ECE 376 - Homework #4

C Programming and LCD Displays. Due Monday, September 25th
Please submit as a hard copy, submit on BlackBoard, or email

1) Determine how many clocks the following C code takes to execute

- Compile and download the code (modify working code and replace the main loop)
- Measure the frequency you see on RC0 (toggles every loop).
  - Use an osiclloscope - or -
  - Connect a speaker to RC0 with a 200 Ohm resistor and measure the frequency with a cell phone app like Piano Tuner
  - RC1 is 1/2 the frequency of RC0, RC2 is 1/4th, RC3 = 1/8th, etc
- The number of clocks it takes to execute each loop is

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

1a) Counting mod 16

```
unsigned char i
while(1) {
      i = (i + 1) % 16;
      if(i == 0) PORTC += 1;
      }
```

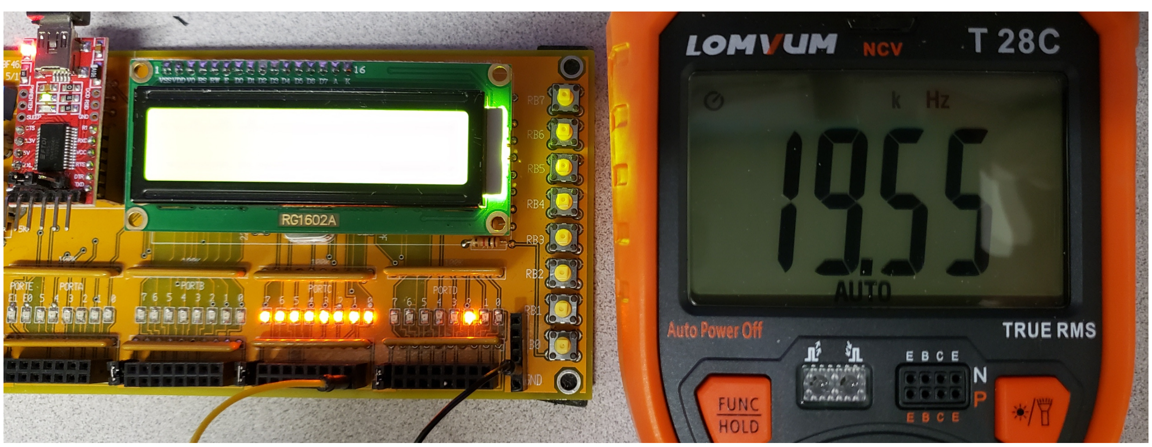
f = 19.55kHz

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 255.75 \qquad \textit{clocks per toggle}$$

$$N/16 = 15.98 \qquad \textit{clocks per loop}$$

It takes about 16 clocks to count mod 16

- ***a little high since this also includes the time to toggle PORTC and loop back***



1b) Counting mod 17

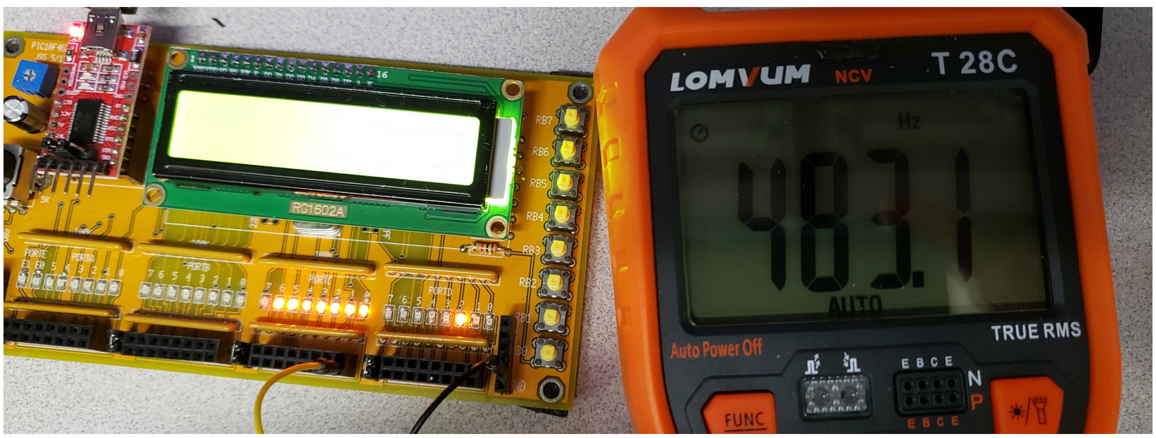
```
unsigned char i
while(1) {
      i = (i + 1)% 17;
      if(i == 0) PORTC += 1;
      }
```

