NANYANG POLYTECHNIC
EGR204 Microcontroller Applications

Laboratory Session 7

Course: Diploma in Robotics and Mechatronics

Module: EGR204 Microcontroller Applications

Experiment: 7

Title: Using the 8051 for counter and timer operations (Electronics Organ)

Objective:

- The students will learn how to use the 8051 Timer to write an electronics organ program in 'C' program for the 8051 microcontroller.

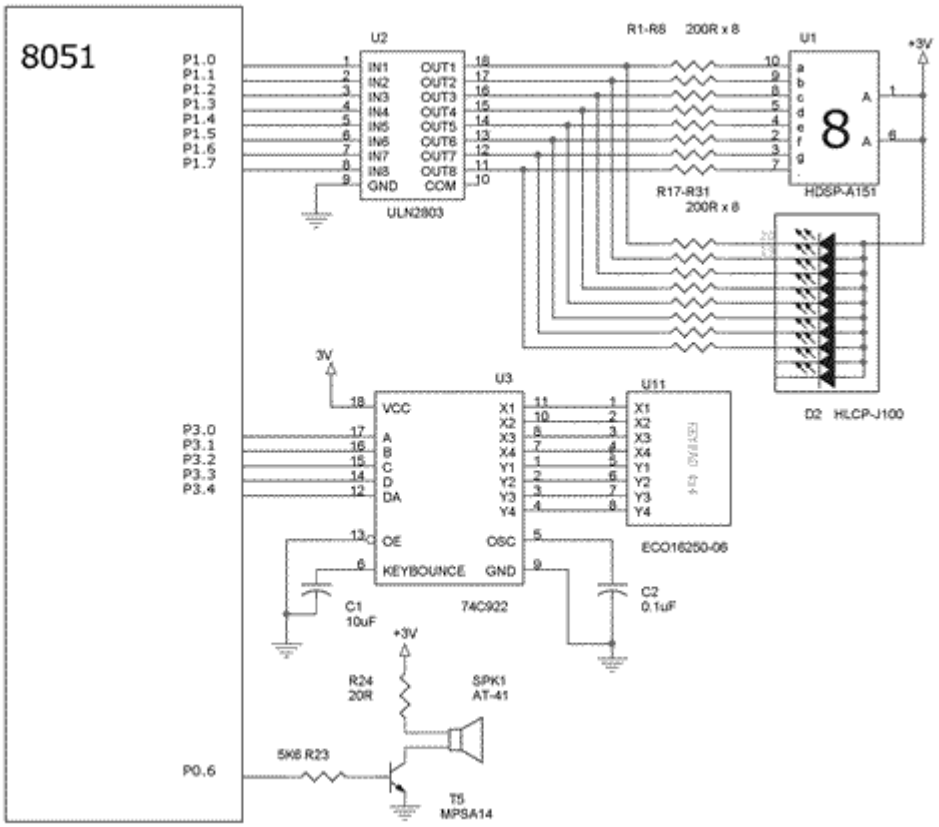
Learning Objectives:

- Understand the concept of generating square waveform of various frequencies with the 8051.
- Learn how to use the 8051 timer to generate various timing.
- Apply knowledge of your 8051 timer to generate square waveform of various frequencies.
- Recall how to use the keyscan routine to read the 4x4 keypad.
- Write a simple electronics organ program.

1. Introduction

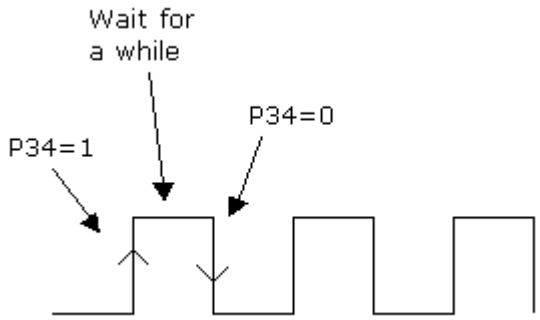
Figure 1-1 shows how the 8051 is interfaced to a 4 x 4 keypad and a speaker.

Figure 1-1
Keypad and Speaker Circuit



2.1 Exercise 1: Output a Square Waveform at the I/O Port.

A square waveform can be generated at Port P06, the I/O port that is connected to the speaker, by toggling the level of P06 (see illustration below). This can be achieved with the program in listing 2-1. Run the program. What is the frequency of the sound generated by the program? How do you generate a 1 KHz tone?



