Embedded C Programming
Laboratory 6
Programs on Timer/Counter

SOFTWARE TASKS:

TIMER

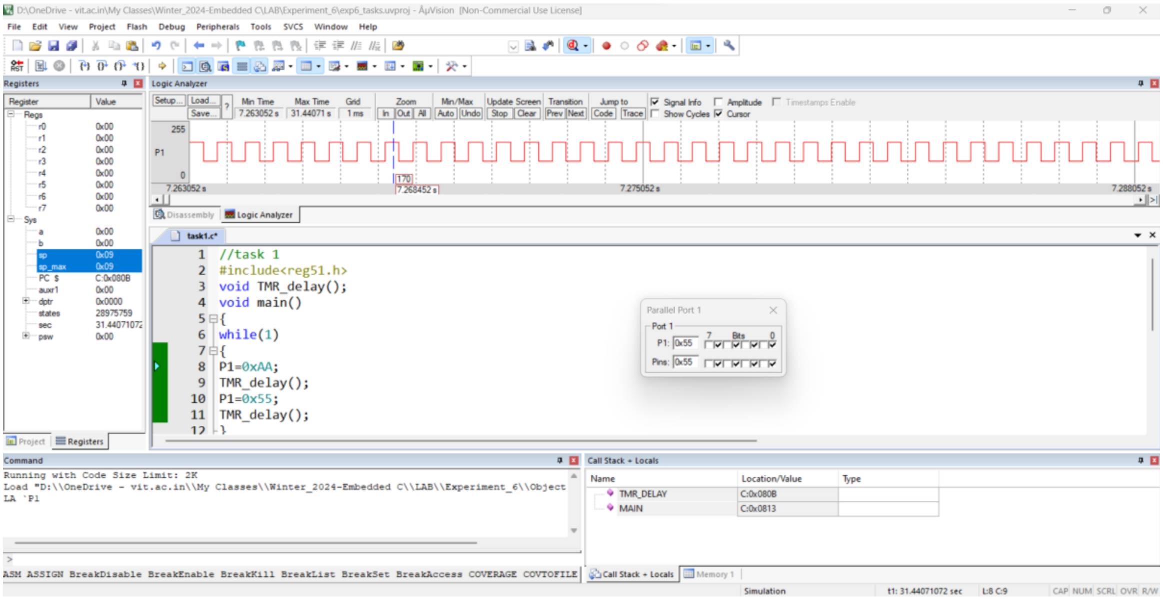
Task 1:

Write a 8051 C program to toggle port P1 continuously with some delay. Use timer 0 16 bit mode for delay.

```
//task 1
#include<reg51.h>
void TMR_delay();
void main()
{
while(1)
{
P1=0xAA;        //Set P1 to 0xAA (alternating 1s and 0s)
TMR_delay();  // Call delay function
P1=0x55;        // Set P1 to 0x55
TMR_delay();  // Call delay function
}
}

void TMR_delay()
{
        TMOD=0x01;  // Set Timer 0 in 16-bit mode
        TH0=0xFE;     // Set Timer 0 high byte to create a specific delay
        TL0=0XBE;     // Set Timer 0 low byte
        TR0=1; // Start Timer 0
        while(TF0==0) // Wait for Timer 0 overflow flag - TF
        TF0=0;          //Clear Timer 0 overflow flag
        TR0=0;          //stop Timer 0
}
```

Output:



D:\OneDrive - vit.ac.in\My Classes\Winter_2024-Embedded C\LAB\Experiment_6\exp6_tasks.uvproj - ÅµVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x09
sp_max	0x09
PC $	C:0x080B
auxr1	0x00
dptr	0x0000
states	28975759
sec	31.44071072
psw	0x00

```
//task 1
#include<reg51.h>
void TMR_delay();
void main()
{
while(1)
{
P1=0xAA;
TMR_delay();
P1=0x55;
TMR_delay();
}
```

