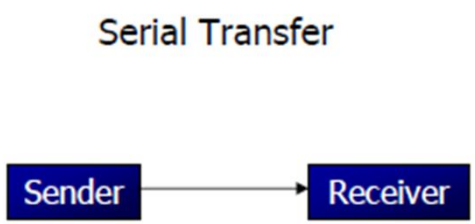
Microprocessors and Microcontrollers (EE-231)

Lab-9

Objective

- Learn Serial Communication
- To learn to program Serial Communication in C
  - In Proteus
  - On 8051 development board

Serial Communication



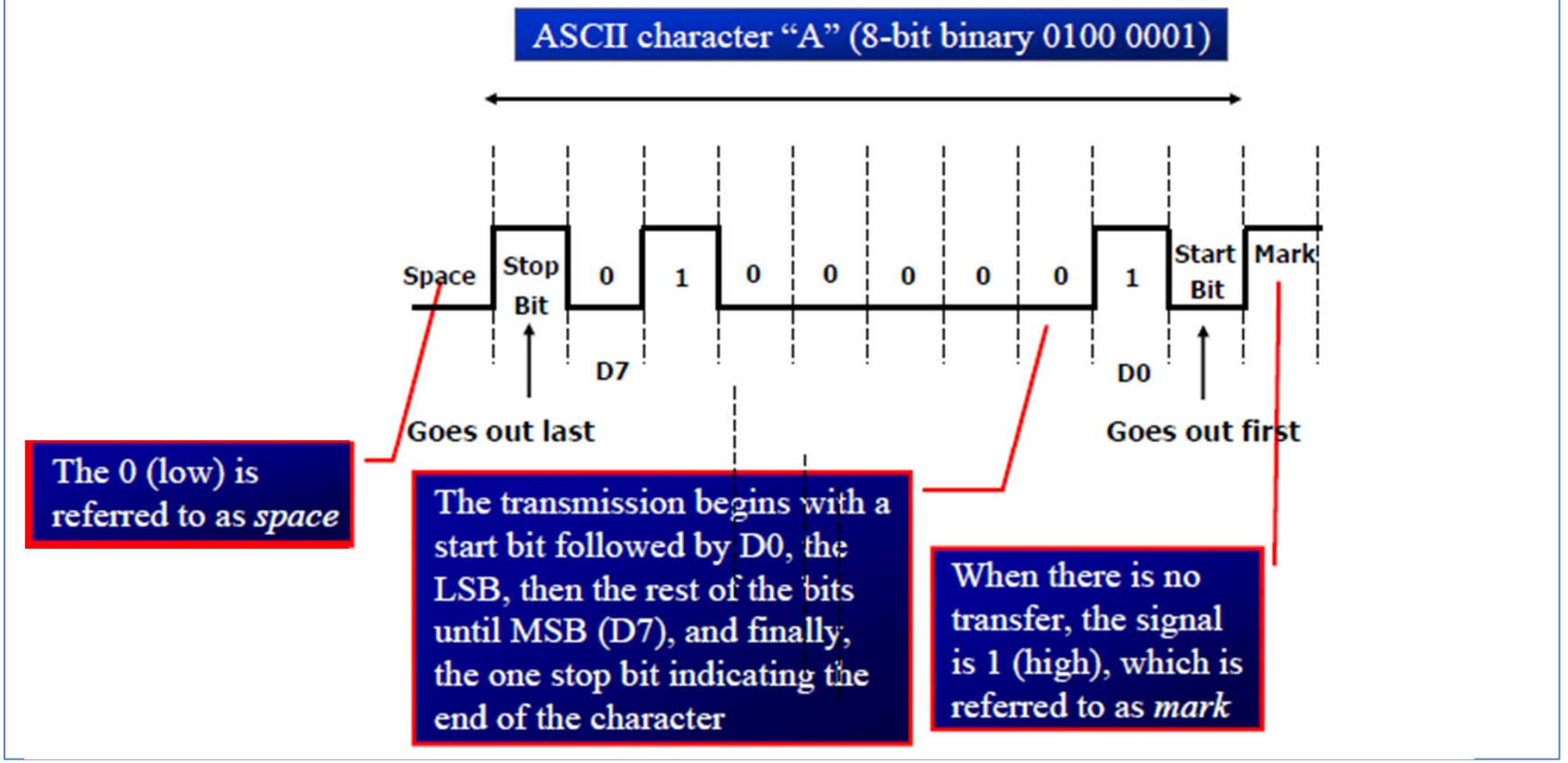
- To transfer to a device located many meters away, the serial method is used
- An Efficient way of data transfer
- At the transmitting end, the byte of data must be converted to serial bits using parallel-in-serial-out shift register
- At the receiving end, there is a serial-in-parallel-out shift register to receive the serial data and pack them into byte

