ECE 376 - Homework #10

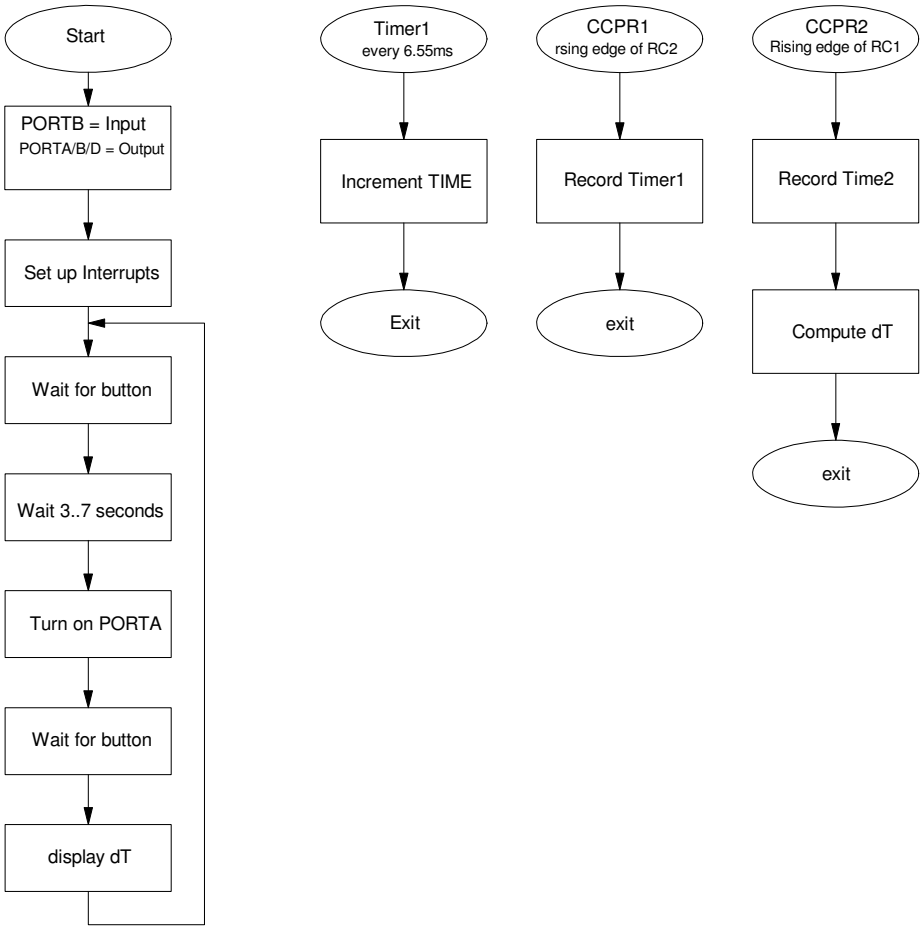
Timer1 Capture & Compare - Due Monday, April 8th

Timer1 Capture

Write a C program to measure your reflex time to 100ns using Timer1 Capture interrupts

- Start the game by pressing RB0. PORTA is turned off when the game starts.
- When pressed, the PIC waits a random time from 4.00 to 7.00 seconds
- After that wait, the lights on PORTA turn on
  - The time of the lights turning on is recorded using Capture1 interrupts
  - Run a wire from PORTA to RC2 to record the rising edge
- When the lights turn on, press RB0 again
  - The time of RB0 is recorded using Capture 2 interrupts
  - Run a wire from RB0 to RC1 to record the rising edge
- The time delay is your reflex time

1) Give a flow chart for this program



2) Write the C code using Timer1 Capture interrupts

```
// Reflex - measure your reflex time to 100ns

#include <pic18.h>

// Global Variables
unsigned long int TIME1, TIME2, dT, TIME;

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

// Interrupt Service Routine

void interrupt IntServe(void)
{
   if (TMR1IF) {
      TIME = TIME + 0x10000;
      TMR1IF = 0;
      }
   if (CCP1IF) {
      TIME1 = TIME + CCPR1;
      CCP1IF = 0;
      }
   if(CCP2IF) {
      TIME2 = TIME + CCPR2;
      dT = TIME2 - TIME1;
      CCP2IF = 0;
      }
   }
```

