ECE 376 - Homework #8

Timer 2 Interrupts.   Due Monday, March 28th

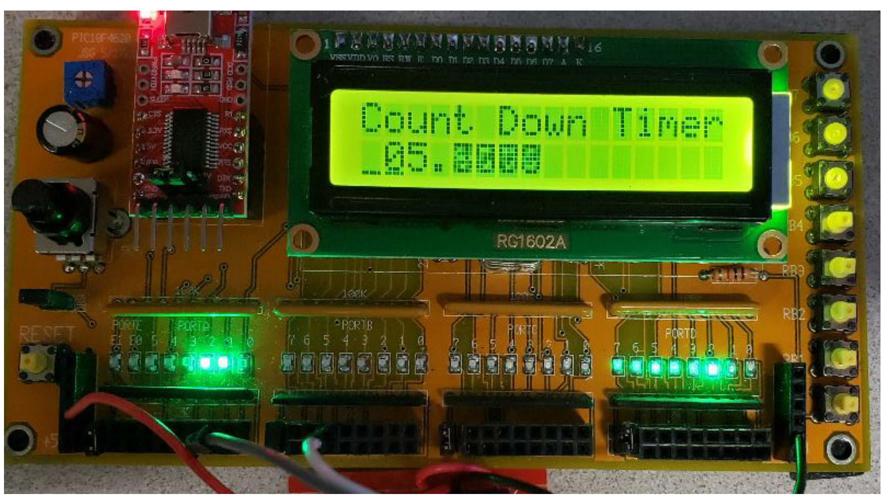
Measuring Time to 0.1ms with Timer2 Interrupts

1) Write a routine for a count-down timer with a resolution of 0.1ms (repeat homework #4 but now with interrupts)

- Time is measured to 0.1ms using Timer2 interrupts
- Each interrupt, pin RC0 is toggled (outputting a 5kHz square wave on RC0)
- Each interrupt (every 0.1ms), TIME is decremented to zero, stopping at zero
- TIME is displayed on the LCD display to 1ms: xx.xxxx
- When you press RB0, the time is reset to 5.0000 seconds
- When you press RB1, the time is reset to 10.0000 seconds
- When you press RB2, the time is reset to 15.0000 seconds
- When you press RB3, the time is reset to 20.0000 seconds

Check the accuracy of your stopwatch

- Measure the frequency on RC0 when sent to a speaker using a cell phone app (Frequency Counter works)



Code:

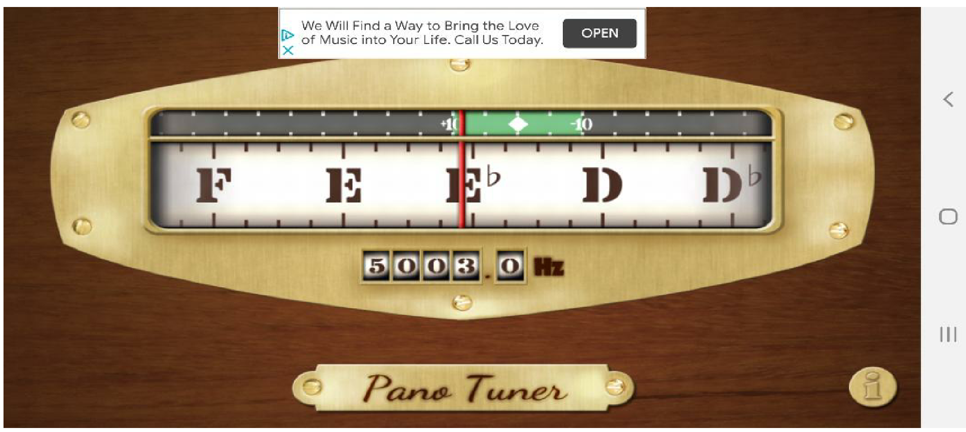
< insert code and flow chart >

Compilation Results:

```
Memory Summary:
    Program space        used   A00h (  2602) of 10000h bytes   (  4.0%)
    Data space           used    33h (    51) of   F80h bytes   (  1.3%)
    EEPROM space         used     0h (     0) of   400h bytes   (  0.0%)
    ID Location space    used     0h (     0) of     8h nibbles (  0.0%)
    Configuration bits   used     0h (     0) of     7h words    (  0.0%)
```

