

Microprocessors and Microcontrollers (EE-231)

Lab-8

Objective

- To familiarize ourselves in C programming by designing
 - A counter from 0-99 in C
- To learn to program Timers in C
 - For Specific Delay generation
 - For Counting external Events

Today's Task 1

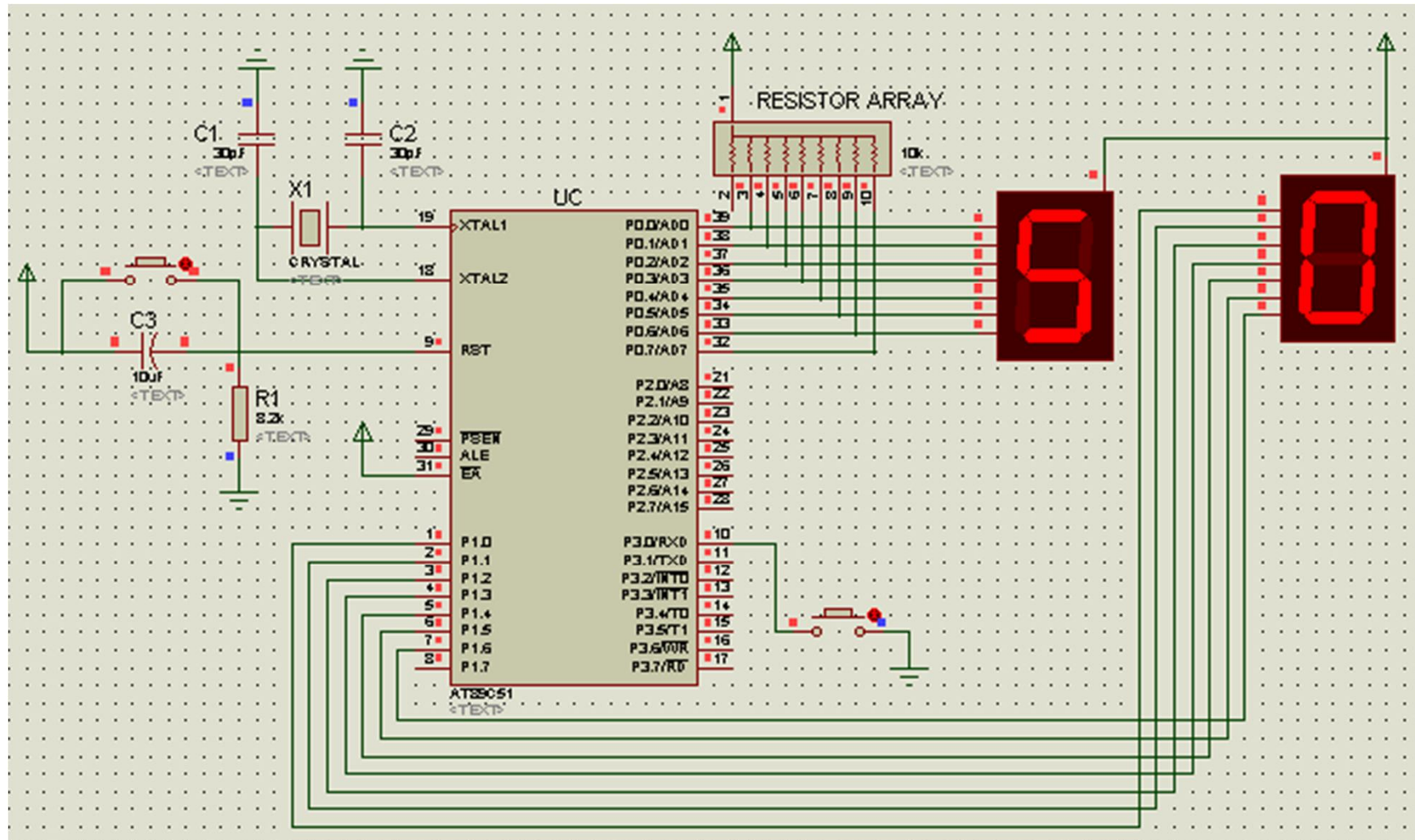
- Design a Counter in C that counts from 0-F. Show the result on Seven Segment. **Simulate the design in Proteus.**
- Modify the counter so that it counts from 0-99 in decimal. **Simulate the design in Proteus.**
- Modify the code in such a way that the counter needs a Push button to be pressed for it to start. And once the counter is started, the button has no role i.e. one shot. **Simulate the design in Proteus.**