f = 483.1Hz

$$N = \left(\frac{10{,}000{,}000}{2 \cdot 483.1}\right) = 10,349.82 \qquad \textit{clocks per toggle}$$

$$N/17 = 608.81 \qquad \textit{clocks per loop}$$

It takes about 608 clocks to count mod 17



1c) Floating Point Division

```
float A, B, C;
A = sqrt(3);
B = sqrt(2);
while(1) {
    i = (i + 1)% 16;
    if(i == 0) PORTC += 1;
    C = A/B;
    }
```

f = 161.8Hz

$$N = \left(\frac{10{,}000{,}000}{2 \cdot 161.8}\right) = 30{,}902.34$$ *clocks per toggle*

$$N - 255.75 = 30{,}646.59$$ *removing time to count mod 16*

$$\left(\frac{30{,}646.59}{16}\right) = 1915.41$$

Each floating point division takes 1915 clocks

1d) Double Precision Floating Point Division

```
double A, B, C;
A = sqrt(3);
B = sqrt(2);
while(1) {
    i = (i + 1)% 16;
    if(i == 0) PORTC += 1;
    C = A/B;
    }
```

f = 161.8Hz

This C compiler doesn't differentiate between floating point and double precision floating point operations.

Beep

2) Write a C program which plays 174.61Hz (note F3) for 50ms on a speaker

```
void Beep(void) {
:
:
}
```

Note: 50ms is equal to

$$n = \left(174.61\frac{cycles}{\sec}\right)(50ms) = 8.73 \text{ cycles}$$

$$= 17.46 \text{ edges}$$

Try the following code and measure the frequency on RC0:

```
// ---  HW4.C -------------------

// Global Variables

// Subroutine Declarations
#include <pic18.h>

void Beep(void) {
   unsigned int i, j;
   for(i=0; i<17; i++) {
      RC0 = !RC0;
      for(j=0; j<1000; j++);
      }
   }

// Main Routine

void main(void)
{
   unsigned long int i;

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

   while(1) {
      Beep();
 //     for(i=0; i<100000; i++);
      }
   }
```

f = 312.2Hz

To make the frequency 174.61Hz, adjust the counter in Beep()

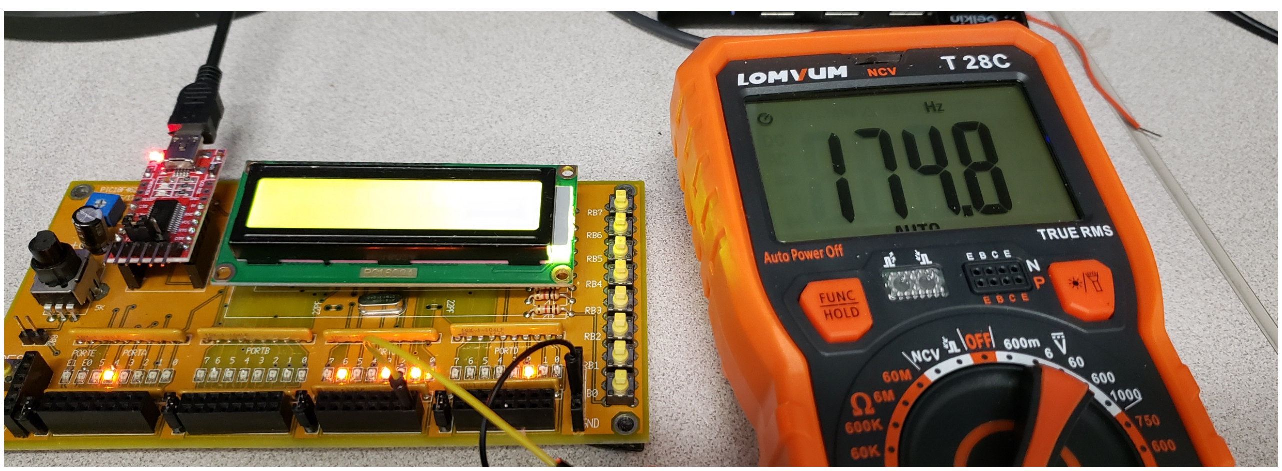
$$n = \left(\frac{312.2Hz}{174.61Hz}\right)1000 = 1787.93$$

