**Timers and Interrupts**

**Task 03**

****

Spring 2022

CSE-307 Microprocessor Based system Design

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr: Bilal Habib**

**April** 25, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Task:**

1. Generate a signal on pin P1.1 having frequency equal to 10 Hz with a duty cycle of 10%.
2. When a user presses a button at P1.2 then frequency changes to 20Hz with a 20% duty cycle.
3. When a user again presses the same button then frequency changes to 40Hz with a duty cycle of 40%.
4. When a user again presses the same button then frequency changes to 80Hz with a duty cycle of 80%.
5. Show it on an oscilloscope.
6. Each time a user presses a button the signal toggles from case A to B, then B to C, then C to D and finally from D to A, on every subsequent button press.
7. Program only in C.
8. Use Timers to generate delays.
9. Button press means press and release. Press doesn’t mean you have to keep the button pressed without releasing it.
10. Use only Proteus.

## Problem Analysis:

**Case A:** To generate a signal of frequency 10Hz we need a time period of 1/10 s So T = 1/f = 1/10 = 0.1s

T = 100 ms

As Duty Cycle is 10% so

P1.1 ON (10 ms) P1.1OFF(90 ms)

**Case B:** To generate a signal of frequency 20Hz we need a time period of 1/20 s So T = 1/f = 1/20 = 0.05s

T = 50 ms

As Duty Cycle is 20% so

P1.1 ON (10 ms)

P1.1 OFF (40 ms)

**Case C:** To generate a signal of frequency 40Hz we need a time period of 1/40 s So T = 1/f = 1/40 = 0.025s

T = 25 ms

As Duty Cycle is 40% so

P1.1 ON (10 ms)

P1.1 OFF (15 ms)

**Case D:** To generate a signal of frequency 80Hz we need a time period of 1/80 s So T = 1/f = 1/80 = 0.0125s

T = 12.5 ms

As Duty Cycle is 80% so

P1.1 ON (10 ms)

P1.1 OFF (2.5 ms)

**Source Code:**

#include <stdio.h>

sbit signal=P1^1;

sbit button=P1^2;

int I;

void Delay(int TH0\_reg,int TL0\_reg ) //give 1.085 micro sec delay.

{

TMOD=01;

TH0=TH0\_reg;

TL0=TL0\_reg;

TR0=1;

while(TF0==0);

TR0=0;

TF0=0;

}

void main()

{

int count=0;

button=1;

while(1)

{

if(button==0)

{

count++;

}

if(count%4==0)

{

signal=1;

Delay(0xDB,0xFE); //10msec

signal=0;

for(i=0; i<9; i++)

{

Delay(0xDB,0xFE); //9x10msec = 90 msec

}

}

if(count%4==1)

{

signal=1;

Delay(0xDB,0xFE); //10 msec

signal=0;

Delay(0x6F,0xFD); //40 msec

}

if(count%4==2)

{

signal=1;

Delay(0xDB,0xFE); //10 msec

signal=0;

Delay(0xC9,0xFF); //15 msec

}

if(count%4==3)

{

signal=1;

Delay(0xDB,0xFE); //10 msec

signal=0;