Serial Communication

- Serial data communication uses two methods
- **Synchronous** method transfers a block of data at a time
- **Asynchronous** method transfers a single byte at a time
- There are special IC chips made by many manufacturers for serial communications
- **UART** (universal asynchronous Receiver transmitter)
- **USART** (universal synchronous-asynchronous Receiver-transmitter)
- Asynchronous serial data communication is widely used for character-oriented transmissions
- Each character is placed in between start and stop bits, this is called **framing**

Serial Communication Protocol

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

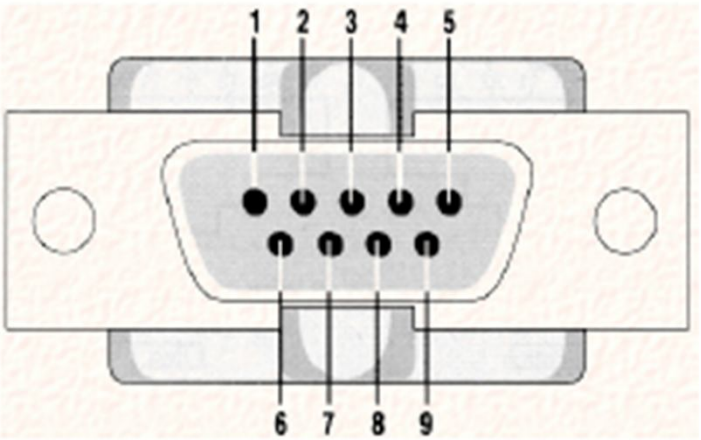


Serial Communication

- The data transfer rate of given computer system depends on communication ports incorporated into that system
- IBM PC/XT could transfer data at the rate of 100 to 9600 bps
- Pentium-based PCs transfer data at rates as high as 56K bps
- An interfacing standard **RS232** was set by the Electronics Industries Association (EIA) in 1960
- The standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible
- In RS232, a 1 (Mark) is represented by -3 ~ -25 V, while a 0 (Space) bit is +3 ~ +25 V, making -3 to +3 undefined

Serial Communication

RS232 Connector DB-9

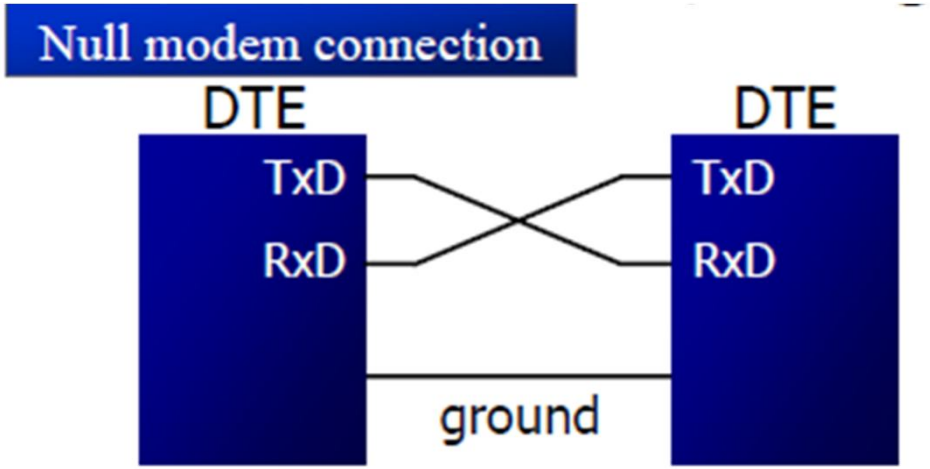


RS232 DB-9 Pins

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)