Main Routine

```
   while(1) {
      PORTE = 0;
      while(!RB0);
      while(RB0);
      N = (TMR1 % 700);
      PORTE = 0xFF;
      Wait_ms(N*10);
      PORTA = 0xFF;
      while(!RB0);
      PORTA = 0;
      LCD_Move(1,0);  LCD_Out(dT, 8, 4);
      Wait_ms(500);
      }
   }
```

3) Validate your code

The delay is between 4.00 and 7.00 seconds:

Modify the code to display the wait time (N):

```
6.460s, 5.950s, 5.450s, 4.050s, 6.630s
```

All delays were in the range of 4..7 seconds

If you press RB0 two seconds after the light turns on, the time reported is 2.000 000 0 (ish)

- Time was 1.4760054 seconds (2.00 seconds ish)

If you press RB0 five seconds after the light turns on, the time reported is 5.000 000 0 (ish)

- 4.9940609 seconds (5.,00 ish)

It looks like the code is working...

4) Record two or more reaction times.

```
191.9911ms,  203.4317ms, 191.4288ms, 194.5970ms, 183.3016ms
```

From your data, determine

- Your mean reflex time
- The standard deviation of this time, and
- The 90% confidence interval for your reaction time

```
>> A = [191.9911, 203.4317, 191.4288, 194.5970, 183.3016]

A =  191.9911  203.4317  191.4288  194.5970  183.3016

>> Xa = mean(A)

Xa =  192.9500

>> Sa = std(A)

Sa =    7.2268

>> Xa + 2.132*Sa

ans =  208.3575

>> Xa - 2.132*Sa

ans =  177.5426
```

Based upon this data, my reaction time should be in the range of (177.54ms, 208.35ms) 90% of the time.

Timer1 Compare

- Step-by-step programming...
- Can you tell the difference between 329.618Hz (E4) and 329.288Hz (0.1% low)?

5) Write a program which plays two notes then pauses for one second:

- 329.618Hz (E4) plays on RC2 for 500ms using Timer1 Compare1 interrupts,
- It pauses for 200ms, then
- 349.228Hz (F4) plays on RC2 for 500ms
- It pauses for 1000ms, then
- Repeats

Check that the two notes play (it should be easy to hear the difference)

yes - easy to tell the difference

Calculations:

$$N_{E4} = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 15.169.07$$

$$N_{F4} = 14,317.2941$$

Interrupt:

```
// Interrupt Service Routine

void interrupt IntServe(void)
{
   if (TMR1IF) {
      TIME = TIME + 0x10000;
      TMR1IF = 0;
      }

   if (CCP2IF) {
      if(PLAY) CCP2CON ^= 1;
      CCPR2 += N2;
      CCP2IF = 0;
      }
   }
```

Main Routine

```
   while(1) {
      N2 = 15169;
      PLAY = 1;
      Wait_ms(500);
      PLAY = 0;
      Wait_ms(200);
      N2 = 14317;
      PLAY = 1;
      Wait_ms(500);
      PLAY = 0;
      Wait_ms(1000);
      }
```

6) Modify this code so that when you press RB0, the code flips a coin

- COIN = TMR1 & 1 should work
- If the coin is heads, play the same note twice in a row
- If the coin is tails, play note E4 then F4

Check that two notes play, with the second being random (sometimes E4, sometimes F4)

Results:

- Same
- Different
- Same
- Same
- Different

Looks OK

Main Routine (no change in interrupts)

```
while(1) {
   while(!RB0);
   Wait_ms(1000);
   COIN = TMR1 & 1;
   N2 = 15169;
   PLAY = 1;
   Wait_ms(500);
   PLAY = 0;
   Wait_ms(200);
   if(COIN) N2 = 14317;
   else     N2 = 15169;
   PLAY = 1;
   Wait_ms(500);
   PLAY = 0;
   Wait_ms(1000);
   }
```

7) Modify this code so that you can then guess if the notes are the same or different

- RB1 means the notes are different
- RB0 means the notes are the same
- After the two notes, the code waits for you to press RB1 or RB0
- If you are correct, a counter is updated and displayed (RIGHT += 1)
- If you are incorrect, a counter is updated and displayed (WRONG += 1)

