ECE 376 - Test #3: Name _____

Fall 2021. Open-Book, Open Note

1) Single Interrupt: Traffic Light, Write a program which uses Timer2 interrupts to control the timing of a traffic light. The red / yellow / green times are to be:

- Green: X seconds (RC0 = 1, RC1 = 0, RC2 = 0) (turn on the green light)
- Yellow = 1 sec (RC0 = 0, RC1 = 1, RC2 = 0) (turn on the yellow light)
- Red: 5 seconds (RC0 = 0, RC1 = 0, RC2 = 1) (turn on the red light)
- repeat

where

- X = 5 + 0.01*A2D seconds (5 to 15 seconds, depending upon traffic)

Timer2 Initialization:

N # clcoks between interrupts	A	B	C
10,000 (1ms)	10	250	4

Main Routine - main loop
cycle from green to yellow to red & repeat
Assume Timer2, A/D, etc are intialized

```
while(1) {
   X = 1000 + 10*A2D_Read(0);
   PORTC = 1;
   DELAY = X;
   while(DELAY);

   PORTC = 2;
   DELAY = 1000;
   while(DELAY);

   PORTC = 4;
   DELAY = 5000;
   while(DELAY);
   }
```

Timer2 Interrupt Routine

```
void Interrupt(void) {
if(TMR2IF) {
   if(DELAY)DELAY -= 1;
   TMR2iF = 0;
   }
```

2) Multiple Interrupts: Telephones operate by generating dual tones when you press a button. Use interrupts to turn your PIC into a dual tone generator that works for numbers 1 (RB1) and 5 (RB5)

- When RB1 is pressed, RC0 plays 687Hz & RC1 plays 1209Hz for 200ms
- When RB5 is pressed, RC0 plays 770Hz & RC1 plays 1336Hz for 200ms

```
// Global Variables

unsigned int N0, N1;
unsigned char PLAY;

// main loop and interrupts:  (specify these sections of code)
```

Main Routine
monitor the buttons, controls the interrupts

```
while(1) {
   if(RB1) {
      N0 = 7278;
      N1 = 4135;
      PLAY = 1;
      Wait_ms(200);
      PLAY = 0;
      }
   if(RB5) {
      N0 = 6493;
      N1 = 3742;
      PLAY = 1;
      Wait_ms(200);
      PLAY = 0;
      }
   while(PORTB);
   }
```

Timer0
plays a note on RC0

```
if(TMR0IF) {
   TMR0 = -N0;
   if(PLAY) RC0 = !RC0;
   else RC0 = 0;
   TMR0IF = 0;
   }
```

Timer1
plays a note on RC1

```
if(TMR1IF) {
   TMR1 = -N1;
   if(PLAY) RC1 = !RC1;
   else RC1 = 0;
   TMR1IF = 0;
   }
```

3) Timer1 Capture: Write a program which uses Timer1 Capture interrupts to monitor a game show.

- As the start of the game, the host presses RB0. This clears the contestant's lights (RA2 = 0, RA1 = 0);
- The host then reads a question. If a contestant thinks they know the answer, they press their button.
  - RC2 = Player A (Capture 1)
  - RC1 = Player B (Capture 2)
- If contestant A presses their button and B does not, A wins (RA2 = 1, RA1 = 0)
- If contestant B presses their button and A does not, B wins (RA2 = 0, RA1 = 1)
- If both contenstants press their buttons, whoever pressed their button first wins
  - Times recorded by Timer1 Capture interrupts, accurate to 100ns

Specify the global variables used, the main loop, and each interrupt

```
// Global variables

unsigned long int TIME, TA, TB;
```

```
// Interrupts
```

Timer1	Capture1 rising edge on RC2 (player A)	Capture2 rising edge on RC1 (Player B)

`if(TMR1IF) {`
`  TIME = TIME + 0x10000;`
`  TMR1IF = 0;`
`  }`
	`if(CCP1IF) {`
	`  TA = TIME + CCPR1;`
	`  CCPR1IF = 0;`
	`  }`
		`if(CCP2IF) {`
		`  TB = TIME + CCPR2;`
		`  CCPR2IF = 0;`
		`  }`

```
while(1) {
   while(!RB0);
   PORTA = 0;

   while( (RA1 == 0) & (RA2 == 0) ) {
      if( (RC2 == 1) & (RC1 == 0) ) RA2 = 1;
      if( (RC2 == 0) & (RC1 == 1) ) RA1 = 1;
      if( (RC2 == 1) & (RC1 == 1) ) {
          if(TA < TB) RA2 = 1;
          else RA1 = 1;
          }
       }
   }
```

4) Filter Design: Design a digital filter, G(z), which has approximately the same gain vs. frequency as

$$G(s) = \left(\frac{20(s+5)}{s+20}\right)$$

Assume a sampling rate of T = 0.01 second.

Convert poles and zeros as

$$z = e^{sT}$$

$$s = -5 \qquad z = e^{sT} = 0.9512$$

$$s = -20 \qquad z = e^{sT} = 0.8187$$

so

$$G(z) = k\left(\frac{z-0.9512}{z-0.8187}\right)$$

Pick 'k' to match the DC gain

$$\left(\frac{20(s+5)}{s+20}\right)_{s=0} = 5$$

$$k\left(\frac{z-0.9512}{z-0.8187}\right)_{z=1} = 5$$

$$k = 3.5340$$

so

$$G(z) = 3.5340\left(\frac{z-0.9512}{z-0.8187}\right)$$

5) Filter Coding: Write a C program to implement the following filter. Assume a sampling rate of T = 0.01 second.

$$Y = \left(\frac{0.01(z-0.9)}{(z-0.8)(z-0.7)} \right) X = \left(\frac{0.01z-0.009}{z^2-1.5z+0.56} \right) X$$

```
while(1) {
   x2 = x1;
   x1 = x0;
   x0 = A2D_Read(0);

   y2 = y1;
   y1 = y0;
   y0 = 1.5*y1 - 0.56*y2 + 0.01*x1 - 0.009*x2;

   D2A(y0);

   Wait_10ms();

   }
```