ECE 376 - Homework #8

Timer 2 Interrupts.   Due Monday, March 27th, 2023
Please email to jacob.glower@ndsu.edu, or submit as a hard copy, or submit on BlackBoard

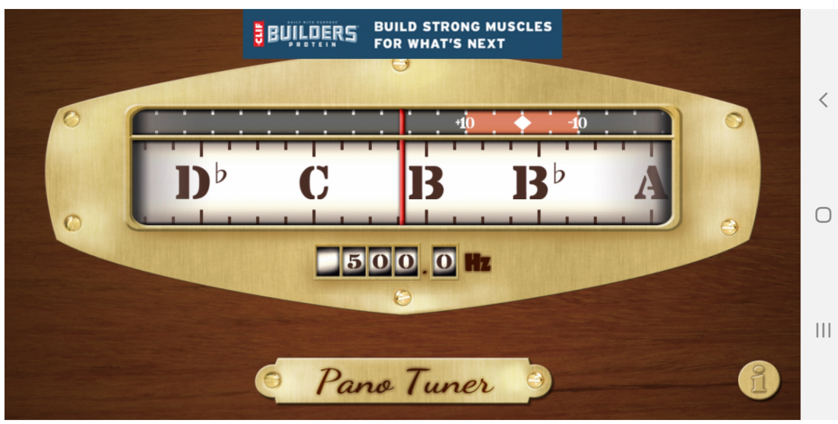
Measuring Time to 1ms with Timer2 Interrupts

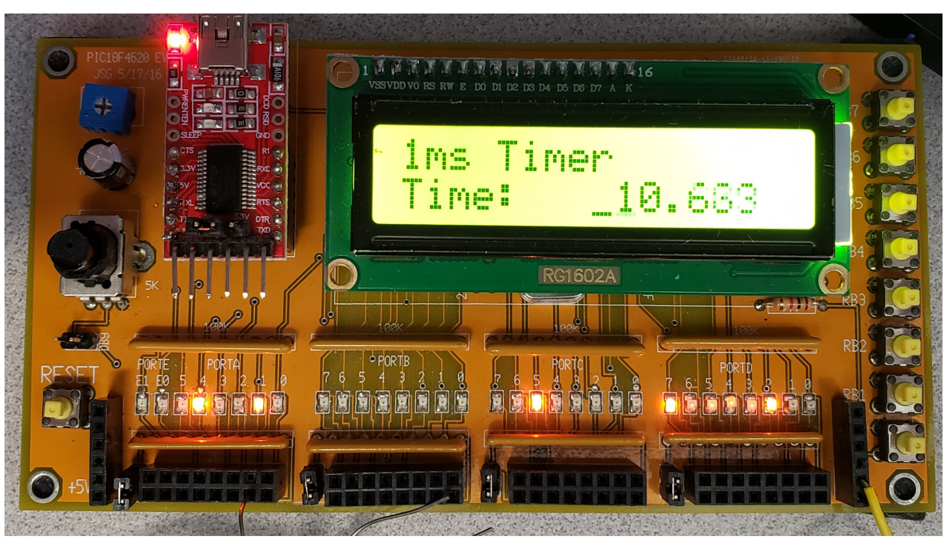
1) Write a routine for a count-down timer with a resolution of 1ms

- Time is measured to 1ms using Timer2 interrupts
- Each interrupt, pin RC0 is toggled (outputting a 500Hz square wave on RC0)
- Each interrupt (every 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 5.000 seconds
- When you press RB1, the time is reset to 10.000 seconds
- When you press RB2, the time is reset to 15.000 seconds
- When you press RB3, the time is reset to 20.000 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: Starting on the rising edge of the button press

```
// Global Variables

const unsigned char MSG0[21] = "1ms Timer           ";
const unsigned char MSG1[21] = "Time:               ";

unsigned int TIME;

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"
```