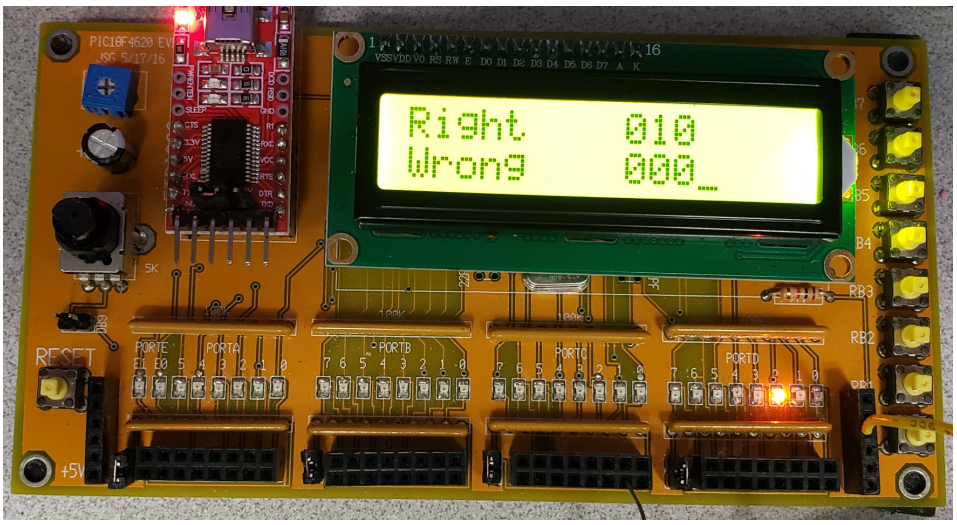
Check that the code is working:

- Two notes play, with the second being the same or different randomly
- When you press a button, it tallies your correct / incorrect responses accordingly

Notes are random (same or different)

- When the notes are the same,
  - RB1 increases RIGHT
  - RB0 increases WRONG
- When the notes are different
  - RB1 increases WRONG
  - RB0 increases RIGHT

With 10 notes, I'm correct 10 / 10 times



From a chi-squared test, I'm not just guessing (I can really hear the difference)

Case	p	np	N	chi-squared
right	0.5	5	10	5
wrong	0.5	5	0	5
			Total:	**10**

p = 0.9984 (99.84% chance I'm not just guessing)

```
while(1) {
   Wait_ms(1000);
   COIN = TMR1 & 1;
   N2 = 15169;
   PLAY = 1;
   Wait_ms(500);
   PLAY = 0;
   Wait_ms(200);
   if(COIN) N2 = 14317;
   else     N2 = 15169;
   PLAY = 1;
   Wait_ms(500);
   PLAY = 0;
   while(PORTB == 0);
   if(RB2) {
      RIGHT = 0;
      WRONG = 0;
      }
   else {
      if( RB0 ^ COIN) WRONG += 1;
      else RIGHT += 1;
   }
   LCD_Move(0,8); LCD_Out(RIGHT, 3, 0);
   LCD_Move(1,8); LCD_Out(WRONG, 3, 0);
   }
}
```

8) Modify this code so that it plays

- 329.618Hz (E4) and
- 329.288Hz (0.1% low)

Run the experiment 10 or more times and record your correct / incorrect results.

