ECE 376 - Homework #8

Timer 2 Interrupts.   Due Monday, October 30th
Please email to jacob.glower@ndsu.edu, or submit as a hard copy, or submit on BlackBoard

Count-Down Timer with 100us Accuracy

1) Write a routine for a count-down timer with a resolution of 100us

- Time is measured to 100us using Timer2 interrupts
- Each interrupt, pin RC0 is toggled (outputting a 5kHz square wave on RC0)
- Each interrupt (every 100us), TIME is decremented to zero, stopping at zero
- TIME is displayed on the LCD display to 0.0001s: 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)

Set Up Timer2 for 0.1ms

N = 1,000

A = 10, C = 4, B = 25

T2CON = 0x4D							
7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	1
	A = 10					C = 4	

Timer2 Initialization:

```
// set up Timer2 for 0.1ms
   T2CON = 0x4D;
   PR2 = 24;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;
```

Set up counters (need to be long integers)

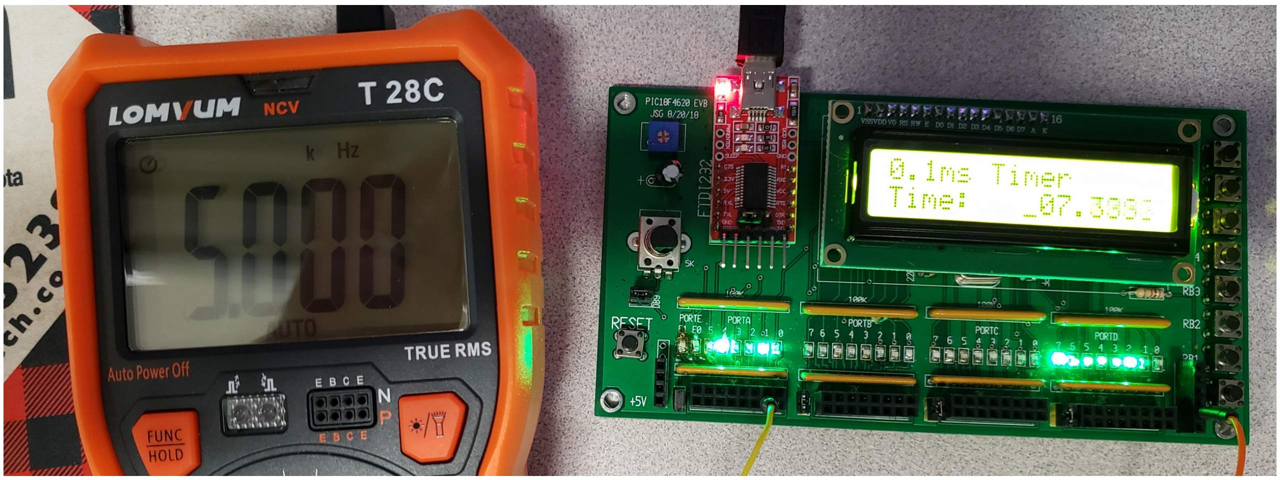
- 5s   N = 50,000
- 10s   N = 100,000
- 15s   N = 150,000
- 20s   N = 200,000

Code Size:

```
Memory Summary:
    Program space        used   A3Eh (  2622) of 10000h bytes   (  4.0%)
    Data space           used    35h (    53) 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%)
```

Resulting Program

- RA1 outputs 5.000kHz
- Timer2 is running at 10kHz (0.1ms)



Generating Frequencies with Timer2 Interrupts

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

- Play note frequency of music note F3 (174.61Hz) 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)

Calculating the number of clocks between interrupts:

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 28{,}635.244$$

One combination of A,B,C that gets you close is

- A = 10
- C = 16
- B = 179 (178.97)

resulting in T2CON being 0x4F

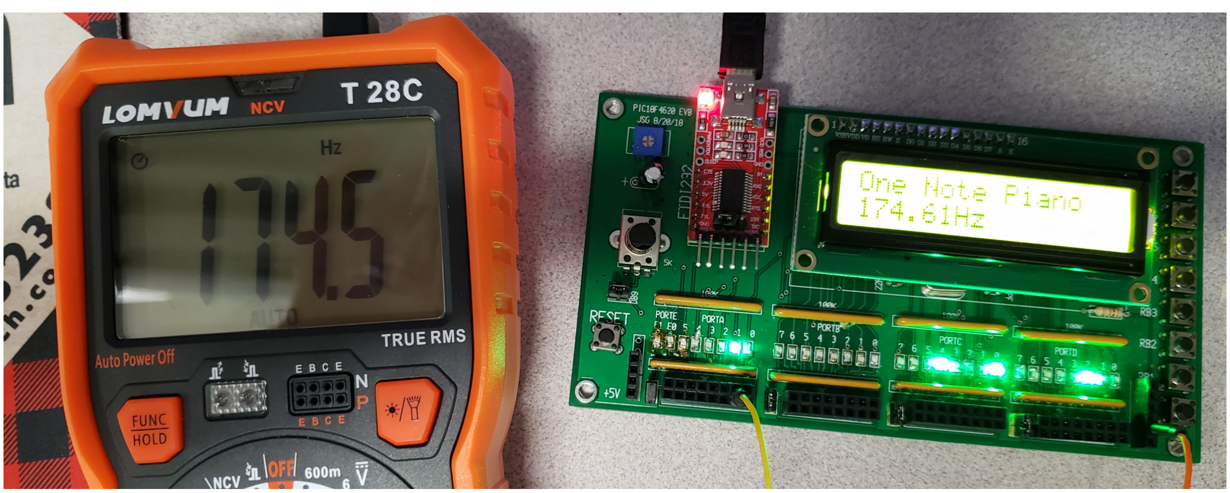
T2CON = 0x4F							
7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	1
	A = 10					C = 16	

