**Problem1: (using single interrupt)**

#include <mega32.h>

#include <delay.h>

#define led\_ddr DDRC

#define led\_port PORTC

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

led\_ddr=0xFF;

led\_port=0x03;

delay\_ms(1000);

led\_port=0x0C;

delay\_ms(1000);

led\_port=0x30;

delay\_ms(1000);

led\_port=0xC0;

delay\_ms(1000);

led\_port=0x00;

}

void main(void)

{

led\_ddr=0xFF;

GICR=(0<<INT1) | (1<<INT0) | (0<<INT2); //using inter0

MCUCR=(0<<ISC11) | (0<<ISC10) | (1<<ISC01) | (1<<ISC00); //using rising edge for int0

while (1)

{ #asm("sei")

led\_port=0x00;

}

}

**Problem2: (using double Interrupt)**

#include <mega32.h>

#include <delay.h>

#define data\_ddr DDRB

#define data\_port PORTB

char digit\_cathode[10] ={0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07,0x7F,0x6F};

char i;

// External Interrupt 0 service routine

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

PORTC=0x00;

#asm("sei")

PORTC.0=1;

PORTC.5=1;

PORTC.6=0;

If (PORTD.3==0){

PORTC.5=0;

PORTC.6=0;

PORTC.5=0;

PORTC.6=1;

delay\_ms(5000);}

delay\_ms(5000);

PORTC=0x00;

}

// External Interrupt 1 service routine

interrupt [EXT\_INT1] void ext\_int1\_isr(void)

{

PORTC=0x00;

PORTC.0=1;

PORTC.5=0;

PORTC.6=1;

delay\_ms(10000);

PORTC=0x00;

}

void main(void) {

data\_ddr=0xFF;

DDRC=0xFF;

PORTC=0x00;

GICR|=(1<<INT1) | (1<<INT0) | (0<<INT2); // as we are using 2 interrupts int0 and int1

MCUCR=(1<<ISC11) | (0<<ISC10) | (1<<ISC01) | (0<<ISC00); //falling edge trigger

MCUCSR=(0<<ISC2); // int2 not used

GIFR=(1<<INTF1) | (1<<INTF0) | (0<<INTF2);

#asm("sei")

while (1) {

for(i=0;i<10;i++)

{

data\_port=digit\_cathode[i];

delay\_ms(2000);

}

}

}

**Problem3: (7 segment display)**

#include <mega32.h>

#include <delay.h>

#define segment\_ddr DDRA

#define segment\_port PORTA

unsigned int cathode[10]={0x3F,0x06,0x5B,0x4F, 0x66,0x6D,0x7D,0x07,0x7F,0x6F};

unsigned int i=0;

interrupt [EXT\_INT2] void ext\_int0\_isr(void)

{

for (i=0;i<10;i++) {

segment\_port=cathode[i];

delay\_ms(1000);

}

segment\_port=0x3F;

}

void main(void)

{

GICR=(0<<INT1) | (0<<INT0) | (1<<INT2);

MCUCSR=(0<<ISC2);

DDRA=0xFF;

#asm("sei")

while (1)

{

segment\_port=0x3F;

}

}

**Problem4: (7 segment display with INT)**

#include <mega32.h>

#include <delay.h>

#define segment\_ddr DDRA

#define segment\_port PORTA

unsigned int cathode[10]={0x3F,0x06,0x5B,0x4F, 0x66,0x6D,0x7D,0x07,0x7F,0x6F};

unsigned int i=0;

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

for (i=0;i<10;i++) {

segment\_port=cathode[i];

delay\_ms(1000);

}

segment\_port=0x3F;

}

void main(void)

{

GICR=(0<<INT1) | (1<<INT0) | (0<<INT2); //using inter0

MCUCR=(0<<ISC11) | (0<<ISC10) | (1<<ISC01) | (1<<ISC00); //rising edge

segment\_ddr =0xFF;

#asm("sei")

while (1)

{

segment\_port=0x3F;

}

}

**Problem5: (motro interfacing)**

#include <mega32.h>

#include <delay.h>

#define ddr\_led DDRA

#define port\_led PORTA

interrupt [EXT\_INT1] void ext\_int1\_isr(void)

{

PORTC=0xFF;

#asm("sei")

PORTC.7=1;

PORTC.5=1;

PORTC.6=0;

delay\_ms(5000);

PORTC=0x00;

}

void main(void)

{

DDRC=0xFF;

ddr\_led=0xFF;

GICR|=(1<<INT1) | (0<<INT0) | (0<<INT2); // we are using interrput1

MCUCR=(0<<ISC11) | (1<<ISC10) | (0<<ISC01) | (0<<ISC00); // using change for interrupt 1

MCUCSR=(0<<ISC2); // not using interrupt2

GIFR=(1<<INTF1) | (0<<INTF0) | (0<<INTF2);

#asm("sei")

while (1)

{

port\_led=0xFF;

delay\_ms(1000);

port\_led=0x00;

delay\_ms(1000);

}

}

**Problem6: (Using timer and counter)**

#include <mega32.h>

interrupt [TIM0\_COMP] void timer0\_comp\_isr(void)

{

    TCNT0=0;

}

void main(void)

{

DDRC.0=1;

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (1<<CS02) | (0<<CS01) | (0<<CS00); // CS scaling factor 256 //WGM normal mode

TCNT0=0;

OCR0=156; //Output compare register value calculated from the question

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (1<<OCIE0) | (0<<TOIE0); //output compare enable

#asm("sei")

PORTC.0=0;

while (1)

      {

      }

}

**Problem7: (using timer and counter)**

#include <maga32.h>

interrupt [TIM1\_OVF]  void timer1\_ovf\_isr (void)

{

PORTB.4=~PORTB.4;

TCNT1L=0XB9;

TCNT1H=0XF0;

}

void main(void)

    {

TCNT1L=0XB9;

TCNT1H=0XF0;

TCCR1A=0x00;

TCCR1B=0x05;

             TIMSK=0x04;

             DDRB.4=1;

#asm(“sei”);

while (1) {

  }

}

**Problem 9: (timer with scaling)**

#include <mega32.h>

interrupt [TIM0\_COMP] void timer0\_comp\_isr(void)

{

    TCNT0=0;

}

void main(void)

{

DDRC.0=1;

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (1<<CS02) | (0<<CS01) | (1<<CS00); // CS scaling factor 1024 //WGM normal mode

TCNT0=0;

OCR0=78; //Output compare register value calculated from the question

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (1<<OCIE0) | (0<<TOIE0); //output compare method

#asm("sei")

PORTC.0=0;

while (1)

      {

      }

}

Problem 8: (timer with int1)

#include <mega32.h>

interrupt [TIM1\_COMPA] void timer1\_compa\_isr(void) //output compare timer1

{

    PORTB.7=~PORTB.7;

    TCNT1H=0x00; // as output compare so starting form zero

    TCNT1L=0x00; // as output compare so starting form zero

}

void main(void)

{

DDRB.7=1;

TCCR1A=0x00;

TCCR1B=0x03; //as prescalar is 64

TCNT1H=0x00;

TCNT1L=0x00;

OCR1AH=0x5B; // calculated form question first 2 digits of hex

OCR1AL=0x8D; //calculated form question last 2 digits of hex

TIMSK=0x10; //output compare OCF1A

#asm("sei")

while (1)

      {

      }

}