MICROCONTROLLERS AND MICROPROCESSORS

Assignment 2

Mis : 111909002

Name: Abhijeet Deshmukh

[GitHub Repository for entire assignment with code, proteus files, hex files.](https://github.com/Sarabhian/Microcontroller_Lab)

# Q1 . Write a program to transmit character “A” continuously on serial line of microcontroller. Receive data serially using hyper terminal (serial terminal) and display

## code

#include <SST89X5XXRD2.H>

unsigned char B;

void main (void)

{

TMOD=0x20 ; // Timer 1 Mode 2

SCON=0x50; // REN enable ,mode 1

TH1= -3; //baud rate= 9600

TR1=1; //start timer

while(1)

{

SBUF='A'; //placing the input value

while(TI==0); // waiting till stop bit it 1

TI=0;

}

}

## simulation output

Shape

Description automatically generated with medium confidence

# Q2. Write a C program for a stopwatch using hyper-terminal and LCD. - S command for START the watch. - E command for STOP the watch. - P command for PAUSE. - C command for continue. Output should display time in Minutes and Seconds on the PC.

## code

[MICRO\_Assignment2.docx](https://coepac-my.sharepoint.com/:w:/g/personal/deshmukhar19_instru_coep_ac_in/EVeJvyFOBMBKstcN53OOPEwB53WGFg1h8vPpgUh3xOfezw?e=5jsrkh) if code won’t run and simulation is uploaded in the link. Please refer this doc of updated code.

#include <SST89X5XXRD2.H>

#define LCDPORT P1

sbit RS = P2^0;

sbit RW = P2^1;

sbit E = P2^2;

//

// initialisation of variables

unsigned char d[] = {"DIGITAL CLOCK"};

unsigned char c[] = {":"};

unsigned char msb, lsb;

unsigned int i, j, k, l;

//declaring functions

void LCD\_data (unsigned char); // // loading data in LCD

void LCD\_cmd (unsigned char); // for LCD to go in command mode

void enable (void); // enable pin

void LCD\_init (void); // initialising LCD

void delay (unsigned int); // adds delay

int msb\_lsb(unsigned int ); // finds msb and lsb of number

void SerialInitialize(void);

void SendByteSerially(unsigned char);

void port\_init(void);

sbit start = P2^3;

sbit ending = P2^4;

void main()

{

port\_init();

SerialInitialize();

EA = 1;

ES = 1;

if( start == 1 )

{

LCD\_init(); //INITIALIZATION OF LCD

while(1)

{

LCD\_cmd(0x82); // coursor line 1 position 2

for(l=0; l<13; l++) // print "DIGITAL CLOCK" on display

{

LCD\_data(d[l]);

}

for(i=0; i<60; i++)

{

if(ending == 1) break;

// printing for hour

LCD\_cmd(0xC3); // Jump cursor to second line position 3

msb, lsb = msb\_lsb(i);

LCD\_data(msb + 0x30);

LCD\_data(lsb + 0x30);

LCD\_data(c[0]);

for(j=0; j<60; j++)

{

if(ending == 1) break;

// printing for minutes

LCD\_cmd(0xC6); // Jump cursor to second line position 6

msb, lsb = msb\_lsb(j);

LCD\_data(msb + 0x30);

LCD\_data(lsb + 0x30);

LCD\_data(c[0]);

for(k=0; k<60; k++)

{

if(ending == 1) break;

// printing for seconds

LCD\_cmd(0xC9); // Jump cursor to second line position 9

msb, lsb = msb\_lsb(k);

LCD\_data(msb + 0x30);

LCD\_data(lsb + 0x30);

}

}

}

}

}

}

void delay (unsigned int ms)

{

unsigned int i,j;

for(j=0;j<ms;j++)

{

for(i=0;i<=1275;i++);

}

}

void LCD\_init (void)

{

LCD\_cmd(0x38); //5x7 matrix of cursor

LCD\_cmd(0x0E); //cursor blinking commnad curson ON

LCD\_cmd(0x06); //Auto increment right

LCD\_cmd(0x01); //clear screen command

}

void LCD\_data (unsigned char value)

{

RS=1; // Register select in data mode

RW=0; // Writing mode

LCDPORT=value; // writing value to port P1

enable();

delay(5);