Results:

- Frequency = 174.5 Hz
- RA2 toggles when RB0 is pressed (one-key piano)
- Perfect accompanyment for "Burning Down the House" by Talking Heads

Code Size

```
Memory Summary:
    Program space        used   4B4h (  1204) of 10000h bytes   (  1.8%)
    Data space           used    1Eh (    30) of   F80h bytes   (  0.8%)
    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%)
```



Code:

```
// Piano.C
// Set up a Timer2 interrupt for 174.61Hz

// Global Variables

const unsigned char MSG0[21] = "One Note Piano      ";
const unsigned char MSG1[21] = "174.61Hz            ";

unsigned long int TIME;

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"

// High-priority service
void interrupt IntServe(void)
{
   if (TMR2IF) {
      RA1 = !RA1;
      if(RB0) RA2 = !RA2;
      TMR2IF = 0;
      }
   }

// Main Routine

void main(void)
{
   unsigned char i;
   unsigned int j;

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

   ADCON1 = 0x0F;
   PORTA = 0;

   TIME = 0;

   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]);

   Wait_ms(100);


// set up Timer2 for 174.61Hz
   T2CON = 0x4F;
   PR2 = 178;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;

// turn on all interrupts
GIE = 1;

   while(1) {
      PORTC += 1;
      Wait_ms(1000);
      }
   }
```

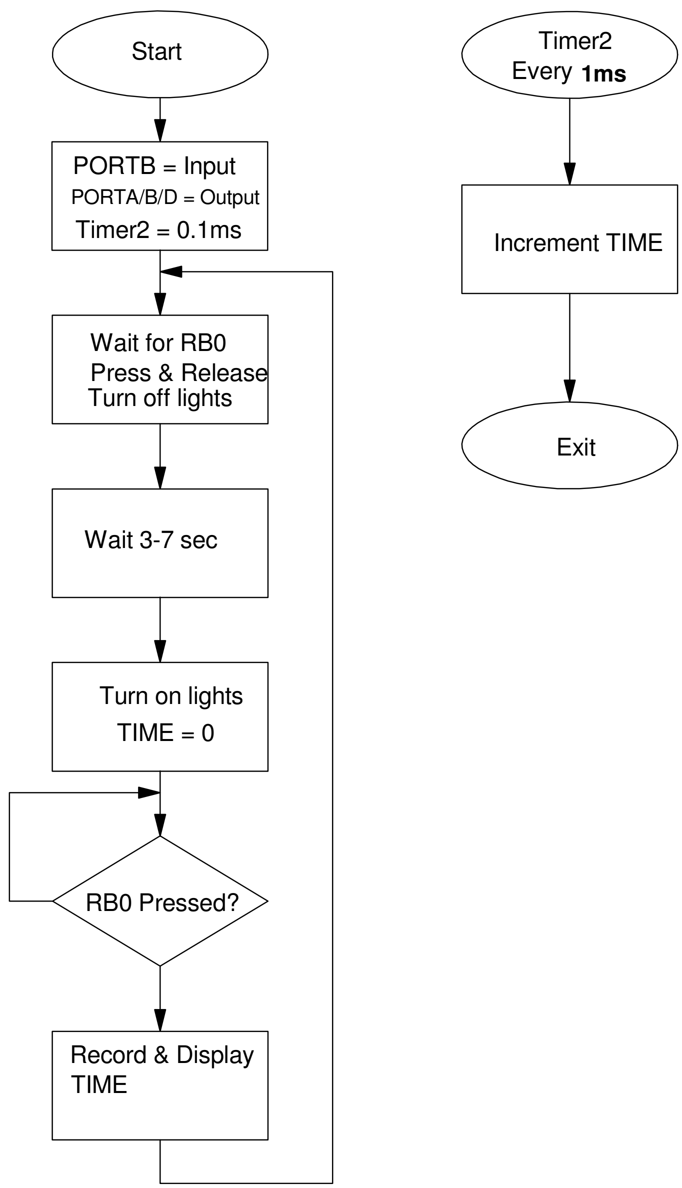
Reflex Timer

Problem 3-7) Build an embedded system which measures your reflex time:

- Start a given trial by pressing and releasing RB0
- Once pressed, the PIC waits between 3.00 and 7.00 seconds (random)
- After that time, all of the lights on PORTA turn on.
- When the lights on PORTA turn on, press RB0 again.
- The time delay from when the lights turn on and you press RB0 is then recorded and displyed on the LCD.

