

ECE 376 - Homework #9

Timer 0/1/2/3 and INT Interrupts

Timer0 Interrupts

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) Double precision floating point sine

```
#include <math.h>
double A;

A = 1.23456;
A = sin(A);           // time to execute this instuction
```

b) LCD display routine

```
long int A;
A = 3141592653;
LCD_Out(A, 10, 7);    // time to execute this instruction
```

c) The time it takes you to press RB0 ten times (without using INT interrupts)

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

Timer 0 / Timer1 / Tlmer2 / Tlmer3

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

Output Pin	RC0	RC1	RC2	RC3
Note	F3	G3	A3	B3
Frequency (Hz)	174.61 Hz	196.00 Hz	220.00 Hz	246.94 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3

Timer & INT Interrupts

Problem 4-8) Write a C program which uses at least one INT interrupt and one Timer interrupt. Some suggestions are

- Hungry-Hungry Hippo: Press RB7 to start the game. Once the game starts, the number of button presses on RB0 / RB1 / RB2 is recorded using INT interrupts. The length of the game is fixed at 10.00 seconds using Timer interrupts. Your score and time remaining are displayed on the LCD.
- Jump Rope: Press RB7 to start the game. One light on PORTC then turns on and shifts left every Xms, with time kept using a Timer interrupt. When you reach the end (RC7), the light wraps around and continues on RC0. When the light reaches RC3, you are to jump (press RB0 to trigger a INT0 interrupt). The time difference between when you press RB0 and when light RC3 turns on is your score. Display your score on the LCD when you go back to RC0.
- Flash Light: Turn on all of the lights on PORTC from 0% to 100% using PWM. Use timer interrupts to control the on/off times for the lights. Use INT interrupts to adjust the on/off times (each press of RB1 makes the light 1% brighter. Each press of RB0 makes the light 1% dimmer). Display the duty cycle on the LCD display.
- Other

4) 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.

5) C-Code and flow chart.

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

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

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