Delay(0xF6,0xFF); //2.5 msec

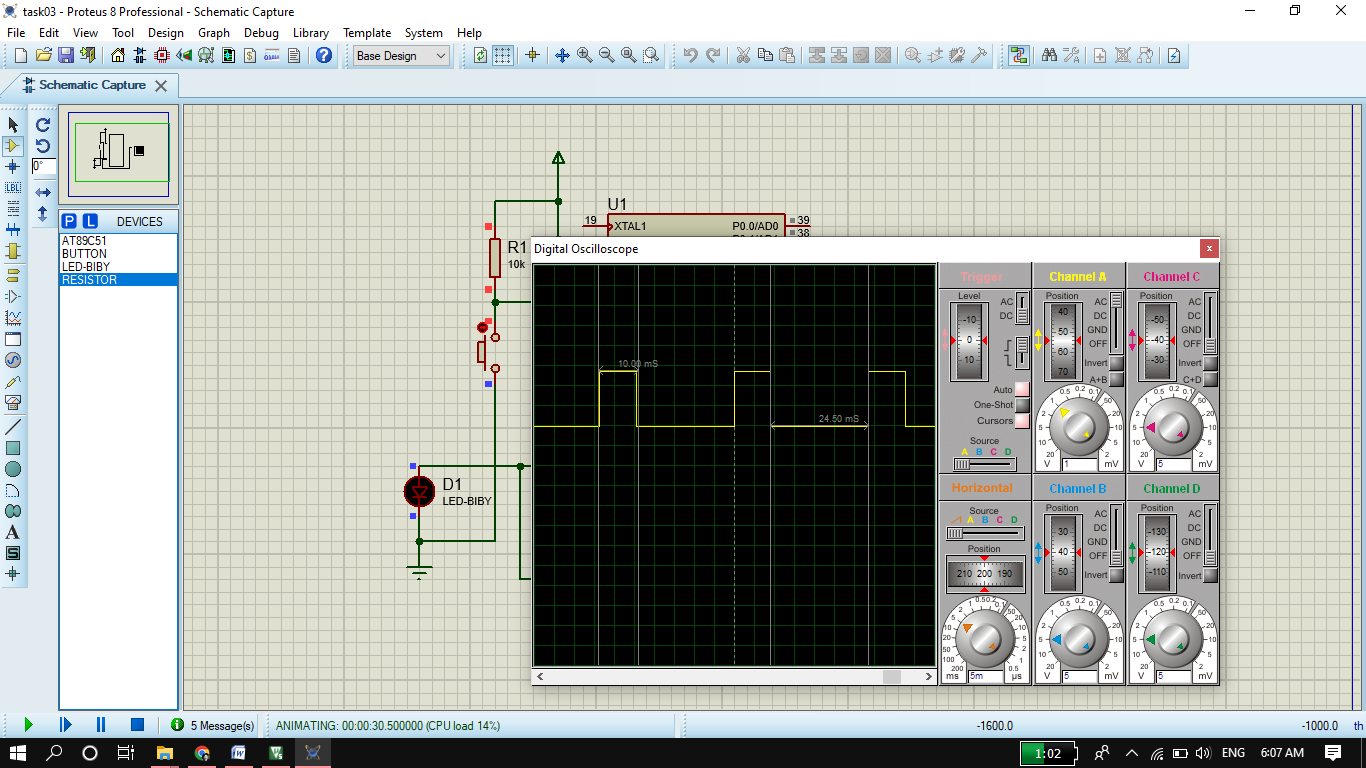
}

}

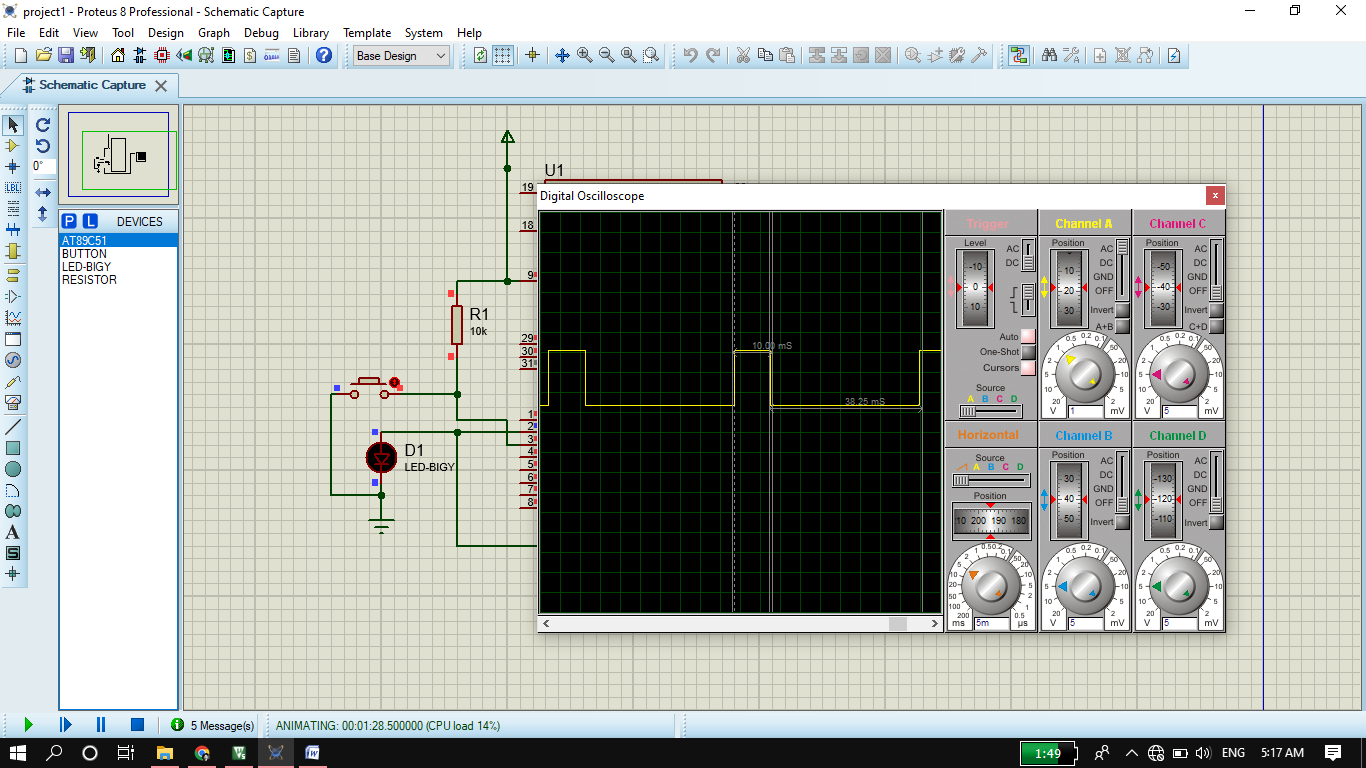
}