}

void LCD\_cmd (unsigned char command)

{

RS=0;

RW=0;

LCDPORT=command;

enable();

delay(5);

}

void enable (void)

{

E=0; // we are creating a pulse here // helping to onn off display

E=1;

E=0;

}

int msb\_lsb(unsigned int val)

{

lsb = val % 10;

msb = val / 10;

return msb, lsb;

}

void port\_init(void) //initialize cct

{

P0 = 0x00; //not used

P1 = 0x00; //Used for Appliances

P2 = 0x00; //not used

P3 = 0x03; //used for serial P3.0 and P3.1 = 1

}

void SerialInitialize(void) // INITIALIZE SERIAL PORT

{

TMOD = 0x20; // Timer 1 IN MODE 2 -AUTO RELOAD TO GENERATE BAUD RATE

SCON = 0x50; // SERIAL MODE 1, 8-DATA BIT 1-START BIT, 1-STOP BIT, REN ENABLED

TH1 = 0xFD; // LOAD BAUDRATE TO TIMER REGISTER

TR1 = 1; // START TIMER

}

void SendByteSerially(unsigned char serialdata)

{

SBUF = serialdata; // LOAD DATA TO SERIAL BUFFER REGISTER

while(TI == 0); // WAIT UNTIL TRANSMISSION TO COMPLETE

TI = 0; // CLEAR TRANSMISSION INTERRUPT FLAG

}

void serial\_ISR (void) interrupt 4 // Timer Interrupt 1 TF1

{

//receive character

char chr;

if(RI == 1)

{

chr = SBUF;

RI = 0;

}

P0 = ~P0; //Show the data has been updated

switch(chr)

{

case 'S': start = 1; ending = 0; SendByteSerially('S'); break;

case 'E': ending = 1; start = 0; SendByteSerially('E'); break;

//case 'P': pause = 1; SendByteSerially('P'); break;

//case 'C': cont = 1; SendByteSerially('C'); break;

default: ; break; //do nothing

}

RI = 0;

}

## simulation output

Diagram, schematic

Description automatically generated

# Q3. Connect to microcontroller using null-modem connection and write a program to transfer data serially from one microcontroller and receive the same data in another microcontroller and display on the LCD.

## code

Add below codes hex file to receiving microcontroller

//for Microcontroller1

#include <SST89X5XXRD2.H>

unsigned char i;

void delay (unsigned int ms);

void main(void)

{

TMOD=0x20;

TH1=0xFD;

SCON=0x50;

TR1=1;

while(1)

{

SBUF='A';

while(TI==0);

delay(10);

TI=0;

}

}

void delay (unsigned int ms)

{

unsigned int i,j;

for(j=0;j<ms;j++)

{

for(i=0;i<=1000;i++);

}

}

**//Code 2**

#include <SST89X5XXRD2.H>

#define LCDPORT P1

sbit RS = P2^0;

sbit RW = P2^1;

sbit E = P2^2;

unsigned char A;

unsigned char d,k;

void LCD\_data (unsigned char);

void LCD\_cmd (unsigned char);

void enable (void);

void LCD\_init (void);

void delay (unsigned int);

void main (void)

{

LCD\_init(); //INITIALIZATION OF LCD

LCD\_cmd(0x80);

while(1)

{

TMOD=0x20;

TH1=0xFD;

SCON=0x50;

TR1=1;

while(RI==0);

d++;

A=SBUF;

RI=0;

LCD\_data(A);

}

}

void delay (unsigned int ms)

{

unsigned int i,j;

for(j=0;j<ms;j++)

{

for(i=0;i<=1000;i++);

}

}

void LCD\_init (void)

{

LCD\_cmd(0x38); //5x7 matrix of cursor

LCD\_cmd(0x0E); //cursor blinking commnad curson ON

LCD\_cmd(0x06); //Auto increment right

LCD\_cmd(0x01); //clear screen command

}

void LCD\_data (unsigned char value)

{

RS=1;

RW=0;

LCDPORT=value;

enable();

delay(5);

}

void LCD\_cmd (unsigned char command)

{

RS=0;

RW=0;

LCDPORT=command;

enable();

delay(5);

}

void enable (void)

{

E=0;

E=1;

E=0;

}

## simulation output

Shape

Description automatically generated with medium confidence