Task Code

```
1 #include <reg51.h>
2 void MSDelay(unsigned int);
3 sbit start=P3^0;
4 void main (void)
5 {
6     code unsigned char SSegment[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,
7     0x82,0xF8,0x80,0x90,0xA0,0x83,0xA7,0xA1,0x84,0x8E};
8
9     unsigned char count_high,count_low;
10    start=1;//Make it input
11    while(start==1);
12    while(1)
13    {
14        if(count_low==9)
15        {
16            if(count_high==9)
17            {
18                count_low=0;
19                count_high=0;
20            }
21            else
22                count_low=0;
23            count_high++;
24        }
25        else
26            count_low++;
27
28        P1=SSegment[count_low];
29        P0=SSegment[count_high];
30        MSDelay(500);
31    }
32 }
```

```
33 //Delay function Definition
34 void MSDelay(unsigned int itime)
35 {
36     unsigned int x,z;
37     for(x=0;x<itime;x++)
38         for(z=0;z<114;z++);
39 }
```

Proteus Simulation



Timer Programming in C

```
1  #include <reg51.h>
2  void Timer(void);
3  sbit mybit=P2^0;
4  void main (void)
5  {
6
7      while(1)
8      {
9          mybit=~mybit;
10         Timer();
11     }
12 }
13 void Timer(void)
14 {
15     TMOD=0x10;//Timer 1 mode 1
16     TH1=0xFE;
17     TL1=0x34;
18     TR1=1;
19     while(TF1==0);
20     TR1=0;
21     TF1=0;
22 }
23
```

