MICROCONTROLLERS AND MICROPROCESSORS

Assignment 3

Interrupt Programming

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 generate 1S delay function using Timer Interrupt method. Use that function in digital clock for to increment seconds count. Compare the performance of polling method and interrupt method for digital clock and comment.

## CODE

#include <SST89X5XXRD2.H>

// this code prints DIGITAL CLOCK in (HRs:MM:SS),

// pin configurations

#define LCDPORT P1 // declaring p1 as LCDPORT ( assigning newe variable )

sbit RS = P2^0;

sbit RW= P2^1;

sbit E= P2^2;

// initializing variables

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

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

unsigned char k,l ,m , lsb,msb;

// function declarations

void LCD\_data( unsigned char );

void LCD\_cmd(unsigned char);

void enable(void);

void LCD\_init(void);

void delay(unsigned int );

int msb\_lsb(int k);

// timer delay

sbit pin = P3^0;

unsigned char count;

void timer0\_isr() interrupt 1

{

TL0=0xFD;

TH0=0x4B;

TF0=1;

pin=~pin;

}

void timerdelay ()

{

// to create delay of 1s timer will delay 50ms for 20 times , 50ms/1.085= 46083 counts => 65536-46083= 19453

count=0;

while (count!=20)

{

TMOD=0x01;

TL0=0xFD;

TH0=0x4B;

TR0 = 1; // Start Timer

//while(TF0 == 0); // Polling for flag bit

IE=0x82; // interrupt method

//TR0 = 0; // Stop Timer

//TF0 = 0;

count++;

}

}

// main funcction

void main(void)

{

LCD\_init();

while(1)

{

LCD\_cmd(0x82);

for(k=0;k<=13;k++)

{

LCD\_data(e[k]);

}

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

{

LCD\_cmd(0xc3);

msb , lsb =msb\_lsb(k);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

LCD\_data(coln[0]);

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

{

LCD\_cmd(0xc6);

msb , lsb =msb\_lsb(l);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

LCD\_data(coln[0]);

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

{

LCD\_cmd(0xc9);

msb , lsb =msb\_lsb(m);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

// adding timer delay of 1s

timerdelay();

}

}

}

}

}

//function definations

// 1. delay function

void delay(unsigned int ms)

{

unsigned int i,j;

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

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

}

// 2. LCD\_init functon ... used for initializing LCD commands

void LCD\_init(void)

{

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

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

LCD\_cmd(0x06); // cursor Auto increament to right

LCD\_cmd(0x01); // clrear screen

}

// 3. LCD\_data ... data mode --> RS=1 , LCD is currently output device so R/W= 0 (i.e. here RW=0)

void LCD\_data(unsigned char value)

{

RS=1; RW=0;

LCDPORT= value;

enable();

delay(5);

}

// 4. LCD\_cmd ... command mode --> RS=0 , RW=0

void LCD\_cmd(unsigned char command )

{

RS=0;RW=0;

LCDPORT=command;

enable();

delay(5);

}

//5. enable ... used to generste pulse

void enable(void)

{

E=0;E=1;E=0;

}

// 6. msb\_lsb()

int msb\_lsb(int a)

{

msb = a/10;

lsb = a%10;

return msb ,lsb ;

}

## sIMULATION OUTPUT

Text

Description automatically generated

# q2. Create an application to show functioning of external interrupt0 (/INT0).

## CODE

#include <SST89X5XXRD2.H>

// port declaration

// initializing variables

unsigned char x ;

unsigned int k,m;

// function declarations

void delay\_ms(unsigned int);

// main function

void main(void)

{

while(1)

{

IE=0x81;

x = 0x01; // 00000001

for(k =0; k<8; k++)

{

P1 = x;

delay\_ms(100);

x = x<<1; // left shift

}

x=0x80;

for(k =0; k<8; k++)

{

P1 = x;

delay\_ms(100);

x = x>>1; // left shift

}

}

}

//function definations

// 1. delay function

void delay\_ms(unsigned int ms) // delay in micro second function

{

unsigned int j,i;

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

{

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

}

}

void external0\_isr() interrupt 0

{

P1=0x55; // alternate LEDswill blink

delay\_ms(100);

P1=0xAA; // alternate LEDswill blink

delay\_ms(100);

}

## SIMULATION OUTPUT

Text

Description automatically generated

# q3.Write a program to transfer your name on serial port by interrupt method

## CODE

#include <SST89X5XXRD2.H>

// in this code we will transfer Data using serial communication via interrupt method

unsigned char y[23] = {" Abhijeet Deshmukh "};

unsigned char x=0, i=0;

void baud\_rate (void)

{

TMOD = 0x20; // Timer one in Mode two (Auto Reload Mode)