$$N_{E4} = 15,169.07$$

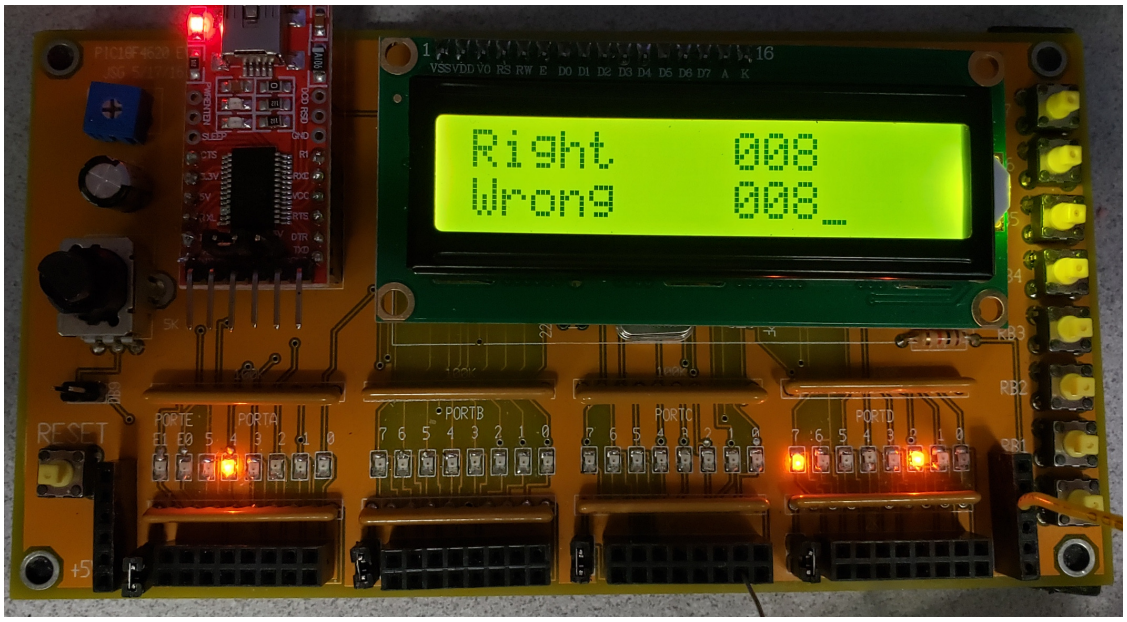
$$N_{E4\#} = 15,184.27$$

```
void interrupt IntServe(void)
{
   if (TMR1IF) {
      TIME = TIME + 0x10000;
      TMR1IF = 0;
      }

   if (CCP2IF) {
      if(PLAY) CCP2CON ^= 1;
      CCPR2 += N2;
      CCP2IF = 0;
      }
   }

// Subroutines
#include        "lcd_portd.c"

   while(1) {
      Wait_ms(1000);
      COIN = TMR1 & 1;
      N2 = 15169;
      PLAY = 1;
      Wait_ms(500);
      PLAY = 0;
      Wait_ms(200);
      if(COIN) N2 = 15184;
      else     N2 = 15169;
      PLAY = 1;
      Wait_ms(500);
      PLAY = 0;
      while(PORTB == 0);
      if(RB2) {
         RIGHT = 0;
         WRONG = 0;
         }
      else {
         if( RB0 ^ COIN) WRONG += 1;
         else RIGHT += 1;
      }
      LCD_Move(0,8); LCD_Out(RIGHT, 3, 0);
      LCD_Move(1,8); LCD_Out(WRONG, 3, 0);
      }
   }
```

Result:

- Right 8 times
- Wrong 8 times

I'm just guessing (chi-squared score = 0)


Right.
Wrong
008
003
RESET
PORTE
PORTA
PORTB
PORTC
PORTD

9) Use a chi-squared test to determine if you were guessing or if you could really hear a difference

Just for fun, try 100 clocks (0.65% change)

```
if(COIN) N2 = 15269;
else     N2 = 15169;
```

With this difference, I got it right 12/12 times (12 right, 0 wrong)

Case	p	np	N	chi-squared
right	0.5	6	12	6
wrong	0.5	6	0	6
			Total:	**12**

From StatTrek, p = 0.99947

There is a 99.947% chance I'm not just guessing

Try a change of 50 clocks (0.329% change)

```
if(COIN) N2 = 15219;
else     N2 = 15169;
```

With this difference, I got it right 12/13 times (12 right, 1 wrong)

Case	p	np	N	chi-squared
right	0.5	6.5	12	4.6538
wrong	0.5	6.5	1	4.6538
			Total:	**9.3077**

From StatTrek, p = 0.9972

There is a 99.72% chance I'm not just guessing

Try a change of 25 clocks (0.1648%)

```
if(COIN) N2 = 15291;
else     N2 = 15169;
```

With this difference, I got it right 13/21 times (13 right, 8 wrong)

Case	p	np	N	chi-squared
right	0.5	11.5	13	0.1957
wrong	0.5	11.5	8	0.1957
			Total:	0.3913

From StatTrek, with one degree of freedom this corresponds to 0.46838

There is a 46.838% probability of rejecting the null hypothesis (I'm just guessing)

Try a change of 15 clocks (0.1%)

```
if(COIN) N2 = 15184;
else     N2 = 15169;
```

With this difference, I got it right 8/16 times (8 right, 8 wrong)

Case	p	np	N	chi-squared
right	0.5	8	8	0
wrong	0.5	8	8	0
			Total:	0

There is a 0% chance I'm not just guessing

Net result: It looks like I can consistently hear a 0.329% difference in frequency (or more)