Frequency on RA1: 5003.0 Hz

- verifies that timer2 is running every 100us
- (99.940us measured)



Generating Frequencies with Timer2 Interrupts

2) Write a routine which turns plays your PIC into a 1-string banjo using Timer2 interrupts

- Play note D3# (155.56Hz) on pin RC0 when button RB0 is pressed
- Check the accuracy of your music note using your cell phone (or whatever else you have on hand)

$$N = ABC = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 32,141.939$$

Let

- A = 8
- B = 251
- C = 16

$$N = ABC = 32,128 \text{ (0.043\% low)}$$

To do this

T2CON

7	6	5	4	3	2	1	0
0	0	1	1	1	1	1	1
	A = 8				T2E	C = 16	

- T2CON = 0x3F
- PR2 = 250

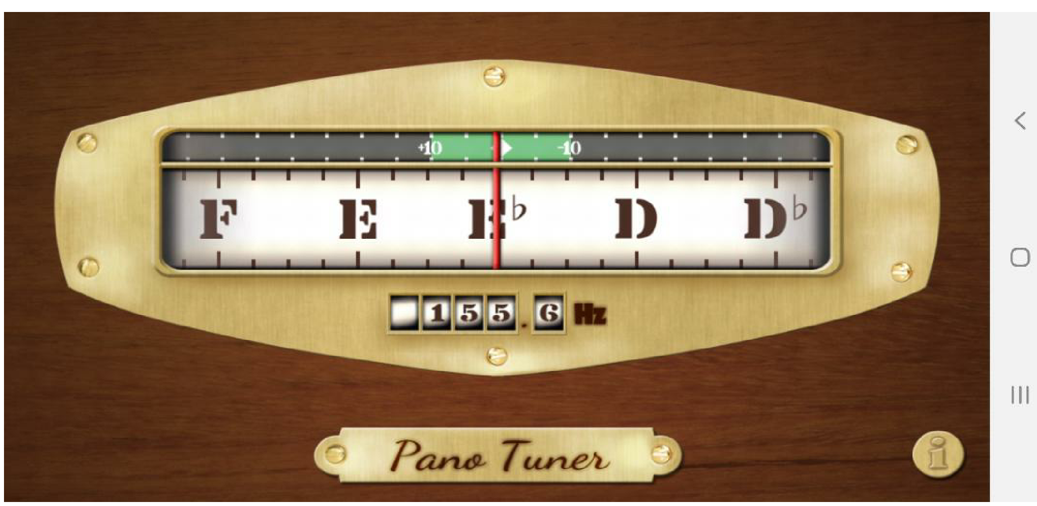
Result:

- f = 155.6Hz

Code:

```
// Problem #2
// 123.47Hz

// Global Variables

const unsigned char MSG0[21] = "155.56Hz            ";
const unsigned char MSG1[21] = "                    ";

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"

// High-priority service
void interrupt IntServe(void)
{
   if (TMR2IF) {
      if(RB0) RC1 = !RC1;
      TMR2IF = 0;
      }
```

```
    }

// Main Routine

void main(void)
{
   unsigned char i;

   TRISA = 0;
   TRISB = 0xFF;
   TRISC = 0;
   TRISD = 0;
   TRISE = 0;
   ADCON1 = 0x0F;


   LCD_Init();                      // initialize the LCD

   LCD_Move(0,0);   for (i=0; i<20; i++) LCD_Write(MSG0[i]);
   LCD_Move(1,0);   for (i=0; i<20; i++) LCD_Write(MSG1[i]);

// set up Timer2 155.56Hz
   T2CON = 0x3F;
   PR2 = 250;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;


// turn on all interrupts
   GIE = 1;
   i = 0;

   while(1) {
      i = i + 1;
      LCD_Move(1,0);   LCD_Out(i, 3, 0);
      Wait_ms(250);
      }
   }
```



Steppper Motor Roulette Wheel

3) Requirements: Explain what the inputs are / what the outputs are / and how they relate. Also explain how Timer2 interrupts will be used in your embedded system.