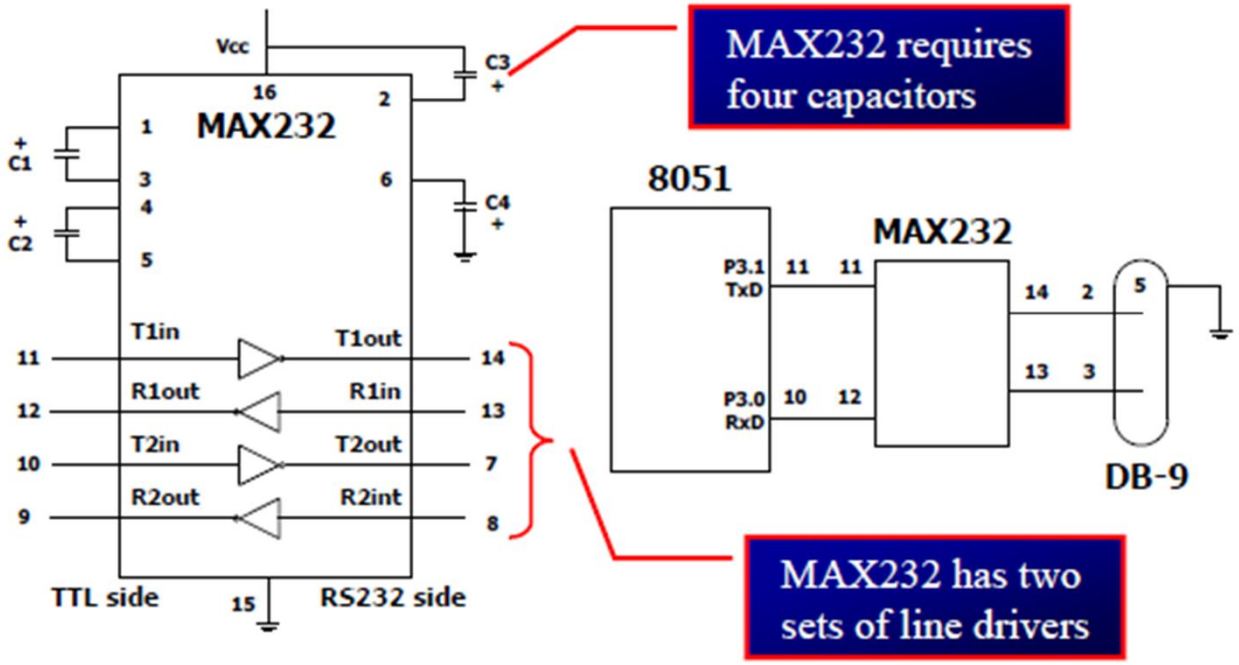
Serial Communication

- The simplest connection between a PC and microcontroller requires a minimum of three pins, TxD, RxD, and ground



Serial Communication

- A line driver such as the **MAX232** chip is required to convert RS232 voltage levels to TTL levels, and vice versa



- There is another version Max 233 which doesn't require capacitors

Baud Rate

- In 8051 baud rate is set using Timer1 in mode 2.

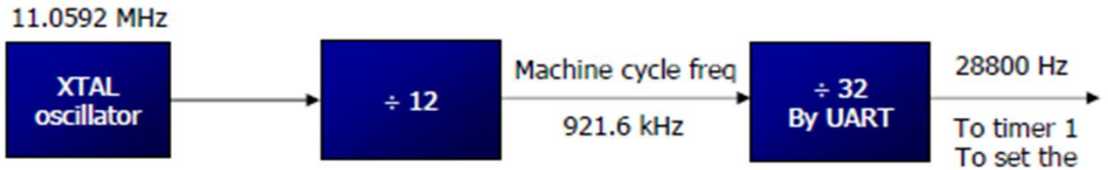
With XTAL = 11.0592 MHz, find the TH1 value needed to have the following baud rates. (a) 9600 (b) 2400 (c) 1200

Solution:

The machine cycle frequency of 8051 = 11.0592 / 12 = 921.6 kHz, and 921.6 kHz / 32 = 28,800 Hz is frequency by UART to timer 1 to set baud rate.