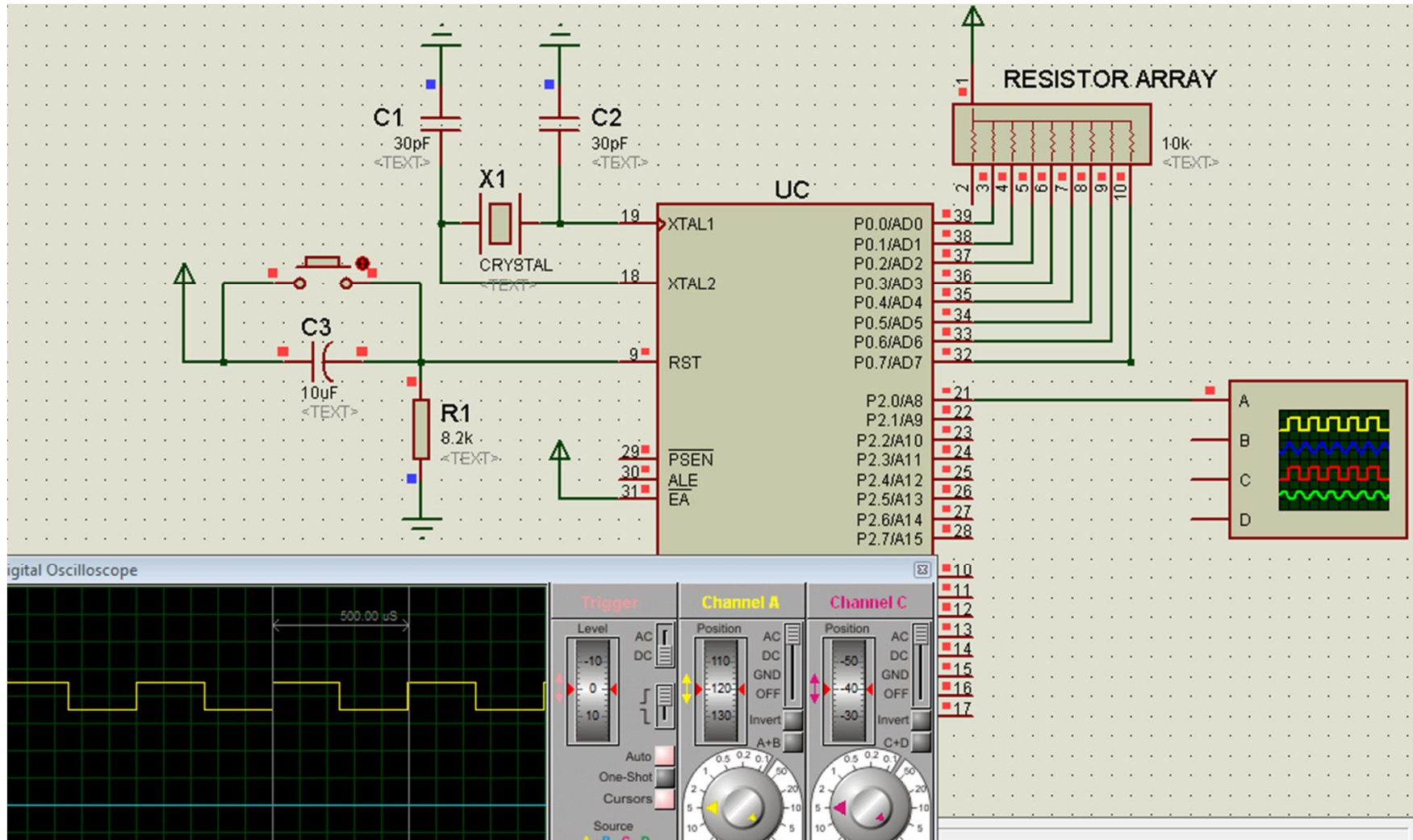
Today's Task 2

- Generate a **square wave** of 2 KHz on pin P2.0
- Use Timer in mode 2
- Verify the design in Proteus ISIS by measuring the frequency using oscilloscope

Task Code

```
1 #include <reg51.h>
2
3 sbit mybit=P2^0;
4 void main (void)
5 {
6     TMOD=0x20;//Timer 1 Mode 2
7     TH1=-230;//Want to count for 230. "OR" I could have also written 256-230 = 26 or 0x1A
8     TR1=1;
9
10    while(1)
11    {
12        mybit=~mybit;
13        while(TF1==0);
14        TF1=0;
15    }
16 }
```