Risng Edge	Falling Edge

<pre>// High-priority service
void interrupt IntServe(void)
{
   if (TMR2IF) {
      RA1 = !RA1;
      if(TIME) TIME -= 1;
      else {
         if(RB0) TIME = 5000;
         if(RB1) TIME = 10000;
         if(RB2) TIME = 15000;
         if(RB3) TIME = 20000;
      }
      TMR2IF = 0;
      }
   }</pre>
	<pre>// High-priority service
	void interrupt IntServe(void)
	{
	   if (TMR2IF) {
	      RA1 = !RA1;
	      if(TIME) TIME -= 1;
	      if(RB0) TIME = 5000;
	      if(RB1) TIME = 10000;
	      if(RB2) TIME = 15000;
	      if(RB3) TIME = 20000;
	      TMR2IF = 0;
	      }
	   }</pre>

```
// Main Routine

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

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

   ADCON1 = 0x0F;

   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 1ms
   T2CON = 0x4D;
   PR2 = 249;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;

// turn on all interrupts
GIE = 1;

   while(1) {
      LCD_Move(1,  8);  LCD_Out(TIME, 5, 3);
      }
   }
```

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 D2 (73.42Hz) 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)
- note: You might need to use a coutner and toggle RC0 every 4th interrupt.

Calculations: To generate 73.42Hz

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 68,101.33$$

That's bigger than the maximum value of A*B*C, so toggle RC0 every other interrupt

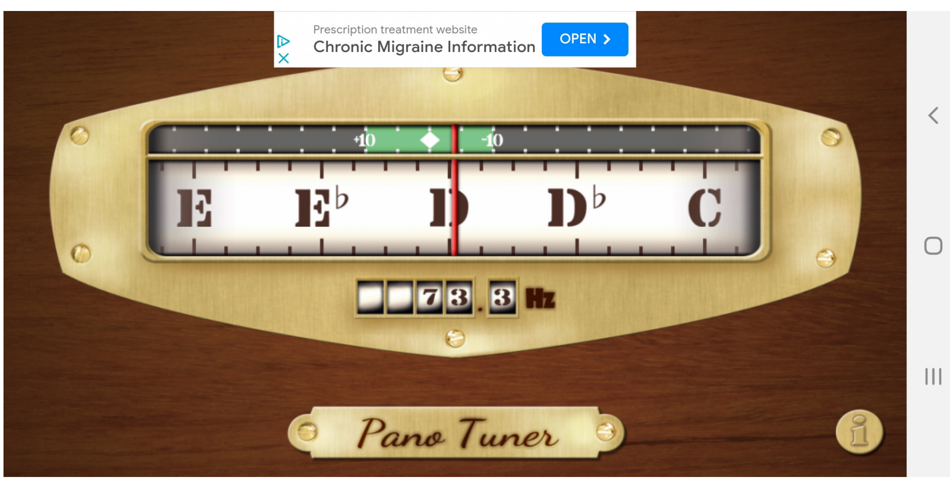
$$N/2 = 34,050.66$$

One combination of A*B*C that's close is

- A = 14, B = 152, C = 16
- A*B*C = 34,048 (-0.0078% low)

T2CONis

T2CON = 0x6F							
7	6	5	4	3	2	1	0
0	1	1	0	1	1	1	1
	A = 14					C = 16	



The C code is then

```
// Global Variables

const unsigned char MSG0[21] = "1-String Banjo      ";
const unsigned char MSG1[21] = "                    ";

unsigned int TIME, N;

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"

// High-priority service
void interrupt IntServe(void)
{
   if (TMR2IF) {
      N = (N + 1)%2;
      if(N == 0) {
         if(RB0) RA1 = !RA1;
         }
      }
   TMR2IF = 0;
   }


// Main Routine

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

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

   ADCON1 = 0x0F;

   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 1ms
   T2CON = 0x6F;
   PR2 = 151;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;

// turn on all interrupts
   GIE = 1;

   while(1) {
      X = X + 1;
      LCD_Move(1,  8);  LCD_Out(X, 3, 1);
      Wait_ms(100);
      }
   }
```