(a) 28,800 / 3 = 9600 where -3 = FD (hex) is loaded into TH1
(b) 28,800 / 12 = 2400 where -12 = F4 (hex) is loaded into TH1
(c) 28,800 / 24 = 1200 where -24 = E8 (hex) is loaded into TH1

Notice that dividing 1/12 of the crystal frequency by 32 is the default value upon activation of the 8051 RESET pin.



Baud Rate	TH1 (Decimal)	TH1 (Hex)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

TF is set to 1 every 12 ticks, so it functions as a frequency divider

SBUF

- SBUF is an 8-bit register used solely for serial communication
- 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

SCON

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

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication
REN	SCON.4	Set/cleared by software to enable/disable reception
TB8	SCON.3	Not widely used
RB8	SCON.2	Not widely used
TI	SCON.1	Transmit interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW
RI	SCON.0	Receive interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW

Note: *Make SM2, TB8, and RB8 =0*

SCON

SM0	SM1	
0	0	Serial Mode 0
0	**1**	**Serial Mode 1, 8-bit data, 1 stop bit, 1 start bit**
1	0	Serial Mode 2
1	1	Serial Mode 3

Only mode 1 is of interest to us

- SM2 enables the multiprocessing capability of the 8051

SCON

- **REN (receive enable)**
- When it is high, it allows 8051 to receive data on RxD pin
- If low, the receiver is disable
- **TI (transmit interrupt)**
- When 8051 finishes the transfer of 8-bit character
- It raises TI flag to indicate that it is ready to transfer another byte
- TI bit is raised at the beginning of the stop bit
- **RI (receive interrupt)**
- When 8051 receives data serially via RxD, it gets rid of the start and stop bits and places the byte in SBUF register
- It raises the RI flag bit to indicate that a byte has been received and should be picked up before it is lost
- RI is raised halfway through the stop bit

Tx Programming

- In programming the 8051 to transfer character bytes serially

1. TMOD register is loaded with the value 20H, indicating the use of timer 1 in mode (8-bit auto-reload) to set baud rate
2. The TH1 is loaded with one of the values to set 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
6. The character byte to be transferred serially is written into SBUF register
7. The TI flag bit is monitored to see if the character has been transferred completely
8. To transfer the next byte, go to step 5