Proteus Simulation



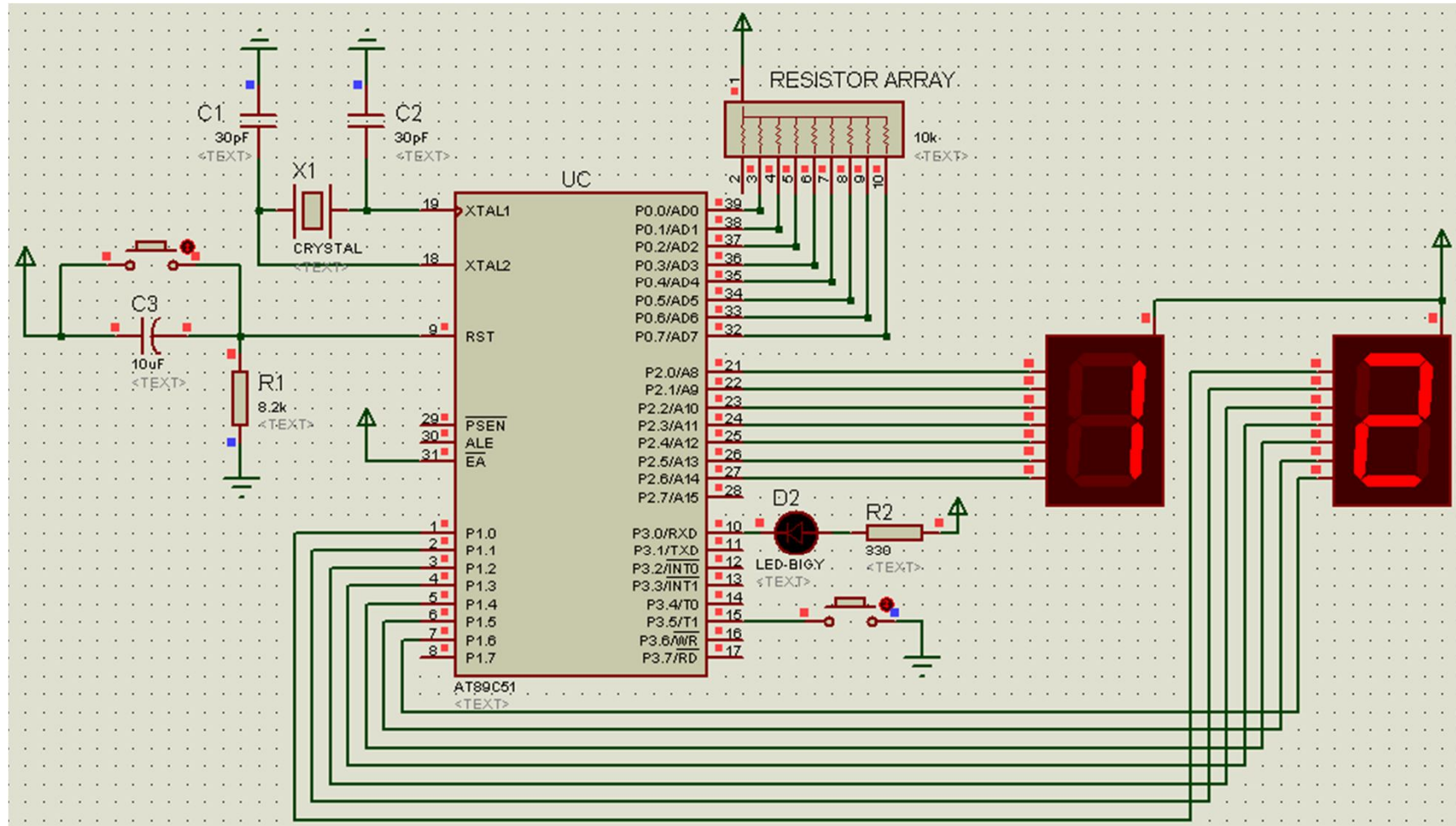
Today's Task 3

- Implement the last week's assignment in C.
- Use Timer 1 as **counter in mode 2** to count the external events from 0-FF and show the status of count (i.e. value of TL1) on **two seven segments**. When the count is completed, i.e. value is overflowed from FF to 00, then turn on an **LED** to indicate the completion of the count.
- Simulate the design on Proteus
- Use Push Button as input of external events (to be connected to T0 pin i.e. P3.4 pin)

Task Code

```
1 #include <reg51.h>
2
3 sbit LED=P3^0;
4 void main (void)
5 {
6     code unsigned char SSegment[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,
7                                     0x82,0xF8,0x80,0x90,0xA0,0x83,
8                                     0xA7,0xA1,0x84,0x8E};
9     TMOD=0x60;//Timer 1 Mode 2
10    TH1=0x00;//
11    TR1=1;
12    while (TF1==0)
13    {
14        ACC=TL1 & 0x0F;
15        P1=SSegment[ACC];
16        ACC=TL1 & 0xF0;
17        ACC=ACC>>4;
18        P2=SSegment[ACC];
19    }
20    LED=0;// Turn on LED
21 }
```

Proteus Simulation



Next weeks Assignment

- Design and Simulate the Following on Proteus.
- Use C language to implement a **square wave generator** with frequency selection capability. Use Pin 2.0 for frequency generation.
- Use Pins P3.0 , P3.1 and P3.2 for Frequency Selection according to following table.
- Use whichever timer mode you like.

P3.2	P3.1	P3.0	Frequency
0	0	0	1Hz
0	0	1	100Hz
0	1	0	250Hz
0	1	1	500Hz
1	0	0	1KHz
1	0	1	2KHz
1	1	0	2.5KHz
1	1	1	4KHz