ECE 376 - Homework #9

Timer 0/1/2/3 Interrupts. Due Monday, April 3rd
Please email to jacob.glower@ndsu.edu, or submit as a hard copy, or submit on BlackBoard

1) Write a C routine using Timer0 interrupts to measure time to 100ns. Using this routine, determine how long a the following operations in C take:

a) LCD display routine

```
long int A;
A = 3141592654;
LCD_Out(A, 10, 9);
```

- Time = 6.2543ms

b) The time it takes you to press all buttons on PORTB sequentially

```
TRISB = 0xFF;
while(!RB0);
while(!RB1);
while(!RB2);
while(!RB3);
while(!RB4);
while(!RB5);
while(!RB6);
while(!RB7);
```

c) The time it takes you to press and release RB0 10 times

```
TRISB = 0xFF;
for(i=0; i<10; i++) {
   while(!RB0);
   while(RB0);
   }
```

2) Write a C routine using Timer0 / Timer1 / Tirme2 / Timer3 interrupts to play 4 notes at the same time when you press button RB0 (4-string Violin)

Output Pin	RC0	RC1	RC2	RC3
Note	C3	D3	E3	F3
Frequency (Hz)	130.81 Hz	146.83 Hz	164.81 Hz	174.61 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3

Output Pin	RC0	RC1	RC2	RC3
Note	C3	D3	E3	F3
Frequency (Hz)	130.81 Hz	146.83 Hz	164.81 Hz	174.61 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3
N	38,223.377	34,052.986	30,337.96	28,635.24
PS	1	1	A = 8, C = 16 B = 237	1
Measured	**130.6 Hz**	**146.6 Hz**	**164.7 Hz**	**174.6 Hz**
Error (%)	**-0.16%**	**-0.15%**	**-0.07%**	**-0.01%**

Note: T2CON is as follows

T2CON = 0x3F							
7	6	5	4	3	2	1	0
0	0	1	1	1	1	1	1
	A = 8					C = 16	

Code:

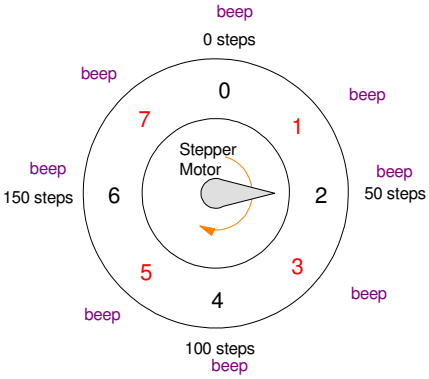
```
void interrupt IntServe(void)
{
   if (TMR0IF) {
      TMR0 = -38223 + 50;
      RC0 = !RC0;
      TMR0IF = 0;
      }
   if (TMR1IF) {
      TMR1 = -34052 + 50;
      RC1 = !RC1;
      TMR1IF = 0;
      }
   if (TMR2IF) {
      RC2 = !RC2;
      TMR2IF = 0;
      }
   if (TMR3IF) {
      TMR3 = -28635   + 50;
      RC3 = !RC3;
      TMR3IF = 0;
      }
   }
```

Roulette Wheel

Use multiple interrupts to create a Roulette wheel which drives a stepper motor:

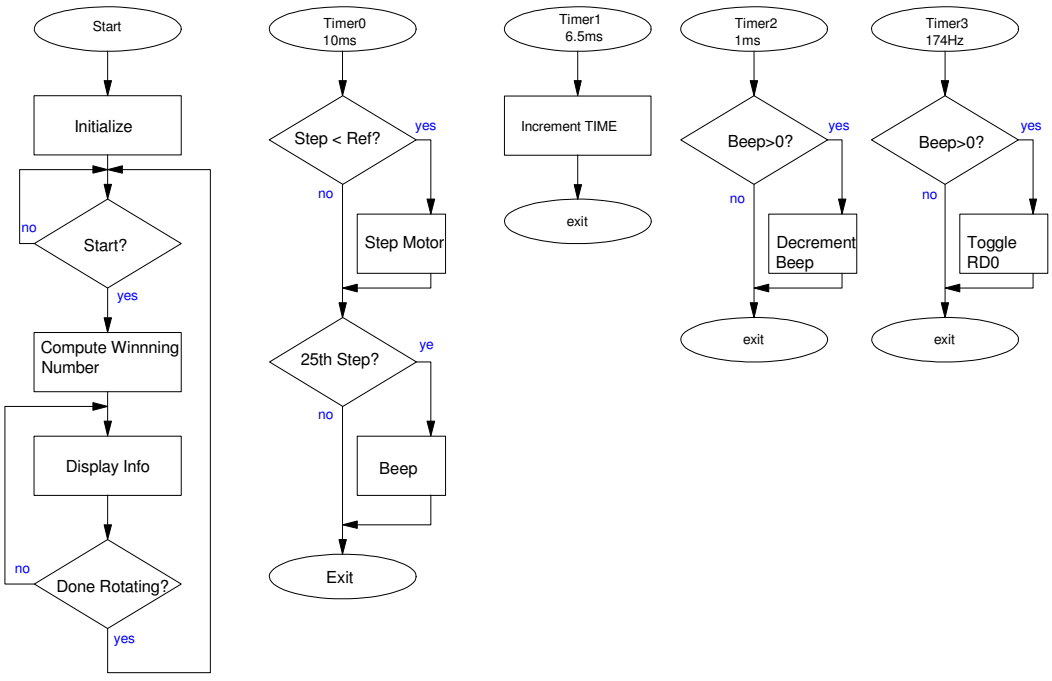
- Timer0: Set to 10ms. Steps the motor every 10ms
- Timer1: Keeps track of time to 100ns. Also used to generate random numbers
- Timer2: Set to 1ms. Controls the duration of the beep noise (100ms beep)
- Timer3: Set to 174.61Hz. Sets the frequency of the note to F3 (174.61Hz)