**Outputs:**

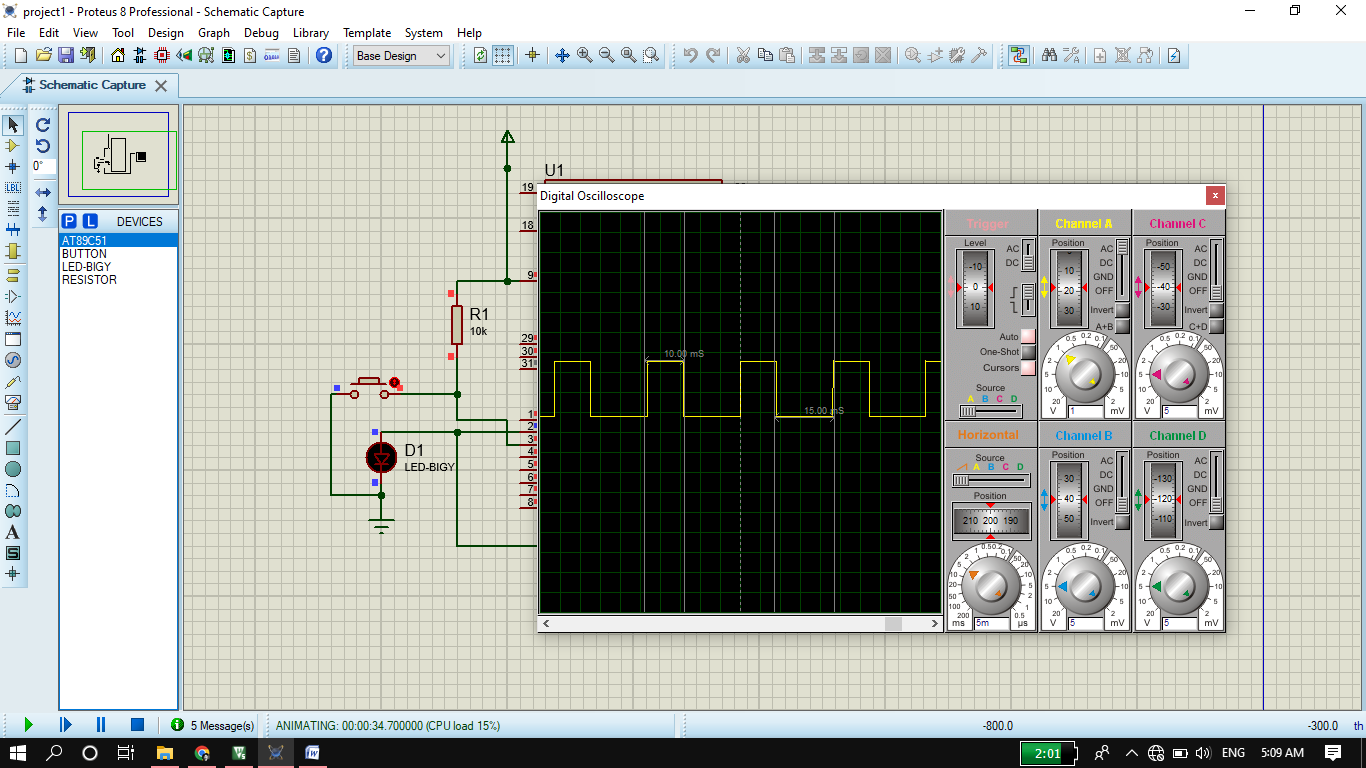
* **A** **signal on pin P1.1 having frequency equal to 10 Hz with a duty cycle of 10**%.