P1: 0x55 □☑□☑□☑□☑

Pins: 0x55 □☑□☑□☑□☑

```
Running with Code Size Limit: 2K
Load "D:\\OneDrive - vit.ac.in\\My Classes\\Winter_2024-Embedded C\\LAB\\Experiment_6\\Object
LA `P1
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
TMR_DELAY	C:0x080B	
MAIN	C:0x0813	

Simulation t1: 31.44071072 sec L:8 C:9

Task 2:

Write an 8051 C program to toggle all bits of P2 continuously every 500ms. Use Timer 1, mode 1 to create the delay.

Making TH and TL both zero means that the timer will count from 0000 to FFFF, and then roll over to raise the TF flag. As a result, it goes through a total Of 65536 states. Therefore, we have delay =

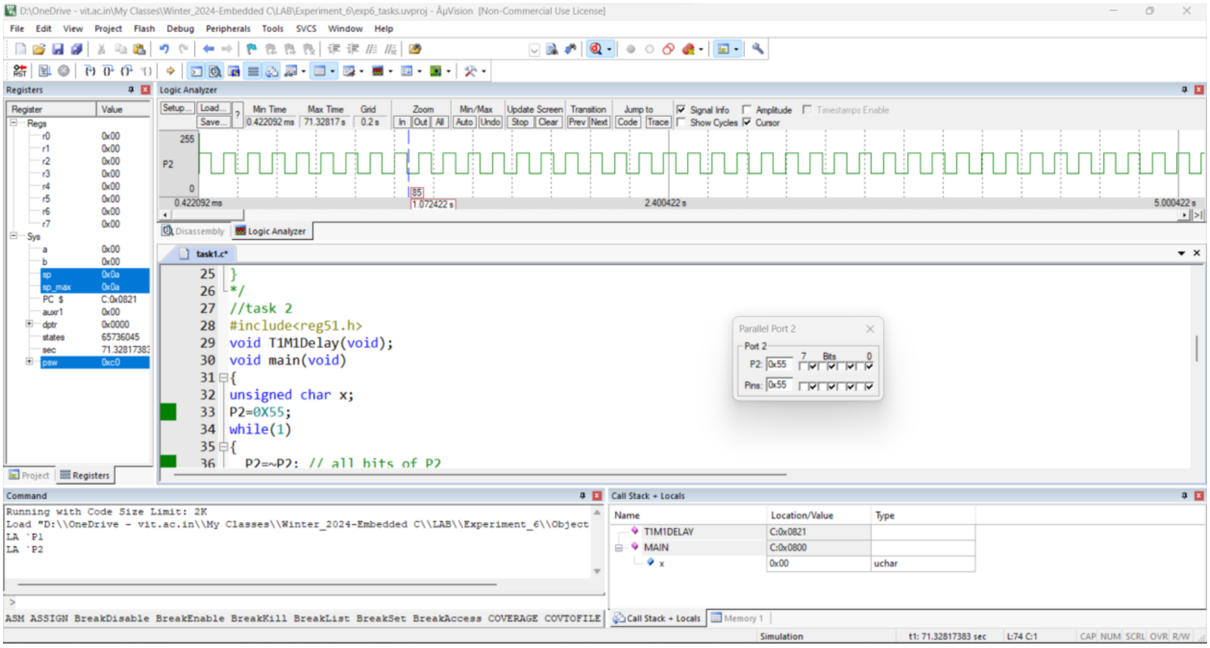
(65536 - 0) × 1.085 us = 71.1065ms.

71.1065ms * X =500ms

X = 500/71.1065 = 7

```
#include<reg51.h>
void T1M1Delay(void);
void main(void)
{
unsigned char x;
P2=0X55;
while(1)
{
  P2=~P2; // all bits of P2
for(x=0;x<7;x++)
            T1M1Delay();
}
}
void T1M1Delay(void)
{
            TMOD=0x10;//Timer 1 Mode 1
            TH1=0x00;//Set Timer 1 high byte to 0
            TL1=0X00;//Set Timer 0 high byte to 0
            TR1=1;//Start timer 1
```

```
        while(TF1==0) //Check overflow flag
        TF1=0;//
        TR1=0; //Stop timer 1
}
```



D:\OneDrive - vit.ac.in\My Classes\Winter_2024-Embedded C\LAB\Experiment_6\exp6_tasks.uvproj - µVision [Non-Commercial Use License]

Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0a
sp_max	0x0a
PC $	C:0x0821
auxr1	0x00
dptr	0x0000
states	65736045
sec	71.32817383
psw	0xc0

```
}
*/
//task 2
#include<reg51.h>
void T1M1Delay(void);
void main(void)
{
unsigned char x;
P2=0X55;
while(1)
{
  P2=~P2; // all bits of P2
```

Command

```
Running with Code Size Limit: 2K
Load "D:\\OneDrive - vit.ac.in\\My Classes\\Winter_2024-Embedded C\\LAB\\Experiment_6\\Object
LA `P1
LA `P2
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Call Stack + Locals

Name	Location/Value	Type
T1M1DELAY	C:0x0821	
MAIN	C:0x0800	
x	0x00	uchar

Task 3:

Write an 8051 C program to toggle all bits of P2 continuously every 500ms. Use Timer 1, mode 1 to create the delay. (consider TH1 as A5, TL1 as FE)

```
#include <reg51.h>
void T1M1Delay(void);
void main(void){
unsigned char x;
P2=0x55;
while (1) {
P2=~P2;
for (x=0;x<20;x++)
T1M1Delay();
}
}
void T1M1Delay(void){
TMOD=0x10;
TL1=0xFE;
TH1=0xA5;
TR1=1;
while (TF1==0);
TR1=0;
TF1=0;
}
```

COUNTER

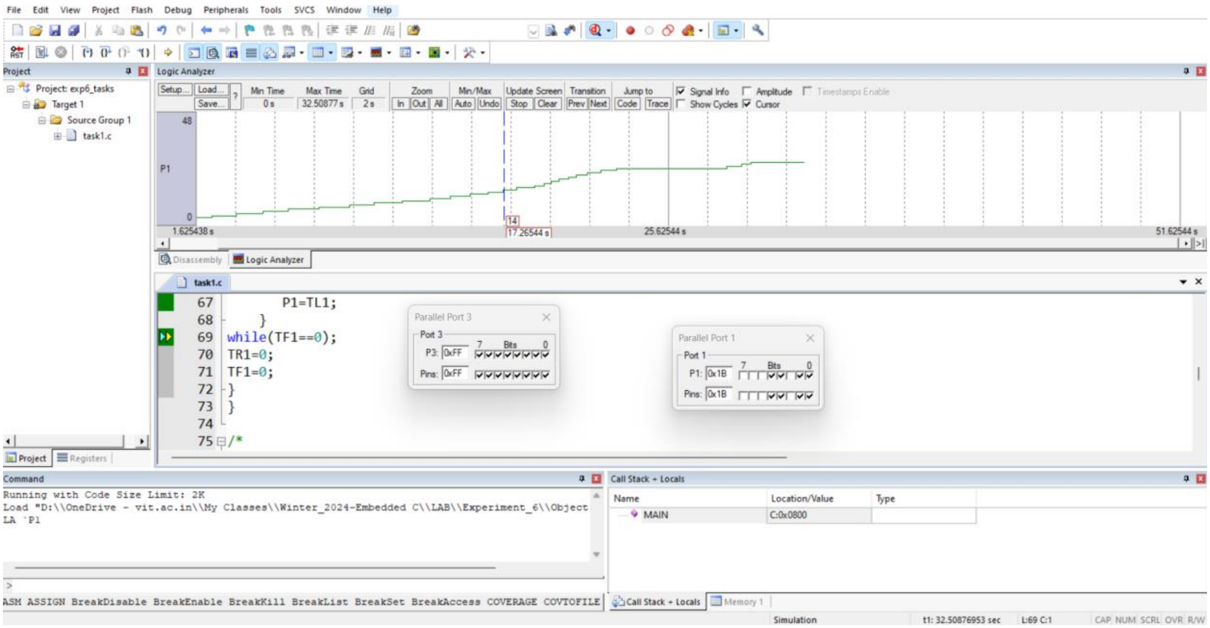
Task 4:

Assume that a 1-Hz external clock is being fed into pin T1(P3.5). Write a C program for counter 1 in mode 2 (8-bit auto reload) to count up and display the state of the TL1 count on P1. Start the count at 0H.

```
#include <reg51.h>
void main(void)
{
   T1=1;
   TMOD = 0X60;
   TH1 = 0;

   while(1)
{
 do
    {
       TR1=1;
       P1=TL1;
    }
while(TF1==0)
TR1=0;
TF1=0;
}
}
```

Output:



Min Time: 0 s
Max Time: 32.50877 s
Grid: 2 s

☑ Signal Info ☐ Amplitude
☐ Show Cycles ☑ Cursor

1.625438 s | 17.26544 s | 25.62544 s | 51.62544 s

```
          P1=TL1;
      }
while(TF1==0);
TR1=0;
TF1=0;
}
}

/*
```