3) Write a flow-chart for this program

note: you should have two flow charts: one for the main routine, one for the interrupt



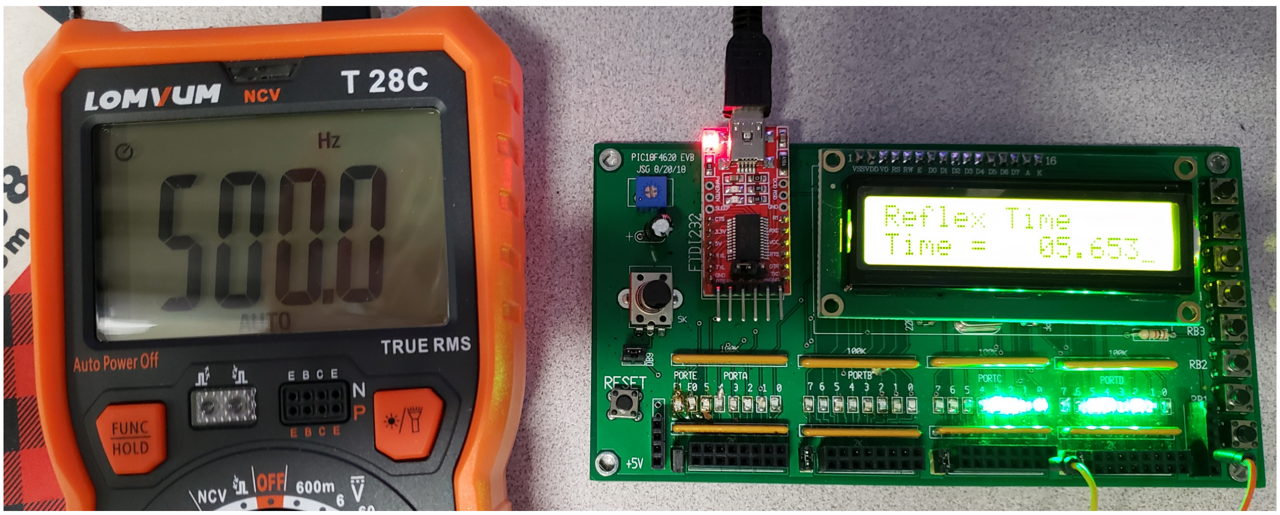
4) Write the corresponding C code

```
Memory Summary:
    Program space        used   B02h (  2818) of 10000h bytes   (  4.3%)
    Data space           used    37h (    55) of   F80h bytes   (  1.4%)
    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%)
```

5) Collect data on your reaction time

Checking the timing

- RC0 outputs 500.0Hz
- Timer2 is running at 1.000ms



Collecting some data

A = {168.2, 169.7, 171.1, 173.5} ms

6) (Population A): From your data, determine

- The 90% confidence interval for your reaction time, and
- The probability that your next trial will be less than 200ms
- The probability tht your average reaction time is less than 200ns

```
>> A = [168.2, 169.7, 171.1, 173.5];
>> Xa = mean(A)

Xa =  170.7500

>> Sa = std(A)

Sa =    2.6300

>> Xa + 2.3534*Sa

ans =  176.9393

>> Xa - 2.3534*Sa

ans =  164.5607
```

The 90% confidence interval for my reaction time is (164.56ms, 176.93ms)

The probability that my reaction time is less than 200ms is:

```
>> t = (200 - Xa)/Sa

t =  11.1219
```

This corresponds to a probability of 0.998

I'm 99.8% likely to have a reaction time less than 200ms

7) (Population B): Change something

Head off to class, came back an hour later to take the test again

Times: (ms)

B = {209.2, 200.6, 180.3, 188.4}

8) Determine the probability that

- A will have a lower reaction time than B in the next trial
- A has a lower average rection time than B

In Matlab

```
>> B = [209.2, 200.6, 180.3, 188.4]

>> Xb = mean(B)

Xb =  194.2500

>> Sb = std(B)

Sb =   12.8160

>> Xw = Xa - Xb

Xw =  -23.5000

>> Sw = sqrt(Sa^2 + Sb^2)

Sw =   13.0831

>> tw = Xw / Sw

tw =   -1.7962
```

From a t-table with 3 degrees of freedom, a t-score of 1.7962 corresponds to a probability of 8%

There is an 8% chance that B will be A in the next trial

```
>> Sw = sqrt(Sa^2 / 4 + Sb^2 / 4)

Sw =    6.5415

>> tw = Xw / Sw

tw =   -3.5924
```

From a t-table with 3 degrees of freedon, a t-score of 3.59 corresponds to a probability of 1.8%

There is a 1.8% chance my overall reaction time was better after class