TH1 = -3; // 9600 baud rate @ 11.0592 Mhz crystal

TR1=1;// Transfer Data to virtual terminal

while(!TF1); // Overflow condition in Timer 1 , Check the transfer flag for Data tranfer

TF1=0; // Timer 0 in running mode,Clear the tranfer flag for next round

}

void main (void)

{

SCON = 0x50; // REN ENABLE, TB8 = 0, RB8 =0, MODE 1 OF Serial Communication (10-Bit frame format) 1-start, 8- databits and 1-stop

EA=1; //Enable use of Register

ES=1; //Enable Serial Interrupt

baud\_rate ();

SBUF=y[i];

while(1);

}

void serial\_isr (void) interrupt 4

{

if (i>22)

(i=-1);

if(x==1)

{

SBUF=y[i];

i=i+1;

TI=0;

x=0;

}

else

{

SBUF=y[i];

i=i+1;

TI=0;

x=1;

}

}

## SIMULATION oUTPUT

Text

Description automatically generated

# q4.Consider you are using two or more than two interrupts; you are assigning high priority to external interrupt\_1 than timer\_0 interrupt. Create an application and show the importance of change in priority of the interrupts.

## CODE

#include <SST89X5XXRD2.H>

// this code prints DIGITAL CLOCK in (HRs:MM:SS), application of high priority interrupt.

// pin configurations

#define LCDPORT P1 // declaring p1 as LCDPORT ( assigning newe variable )

sbit RS = P2^0;

sbit RW= P2^1;

sbit E= P2^2;

// initializing variables

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

unsigned char d[]={"Count Reset "};

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

unsigned char k,l ,m,s1,sw , lsb,msb;

// function declarations

void LCD\_data( unsigned char );

void LCD\_cmd(unsigned char);

void enable(void);

void LCD\_init(void);

void delay(unsigned int );

int msb\_lsb(int k);

// timer delay

sbit pin = P3^0;

unsigned char count;

void timer0\_isr() interrupt 1

{

TL0=0xFD;

TH0=0x4B;

pin=~pin;

TF0=1;

}

void int1\_ISR() interrupt 2

{

LCD\_cmd(0x01); // clrear screen

LCD\_cmd(0x82);

for(k=0;k<=13;k++)

{

LCD\_data(e[k]);

}

LCD\_cmd(0xC2);

for(k=0;k<=12;k++)

{

LCD\_data(d[k]);

}

sw=1;

}

void timerdelay ()

{

// to create delay of 1s timer will delay 50ms for 20 times , 50ms/1.085= 46083 counts => 65536-46083= 19453

count=0;

while (count<=20)

{

TMOD=0x01;

TL0=0xFD;

TH0=0x4B;

TR0 = 1; // Start Timer

//while(TF0 == 0); // Polling for flag bit

// // Stop Timer

//

count++;

}

}

// main funcction

void main(void)

{

EA=1 ; // enable interrupt

ET0=1; // timer 0 interrupt

EX1=1; // external interrupt 1

PX1=1; // assign high priority to EXT1

LCD\_init();

while(1)

{

s1 :

sw=0;

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

LCD\_cmd(0x82);

for(k=0;k<=13;k++)

{

LCD\_data(e[k]);

}

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

{

LCD\_cmd(0xc3);

msb , lsb =msb\_lsb(k);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

LCD\_data(coln[0]);

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

{

LCD\_cmd(0xc6);

msb , lsb =msb\_lsb(l);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

LCD\_data(coln[0]);

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

{

LCD\_cmd(0xc9);

msb , lsb =msb\_lsb(m);

LCD\_data(msb+ 0x30);

LCD\_data(lsb +0x30);

// adding timer delay of 1s

timerdelay();

if(sw==1)

goto s1;

}

}

}

}

}

//function definations

// 1. delay function

void delay(unsigned int ms)

{

unsigned int i,j;

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

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

}

// 2. LCD\_init functon ... used for initializing LCD commands

void LCD\_init(void)

{

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

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

LCD\_cmd(0x06); // cursor Auto increament to right

LCD\_cmd(0x01); // clrear screen

}

// 3. LCD\_data ... data mode --> RS=1 , LCD is currently output device so R/W= 0 (i.e. here RW=0)

void LCD\_data(unsigned char value)

{

RS=1; RW=0;

LCDPORT= value;

enable();

delay(5);

}

// 4. LCD\_cmd ... command mode --> RS=0 , RW=0

void LCD\_cmd(unsigned char command )

{

RS=0;RW=0;

LCDPORT=command;

enable();

delay(5);

}

//5. enable ... used to generste pulse

void enable(void)

{

E=0;E=1;E=0;

}

// 6. msb\_lsb()

int msb\_lsb(int a)

{

msb = a/10;

lsb = a%10;

return msb ,lsb ;

}

## SIMULATION OUTPUT

Text

Description automatically generated