

# ECE 376 - Homework #9

Timer 0/1/2/3 Interrupts. Due Monday, April 4th

1) Write a C routine using Timer0 interrupts to measure time to 100ns. Using this routine, determine how long a the following operations in C take:

a) LCD display routine

```
long int A;  
A = 123456789;  
LCD_Out(A, 10, 7);    // time to execute this instruction
```

b) The time it takes you to press all buttons on PORTB sequentially

```
TRISB = 0xFF;  
while(!RB0);           // start  
while(!RB1);  
while(!RB2);  
while(!RB3);  
while(!RB4);  
while(!RB5);  
while(!RB6);  
while(!RB7);           // end
```

c) The time it takes you to press and release RB0 10 times

```
TRISB = 0xFF;  
for(i=0; i<10; i++) {   // start  
    while(!RB0);  
    while(RB0);  
}  
                        // end
```

2) Write a C routine using Timer0 / Timer1 / Timer2 / Timer3 interrupts to play 4 notes at the same time when you press button RB0 (4-string Violin)

Output Pin	RC0	RC1	RC2	RC3
Note	A1#	C2#	D2#	F2#
Frequency (Hz)	58.27 Hz	69.30 Hz	77.78 Hz	92.50 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3

Problem 3-7) Write a C program which uses at least two interrupts Timer0/1/3 interrupts. Some suggestions are

- Random number generator: Generate a 6-sided die (RB0), 8 sided die (RB1), or 12 sided die (RB2) based upon the time you press a button. Use statistics to see if the 6-sided die is really random.
- Hungry-Hungry Hippo: Press RB1 to start the game. Two players (RB0 and RB2) then press their buttons as fast as they can for 10 seconds (timed with Timer0). After 10 seconds, the number of button presses by each player is displayed.
- Reflex Timer: Press RB1 to start. Between 3 and 10 seconds later, all of the lights on PORTA turn on. When they turn on, press RB0. The time delay from the lights turning on and pressing RB0 is displayed to 100ns.
- Music Box. Play a tune with your PIC processor
  - Timer0: Set the duration of each note
  - Timer1: Set the frequency of each note
- Starter Tree for drag racing
  - When RB0 is pressed, turn on the lights on PORTC sequentially, one LED every 500ms
  - When all lights on PORTC are on, play 92.50Hz for 1.000 second
- Other

3) Requirements: Explain what the inputs are / what the outputs are / and how they relate. Also explain how each timer interrupt is used in your embedded system.

4) C-Code and flow chart.

5) Test: Collect data in lab to verify that your interrupts are working properly.

6) Validation: Collect data in lab to verify you met your requirements

7) Demo (in person during Zoom office hours or in a video)