Port 3
P3: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑

Port 1
P1: 0x1B 7 Bits 0 ☐☐☐☑☑☐☑☑
Pins: 0x1B ☐☐☐☑☑☐☑☑

```
Running with Code Size Limit: 2K
Load "D:\\OneDrive - vit.ac.in\\My Classes\\Winter_2024-Embedded C\\LAB\\Experiment_6\\Object
LA `P1
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
MAIN	C:0x0800	

Simulation   t1: 32.50876953 sec   L:69 C:1   CAP NUM SCRL OVR R/W

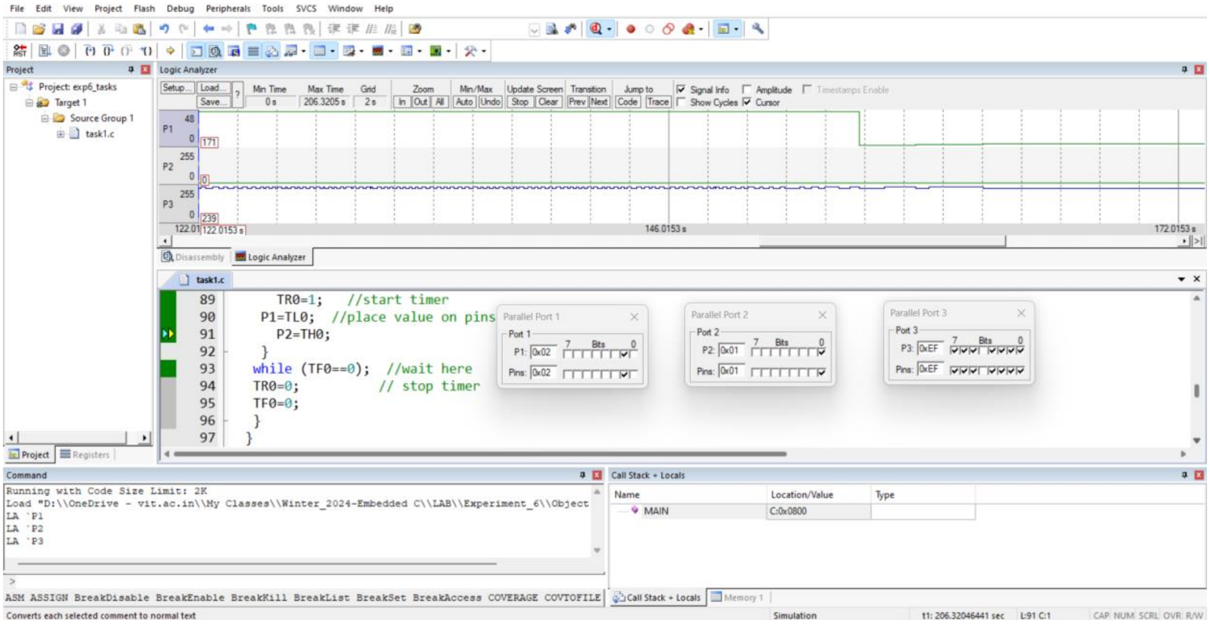
Task 4:

Assume that a 1-Hz external clock is being fed into pin T0(P3.4). Write a C program for counter 0 in mode 1 to count up and display the state of the TL0 and TH0 registers on P2 and P1 respectively. Start the count at 0H.

```
#include <reg51.h>
void main(void)
 {
    T0=1;  //make T0 an input
        TMOD=0x05;
        TL0=0;   //set count to 0
        TH0=0;  //set count to 0

        while(1)   //repeat forever
        {
          do
          {
            TR0=1;   //start timer
                  P1=TL0;  //place value on pins
                  P2=TH0;
          }
         while (TF0==0)  //wait here
         TR0=0;         // stop timer
         TF0=0;
         }
 }
```

Output:



```
        TR0=1;   //start timer
      P1=TL0;  //place value on pins
        P2=TH0;
      }
     while (TF0==0);  //wait here
     TR0=0;          // stop timer
     TF0=0;
     }
    }
```

```
Running with Code Size Limit: 2K
Load "D:\\OneDrive - vit.ac.in\\My Classes\\Winter_2024-Embedded C\\LAB\\Experiment_6\\Object
LA `P1
LA `P2
LA `P3
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
MAIN	C:0x0800	

Converts each selected comment to normal text

Simulation t1: 206.32046441 sec L:91 C:1 CAP NUM SCRL OVR R/W

HARDWARE

TIMER

Task 5:

Write an 8051 C program to toggle port only bit p1.5 continuously every 50ms.

```
#include<reg51.h>
void T0M1Delay(void);
sbit mybit=P1^5;
void main(void)
{
while(1)
{
 mybit=~mybit;
        T0M1Delay();
}
}
void T0M1Delay(void)
{
        TMOD=0x01;
        TH0=0x4B;
        TL0=0XFD;
        TR0=1;
        while(TF0==0);
        TF0=0;
        TR0=0;
}
```