Tx Programming

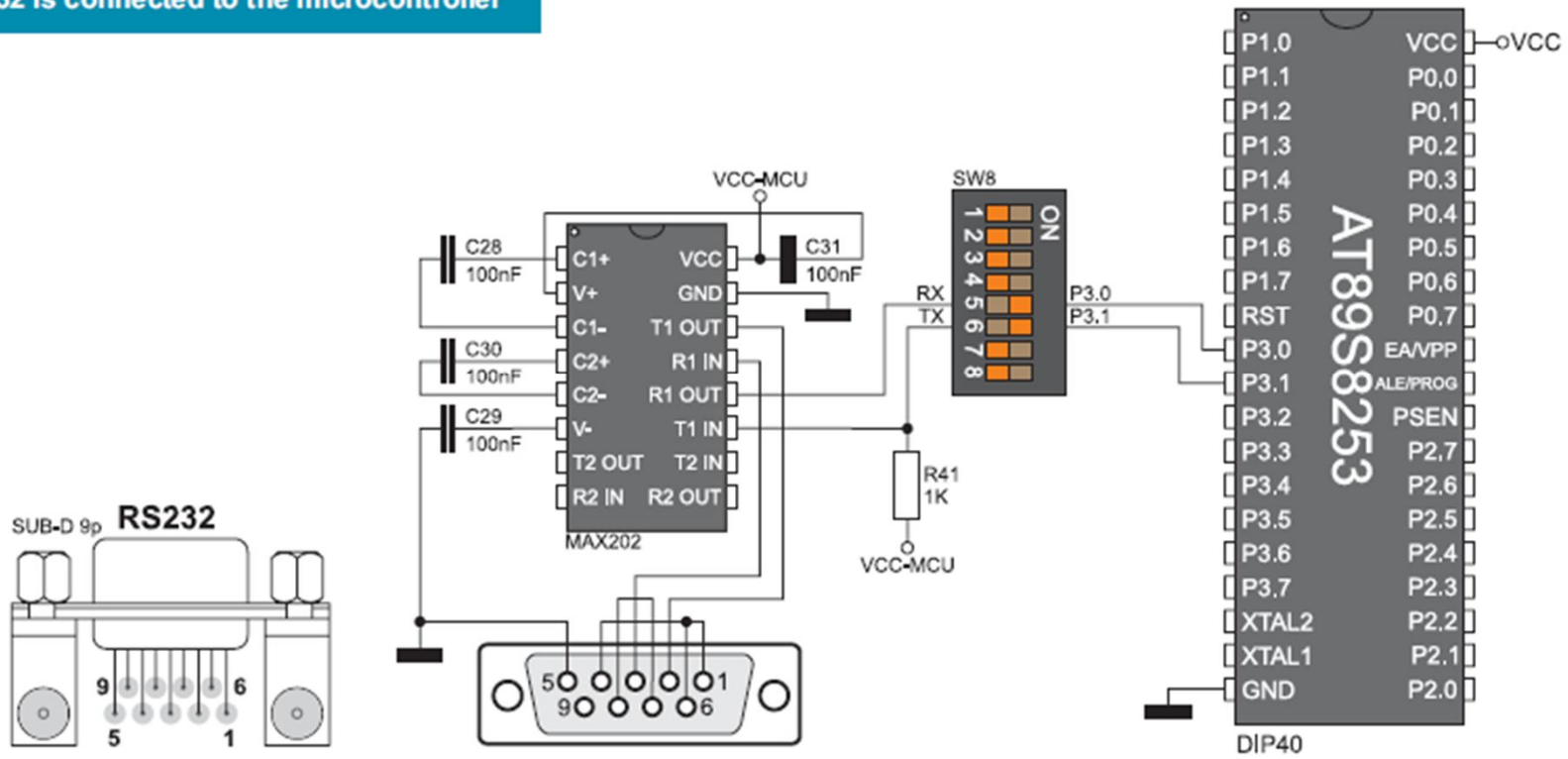
Example 10-15
Write a C program for 8051 to transfer the letter "A" serially at 4800 baud continuously. Use 8-bit data and 1 stop bit.

Solution:

```
#include <reg51.h>
void main(void){
  TMOD=0x20;          //use Timer 1, mode 2
  TH1=0xFA;           //4800 baud rate
  SCON=0x50;
  TR1=1;
  while (1) {
    SBUF='A';         //place value in buffer
    while (TI==0);
    TI=0;
  }
}
```

On Board Connections

Port RS-232 is connected to the microcontroller



Todays Task

- Transmit **Your name** using serial Communication to Computer.
- Simulate the design in Proteus.

- Task 2:

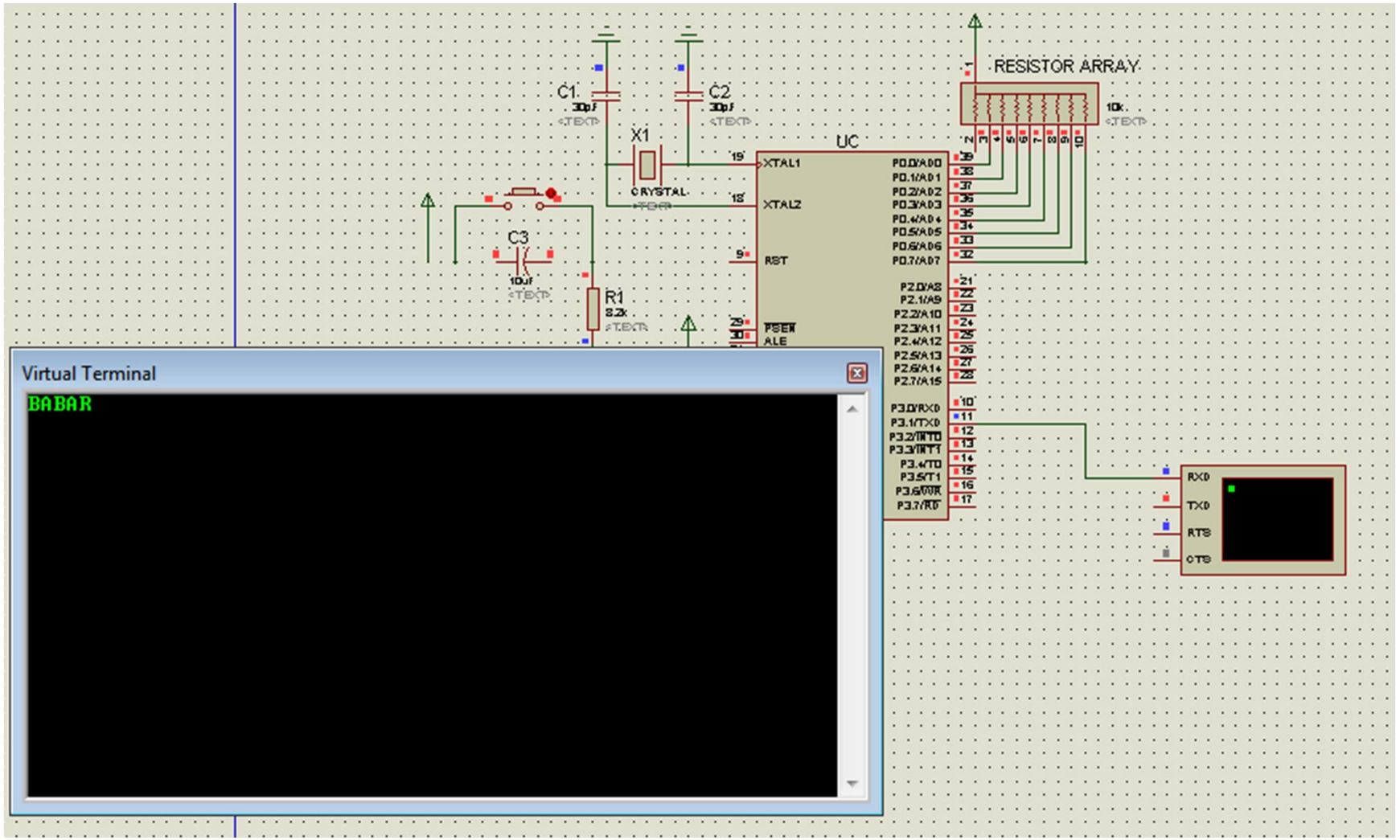
- Send this phrase to computer using UART,
- “ Hello Computer My name is XXXXX ”

Task Code

```
#include <reg51.h>
void TX(unsigned char);
void main(void){
TMOD=0x20; //use Timer 1, mode 2
TH1=0xFA; //4800 baud rate
SCON=0x50;
TX('B');
TX('A');
TX('B');
TX('A');
TX('R');
while(1);
}

void TX(unsigned char mydata)
{
TR1=1;
SBUF=mydata;
while (TI==0);
TI=0;
}
```

Proteus Simulation



Task 2 Code

```
#include <reg51.h>
void TX(unsigned char);
void main(void){
code unsigned char Mystring[]="Hello Computer my name is XXXX";
unsigned char x;
TMOD=0x20; //use Timer 1, mode 2
TH1=-6; //4800 baud rate
SCON=0x50;
for (x=0;x<30;x++)
  TX(Mystring[x]);

while(1);
}

void TX(unsigned char mydata)
{
TR1=1;
SBUF=mydata;
while (TI==0);
TI=0;
}
```

Proteus Simulation