It's now 174.8Hz

Add in a wait loop

```
// ---  HW4.C -------------------

// Global Variables

// Subroutine Declarations
#include <pic18.h>

void Beep(void) {
   unsigned int i, j;
   for(i=0; i<17; i++) {
      RC0 = !RC0;
      for(j=0; j<1788; j++);
      }
   }

// Main Routine

void main(void)
{
   unsigned long int i;

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

   while(1) {
      Beep();
      for(i=0; i<100000; i++);
      }
   }
```

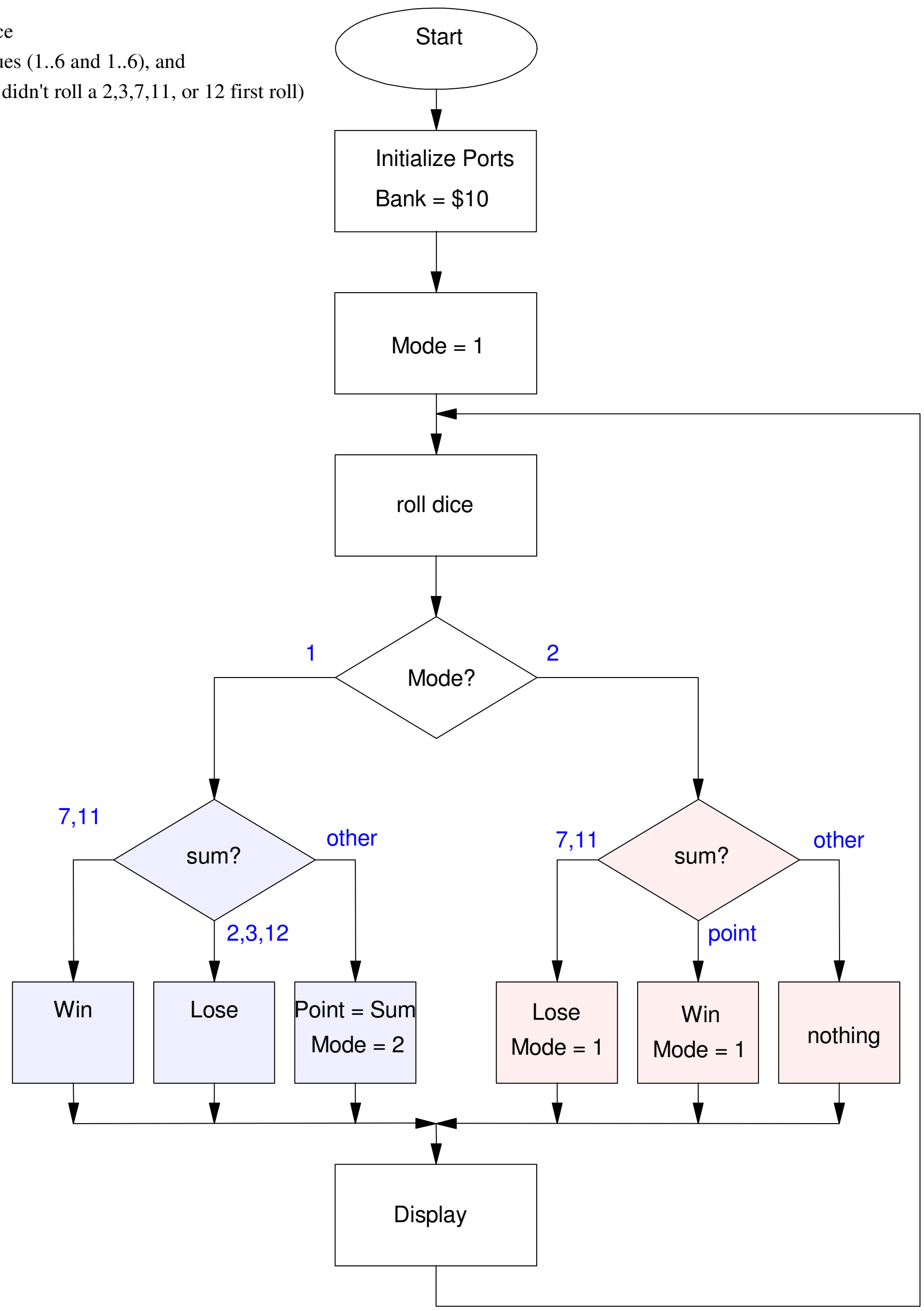
3) Verify the frequency and duration of your note



