device.

Solution:

```
MOV TMOD, #20H
MOV TH1, #-3
MOV SCON, #50H
SETB TR1
AGAIN: MOV SBUF, #’Y’
HERE: JNB TI, HERE
CLR TI
```

```
MOV P0, #'N'
SJMP AGAIN
```

13.Write an 8051 C program to send two messages "Normal Speed" and "High Speed" to the serial

port. Assuming SW is connected to pin 2.0. Monitor its status and set baud rate as follows

1) SW=0; Baud Rate= 28800

2) SW=1; Baud Rate= 56000

Assume Crystal Frequency 11.0592 MHz.

Solution:

```
#include <reg51.h>
sbit MYSW=P2^0; //input switch
void main(void){
unsigned char z;
unsigned char Mess1[]="Normal Speed";
unsigned char Mess2[]="High Speed";
TMOD=0x20; //use Timer 1, mode 2
TH1=0xFF; //28800 for normal
SCON=0x50;//mode 1 and recieve enabled;
TR1=1; //start timer
if(MYSW==0) {
for (z=0;z<12;z++) {
SBUF=Mess1[z]; //place value in buffer
while(TI==0); //wait for transmit
```

```
TI=0;
}}
else {
PCON=PCON|0x80; //to double the speed 8th bit is set high in PCON.
for (z=0;z<10;z++) {
SBUF=Mess2[z]; //place value in buffer
while(TI==0); //wait for transmit
TI=0;
}
}
}
```