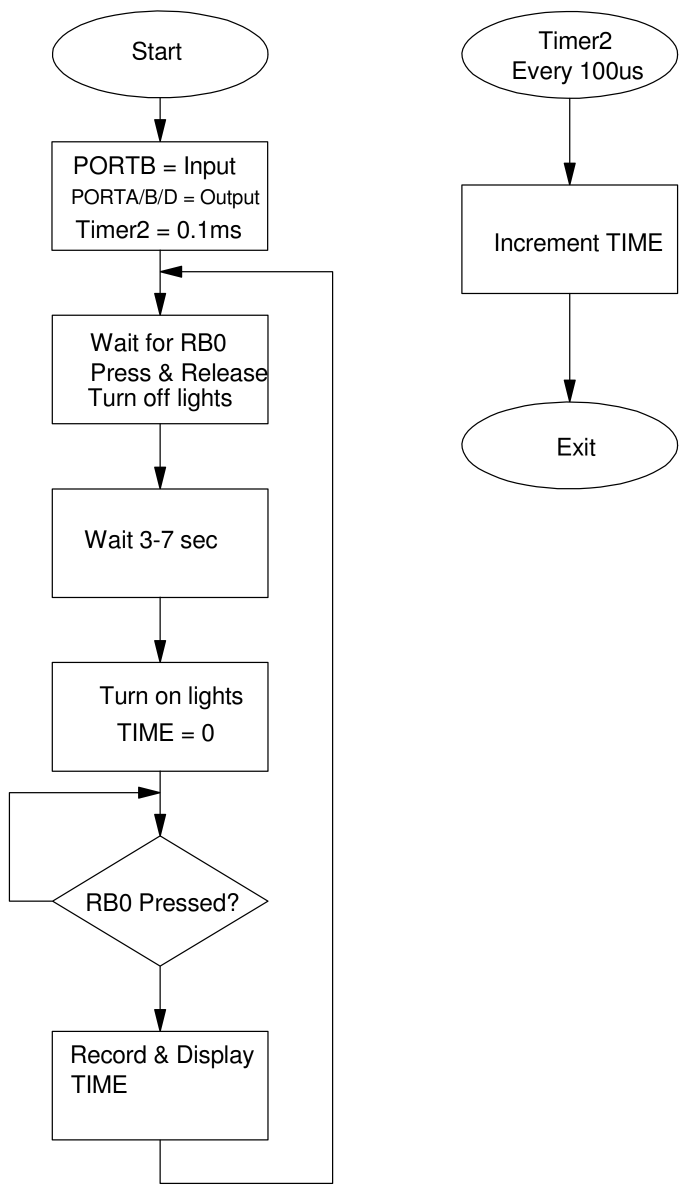
Reflex Timer

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

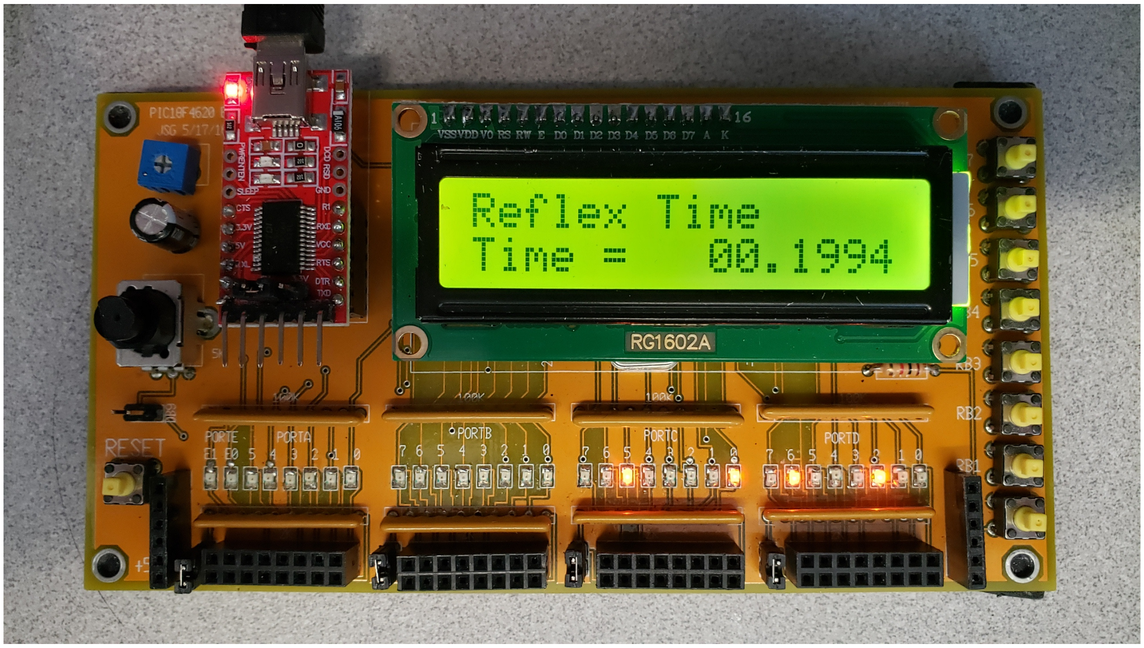


4) Write the corresponding C code

```
// Global Variables

const unsigned char MSG0[21] = "Reflex Time         ";
const unsigned char MSG1[21] = "Time =              ";

unsigned int TIME;

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"

// High-priority service
void interrupt IntServe(void)
{
   if (TMR2IF) {
      RC0 = !RC0;
      TIME = TIME + 1;
      TMR2IF = 0;
      }
}

// Main Routine

void main(void)
{
   unsigned char i;
   unsigned int T0, T1, dT;
   unsigned int DELAY;

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

   ADCON1 = 0x0F;

   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 0.1ms
   T2CON = 0x05;
   PR2 = 249;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;

// turn on all interrupts
   GIE = 1;

   while(1) {
      PORTA = 0;
      PORTE = 0;
      while(!RB0);
      RE0 = 1;
      while(RB0) DELAY = (DELAY + 1)%4000;
      Wait_ms(DELAY + 3000);
      PORTA = 0xFF;
      TIME = 0;
      while(!RB0);
      dT = TIME;
      LCD_Move(1,8);  LCD_Out(dT, 6, 4);
      Wait_ms(1000);
      }
   }
```

5) Collect data on your reaction time

Reflex Times: (Monday, 3:03pm): {0.1883, 0.1819, 0.1844, 0.1824, 0.1994}



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

From Matlab

- mean(A) = 0.1873
- std(A) = 0.0072

From a t-Table with 4 degrees of freedom, 5% tails corresponds to a t-score of 2.1318. The 90% confidence interval for the nex reading is:

```
>> Xa + 2.1318*Sa
    0.2027

>> Xa - 2.1318*Sa
    0.1719
```

The probability that my next reaction will be less than 200ms:

The t-score is:

```
>> t = (0.2 - Xa) / Sa
