**MICROCONTROLLER AND MICROPROCESSOR LAB**

**EXPERIMENT 10 - A**

**AIM**: Write an embedded C program to display values from 0 to 9 on a 7-segment display interfaced with an 8051-microcontroller hardware kit.

**SOFTWARE USED**: Keil uVision5

**CODE**:

#include<reg51.h>

sbit a=P2^7;

sbit b=P2^6;

sbit c=P2^5;

unsigned int i=0,y=0,x=0,z=0,seg=0;

unsigned char array[16]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0xf7, 0x7c, 0x39, 0x5e, 0x79, 0x71};

void main()

{

TMOD=0x11;

IE=0x8a;

TH0=0xf3;

TL0=0Xfb;

TH1=0xf3;

TL1=0Xfb;

TR0=1;

TR1=1;

while(1)

{

if(y==1)

{

y=0;

P0=array[i];

}

if(z==1)

{

z=0;

i++;

if (i==10)

{

i=0;

seg++;

}

if(seg==0)

{

a=0;

b=0;

c=0;

}

if(seg==1)

{

a=0;

b=0;

c=1;

}

if(seg==2)

{

a=0;

b=1;

c=0;

}

if(seg==3)

{

a=0;

b=1;

c=1;

}

if(seg==4)

{seg=0;

}

}

}

}

void timer0\_isr() interrupt 1

{

y=1;

TH0=0xf3;

TL0=0xfb;

}

void timer1\_isr() interrupt 3

{

x++;

if(x==1000)

{x=0;

z=1;

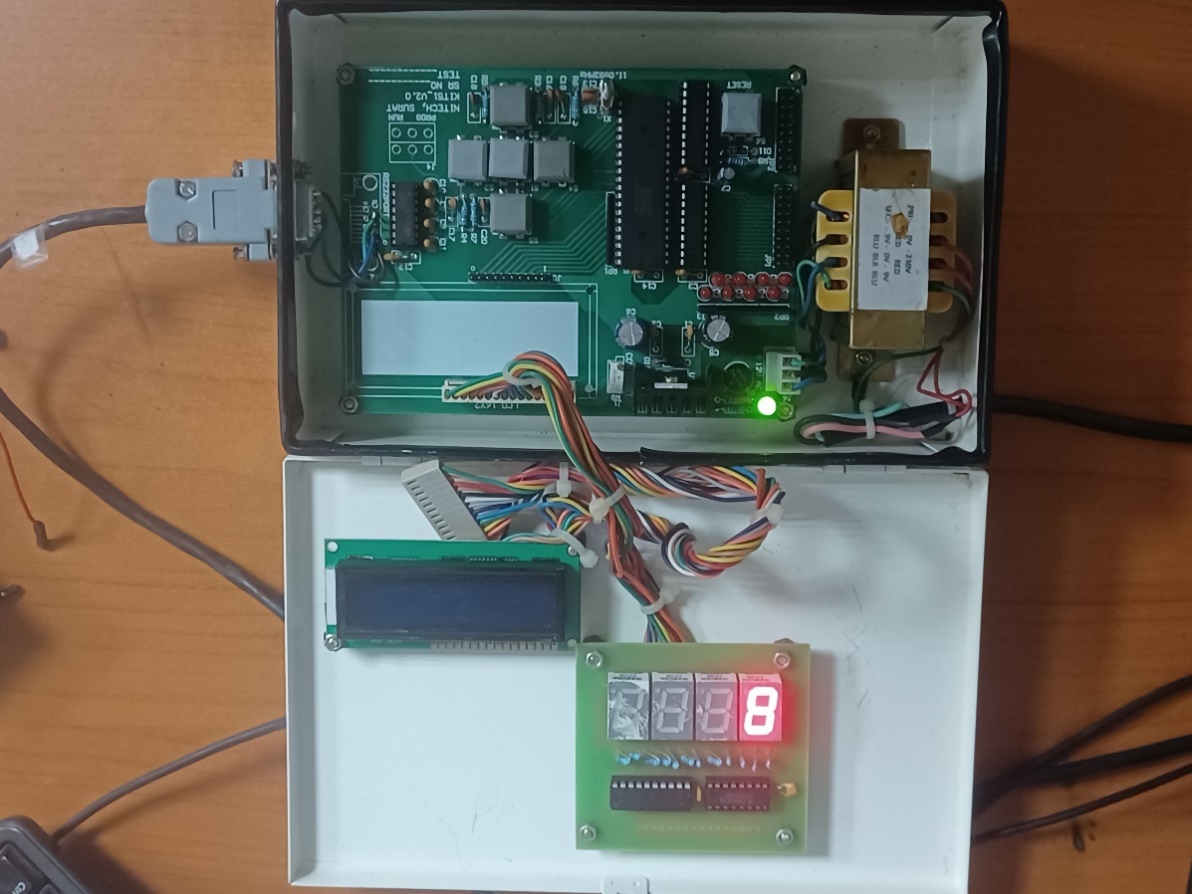
}

TH1=0xf3;

TL1=0xfb;

}

**RESULT**:



**CONCLUSION:**

This embedded C program utilizes timers to cycle through values 0-9 displayed on a 7-segment interface. Interrupts manage timing for display updates. Ensure proper hardware connections and consider memory constraints for expanded functionalities.