- Start the game by pressing RB0.
  - This generates a random number, N, in the range of 0..7 by taking the current time (TMR1) mod 8.
- When RB0 is pressed, the stepper motor then turns three rotations (600 steps) plus 25*N steps
- The stepper motor spins at 10ms per step
- Every 25th step (each number), the speaker plays note F3 for 100ms
- The winning number is the the angle of the stepper motor, mod 200
  - Winning Number = (STEPS mod 200) / 8
- The LCD displays
  - The current number the stepper motor is pointing at
  - The current time, accurate to 100ns (Timer1)



3) Give the flow charts for this program

- note: you need a separate flow chart for the main routine and each interrupt



4) Write the corresponding C code

```
// Global Variables

const unsigned char MSG0[21] = "Roulette Wheel      ";
const unsigned char MSG1[21] = "                    ";
const unsigned char TABLE[4] = {8, 4, 2, 1};

unsigned int TIME, STEP, REF;
unsigned int BEEP;
unsigned char FLAG;

// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include       "lcd_portd.c"

// High-priority service
void interrupt IntServe(void)
{
   if (TMR0IF) {   // PS = 8
      TMR0 = -12500;
      if(STEP < REF) {
         STEP += 1;
         if((STEP % 25) == 0)
            if(BEEP == 0) BEEP = 100;
      }
      PORTA = TABLE[STEP % 4];
      TMR0IF = 0;
      }
   if (TMR1IF) {
      TIME += 0x10000;
      TMR1IF = 0;
      }
   if (TMR2IF) {
      RE0 = !RE0;
      if(BEEP) BEEP -= 1;
      TMR2IF = 0;
      }
   if (TMR3IF) {
      TMR3 = -28635 + 50;
      if(BEEP) RD0 = !RD0;
      else RD0 = 0;
      TMR3IF = 0;
      }
   }

unsigned char Display(X)
{
  unsigned char Y;
  if(X == 0) Y = 1;
  if(X == 1) Y = 2;
  if(X == 2) Y = 4;
  if(X == 3) Y = 8;
  if(X == 4) Y = 16;
  if(X == 5) Y = 32;
  if(X == 6) Y = 64;
  if(X == 7) Y = 128;
  return(Y);
}
```

```
void main(void)
{
   unsigned int i, j, N, NUMBER, N0;

   TRISA = 0;
   TRISB = 0x01;
   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 Timer0 for PS = 8
   T0CS = 0;
   T0CON = 0x82;
   TMR0ON = 1;
   TMR0IE = 1;
   TMR0IP = 1;
   PEIE = 1;
// set up Timer1 for PS = 1
   TMR1CS = 0;
   T1CON = 0x81;
   TMR1ON = 1;
   TMR1IE = 1;
   TMR1IP = 1;
   PEIE = 1;
// set up Timer2 for 1ms;
   T2CON = 0x4D;
   PR2   = 249;
   TMR2ON = 1;
   TMR2IE = 1;
   TMR2IP = 1;
   PEIE = 1;
// set up Timer3 for PS = 1
   TMR3CS = 0;
   T3CON = 0x81;
   TMR3ON = 1;
   TMR3IE = 1;
   TMR3IP = 1;
   PEIE = 1;
// turn on all interrupts
   GIE = 1;
   TRISD0 = 0;
   N0 = 0;
   while(1) {
      while(!RB0);
      while(RB0);
      N = (TMR0 % 8);
      REF = REF % 200;
      STEP = STEP % 200;
      REF = 600 + N*25;
      while(STEP != REF) {
         NUMBER = (STEP % 200) / 25;
         PORTC = Display(NUMBER);
         LCD_Move(1,0);  LCD_Out(NUMBER, 1, 0);
         LCD_Move(1,3);  LCD_Out(N, 1, 0);
         LCD_Move(1,6);  LCD_Out(REF, 4, 0);
         LCD_Move(1,11);  LCD_Out(STEP, 4, 0);
         }
      NUMBER = (STEP % 200) / 25;
      LCD_Move(1,0);  LCD_Out(NUMBER, 1, 0);
      LCD_Move(1,3);  LCD_Out(N, 1, 0);
      LCD_Move(1,6);  LCD_Out(REF, 4, 0);
      LCD_Move(1,11);  LCD_Out(STEP, 4, 0);
      }
   }
```

5) Validation: Verify your code works

Winning numbers are random in the range of 0..7

yes - winning numbers are {0, 3, 4, 4, 3, 5, 2, 7, 6, 7}

The stepper motor is spinning at 10ms/step

yes - two seconds per rotation

The beep noise is at 174.61Hz

measured = 174.6Hz (from problem #2)

The duration of each beep is 100ms

Interrupt set to 1.00ms (measured 500Hz on pin RE0)

6) Statistical Analysis: Use a chi-squared test to determine if using the current time (mod 8) generates a uniform distribution (all numbers have equal probability)

With only ten data points, eight bins are a bit much. Just do even/odd

Bin	p	np	N	chi-squared
Even	0.5	5	5	0
Odd	0.5	5	5	0
			Total	0

This corresponds to a probability of 0%

With only ten data points, I cannot reject the null hypothesis (this is a fair die roll)

7) Demonstration (20pt). In person or on a video