t =   1.7598
```

From a t-Table with 4 degrees of freedom, this corresponds to a probability of 9%

There is a 9% chance my next reaction time will be more than 200ms

The probability that my average reaction time is less than 200ms (population) is:

```
>> t = (0.2 - Xa) / ( Sa / sqrt(5) )

t =    3.9350
```

From a t-table with 4 degrees of freedom, this corresponds to a probability of 0.009

There is a 99.1% chance that my average reaction time is less than 200ms

7) (Population B): Change something: Record my reaction time a week later after some tea:

- Data: {0.1725, 0.1610, 0.1890, 0.1783, 0.1793}

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

```
>> A = [0.1883, 0.1819, 0.1844, 0.1824, 0.1994];
>> Xa = mean(A);
>> Sa = std(A);

>> B = [0.1725,  0.1610, 0.1890, 0.1783, 0.1793];
>> Xb = mean(B);
>> Sb = std(B);
```

Form a new variable, W = A - B. The mean and standard deviation (individual) are:

```
>> Xw = Xa - Xb

Xw =    0.0113

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

Sw =    0.0126
```

The t-score for A > B (W > 0) is:

```
>> tw = Xw / Sw

tw =    0.8962
```

With four degrees of freedom, this corresponds to a probability of 0.25 for the tail

There is a 75% chance that A > B

my reaction time was better in the morning after tea

For a population, the standard deviation and t-score are:

```
>> Sw = sqrt(Sa^2 / 5  +  Sb^2 / 5)

Sw =    0.0056

>> tw = Xw / Sw

tw =    2.0040
```

From a t-table with 4 degrees of freedom, this corresponds to a probabilty of 0.06 for the tail

There is a 94% chance that the mean of A is greater than the mean of B

my overall reaction time is better in the morning than in the afternoon

on average, my reaction time in the morning is 11.3ms faster than in the afternoon