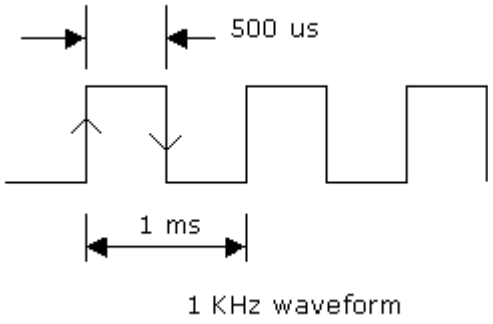
Listing 2-1

```
#include <f200.h>

void delay(unsigned long duration)
{
   while((duration--)!=0);
}

void setSystem();

void main()
{
   setSystem();
   for(;;)
   {
      P06 = ~P06;
      delay(100);
   }
}
```

How do you generate a 1 KHz waveform?

A 1 KHz has a period of 1 ms. Therefore, the delay required in the listing 2-1 program should be 500 us to achieve a 1 KHz waveform.

How do you accurately achieve a 500 us delay timing? Use the 8051 timer, of course.



1 KHz waveform

2.2 Exercise 2: Output a 1 KHz Square Waveform at the I/O Port.

Run the program in listing 2-2. You will hear a 1 KHz tone generated by the speaker connected to P06. The 500 us delay is generated by the timer statement:

```
TH1 = 0xFE;
TL1 = 0x33;
while (TF1==0);
```

Listing 2-2

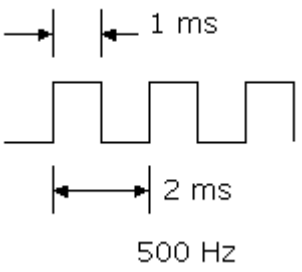
```
#include <f200.h>

void delay(unsigned long duration)
{
   while((duration--)!=0);
}

void setSystem();

void main()
{
   setSystem();
```

```
    TMOD=0x10;
    TF1=0;
    TR1=1;

    for(;;)
    {
       TH1 = 0xFE;
       TL1 = 0x33;
       while(TF1==0);
       TF1 = 0;
       P06 = ~P06;
    }
}
```

3.1 Assignment 1: Generate 500 Hz Tone

The program in listing 2-2 generates a 1 KHz tone. Change the program so that it will generate a 500 Hz tone. Figure 3-1 shows how a 500 Hz waveform looks like. Therefore, you are required to find out the timer load value to generate a delay of 1 ms.

Figure 3-1



3.2 Assignment 2: Designing An Electronics Organ

Now you have learned how to generate tones of different frequencies, you can design an electronics organ with your knowledge. Use the 4 x 4 keypad as your organ keys. Table 3-1 shows the frequencies of the musical tone C5 to C6. Complete the table.

Here is an example of how to calculate the timer value for your 1st note, C5, to help you to start off:

Frequency = 523 Hz
Period = 1/523 = 0.00191 s
Delay Required = 0.00191/2 = 0.000955 s

Timer Count = 0.000955 (s)/ 1.085 (us) = 880

Timer Load Value = 65536 - 880 = 64656
= **0xFC90**

Table 3-1
Musical Tones And Their Frequencies

Notes	Frequencies (Hz)	TH1-TL1
C5	**523**	**0xFC90**
D5	**587**	
E5	**659**	
F5	**698**	
G5	**783**	
A5	**880**	
B5	**987**	
C6	**1046**	

After you have completed table 3-1, start to write your electronics organ program with the help of listing 3-1. Remember to include the keyscan routine in your program.

Listing 3-1
Electronics Organ Program

```
void main()
{
   setSystem();
   TMOD = 0x10;
   TR1=0;
   TF1=0;

   for(;;)
   {
      while (keyscan(&key)==1)
      {
         TR1=1;
         switch(key)
         {
            case 1: // complete your program here….
            case 2:
            default: TH1=0;TL1=0;
         }

         P06 = ~P06;
         while (TF1==0);         // Wait
         TF1 = 0;//Reset
      }
   }
}
```

3.3 Assignment 3: Organ With Fancy Lights

Improve the electronics organ program you have done in your assignment 2. There are eight musical keys in your electronics organ. For each key pressed, make a LED light up. Use the LEDs that is connected to your PORT 1.

4. Program Analysis

Consider writing a program where the music plays by itself? Analyse the program in listing 4-1. How do you compose music with this program? Can you design a musical door bell with the 8051?

Listing 4-1

```
void delay(unsigned int duration)
{
   while(duration--!=0);
}

void music_table(unsigned char notes, unsigned int length)
{
   unsigned int x;

   TR1=1;
   for(x=0;x<length;x++)
   {
      switch(notes)
      {
         case 1: TH1= 0xfc; TL1=0x0c; break;
         case 2: TH1= 0xfc; TL1=0x0c; break;
         case 3: TH1= 0xfc; TL1=0x0c; break;
      }
      while (TF1==0);
      TF1=0;
      P06 = ~P06;
   }
   TR1=0;
}

void main()
{
   setSystem();

   for(;;)
   {
      music_table(1,1000);
      delay(10000);
      music_table(2,1000);
      delay(10000);
      music_table(3,3000);
      delay(20000);
   }
}
```