$65 Craps Table

4) Give a flow chart for a program which turns your PIC into a Craps Table:

- On reset, you start with $10 in your bank (which is displayed on the LCD).
- The game starts by pressing a button RB0. The bet is $1 (fixed).
- When you press and release RB0, it rolls two 6-sided dice
  - hint: count mod 36. Die #1 is count mod 36. Die #2 is count/6.
- If you roll 7 or 11, you win (bank increases by $1)
- If you roll 2, 3, or 12, you lose (bank decreases by $1);
- If you roll a different number, that's your point. On RB0, you roll again.
  - If you roll your point, you win
  - If you roll 7 or 11, you lose
  - If you roll a different number, nothing happens.
  - Keep playing until you win or lose
- On the LCD, display
  - Your bank balance
  - The two dice values (1..6 and 1..6), and
  - The point (if you didn't roll a 2,3,7,11, or 12 first roll)



5) Write the C code for a craps table

Go in steps

- Get it to display the dice, the sum, and the bank value
- Get it to roll two 6-sided dice
- Get it to win when I roll 7 or 11
- Get it to lose when I roll 2, 3, 12
- Get it to keep playing in the other case

Net Code:

```
// Global Variables

const unsigned char MSG0[20] = "Craps Game          ";
const unsigned char MSG1[20] = "                    ";


// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"

void Beep(void) {
   unsigned int i, j;
   for(i=0; i<17; i++) {
      RC0 = !RC0;
      for(j=0; j<1788; j++);
      }
   }

void Roll(char* d1, char* d2) {
   unsigned char n;
   while(!RB0);
   while(RB0) {
      n = (n + 1)%36;
      }
   *d1 = n/6 + 1;
   *d2 = n%6 + 1;
   }

void Display(char d1, char d2, int Point, int Bank) {
   LCD_Move(1,0);  LCD_Out(d1, 1, 0);
   LCD_Move(1,2);  LCD_Out(d2, 1, 0);
   LCD_Move(1,6);  LCD_Out(Point, 2, 0);
   LCD_Move(1,12);  LCD_Out(Bank, 2, 0);
   }
```

```
// Main Routine

void main(void)
{
   unsigned int i;
   unsigned int Bank, Point;
   unsigned char DICE, d1, d2, Sum;
   unsigned char Flag;

   TRISA = 0;
   TRISB = 0xFF;
   TRISC = 0;
   TRISD = 0;
   TRISE = 0;
   ADCON1 = 0x0F;
   Bank = 10;

   LCD_Init();

   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(70);

   Bank = 10;
   Mode = 1;

   while(1) {

      :
      :
      :
      :
      :

      }
  }
}
```

6) Verify your program

- On reset, you start with $10 in your bank
- Numbers generated are random: two dice each in the range of 1..6
- The LCD displays information correctly
- When you win, you gain $1. When you lose, you lose $1.

Going step by step

- Display routine displays four numbers (d1, d2, point, bank)
- Dice rolls two 6-sided dice
- Win on 7 or 11
- Lose on 2, 3, 12
- Set up a point on other numbers
- Keeps rolling until I hit the point (win) or 7 or 11 (lose)

I tend to lose (house has the advantage)

7) (20pt) Demonstration (in person or on a video)