Input:

- RB0

Output:

- Stepper Motor (on PORTA)
- LCD Display (on PORTD)

Relationship:

- To start the game, press and release RB0.
- This generates a random number from 0..7
- The stepper motor then turns 3 rotations plus 25*N steps at a rate of 10ms/step (set by Timer2)
- The number (0..7) is also displayed on the LCD display as the stepper motor turns

Calculations:

10ms/step is too large for Timer2 directly. So, a counter is added so that the stepper motor turns every 10th interrupt

- Timer2: 1ms
  - A = 10
  - B = 250
  - C = 4
  - Toggle RD0 every interrupt (results in 500Hz square wave on RD0)
- 10th interrupt = 10ms
  - Step the motor every 10ms

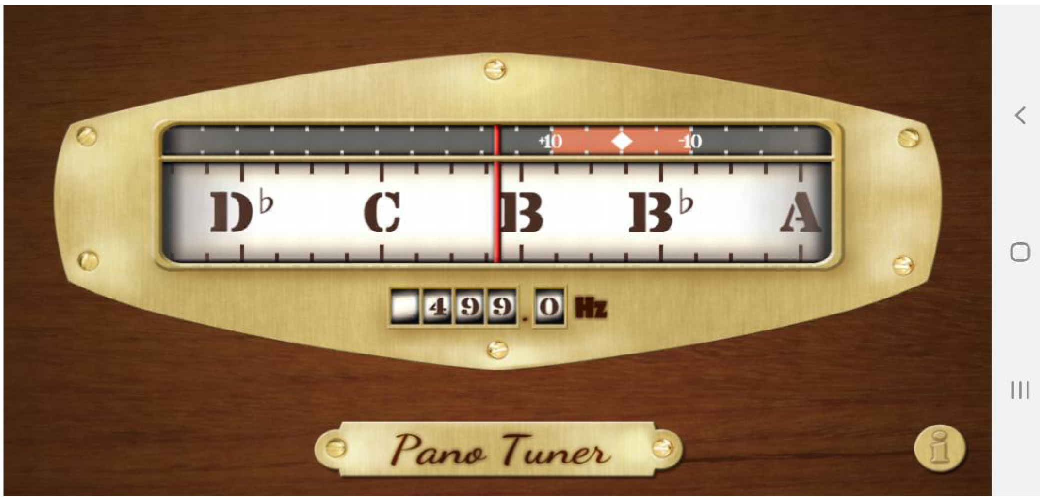
4) C-Code and flow chart.

```
< insert code >
```

5) Data. Your raw data (at least two data points)

Timer2 Interrupt

- 499.0Hz



Winning Numbers

- 1, 6, 3, 0, 1, 5, 0, 3, 7, 5, 6, 0, 5

6) Statistical Analysis: Analyze your data to determine

- The 90% confidence interval, or
- Who in your group can jump the highest (with what probability level), or
- Something else (your pick - just use some statistics to anlayze your data)

With only 14 numbers, there isn't enough data to do a chi-squared test with 8 bins, so use two bins

bin	p	np	N	chi-squared
even	0.5	7	8	1/7
odd	0.5	7	6	1/7
		Total		2/7

From StatTrek, with 1 degree of freedom, this corresponds to a probability of 0.41

I am 41% certain this is not a fair die

Using a different grouping:

bin	p	np	N	chi-squared
0..3	0.5	7	7	0/7
4..7	0.5	7	7	0/7
		Total		0/7

I am 0% certain this is not a fair die

Using yet another grouping:

bin	p	np	N	chi-squared
0 or 5	2/8	3.5	6	1.786
other	6/8	10.5	11	0.595
		Total		2.381

From a chi-squared table with 1 degree of freedom, this corresponds to a probability of 0.88

I am 88% certain this is not a fair die

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