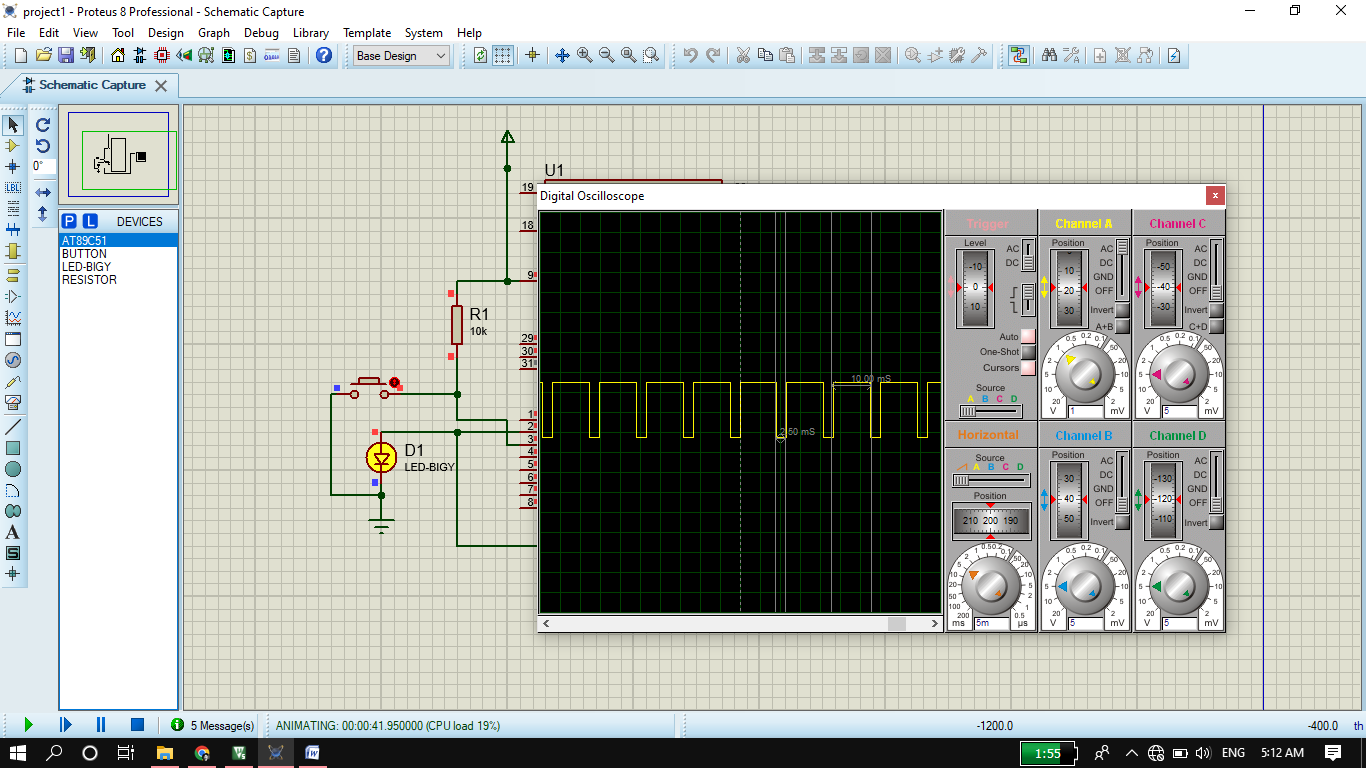
* **After pressing button frequency changes to 20Hz with a 20% duty cycle.**



* **Again after pressing button frequency changes to 40Hz with a 40% duty cycle.**



* **Again after pressing button frequency changes to 80Hz with a 80% duty cycle.**



**Circuit Diagram:**

