ECE 376 - Homework #10

Timer1 Capture / Compare. Due Wednesday, April 12th
Please email to jacob.glower@ndsu.edu, or submit as a hard copy, or submit on BlackBoard

$65 Theremin

A Theramin is a musical instrument which is controlled by moving your hands around the device.

- One hand controls the frequency (this homework set)
- The other hand controls the amplitude (not part of this homework set)

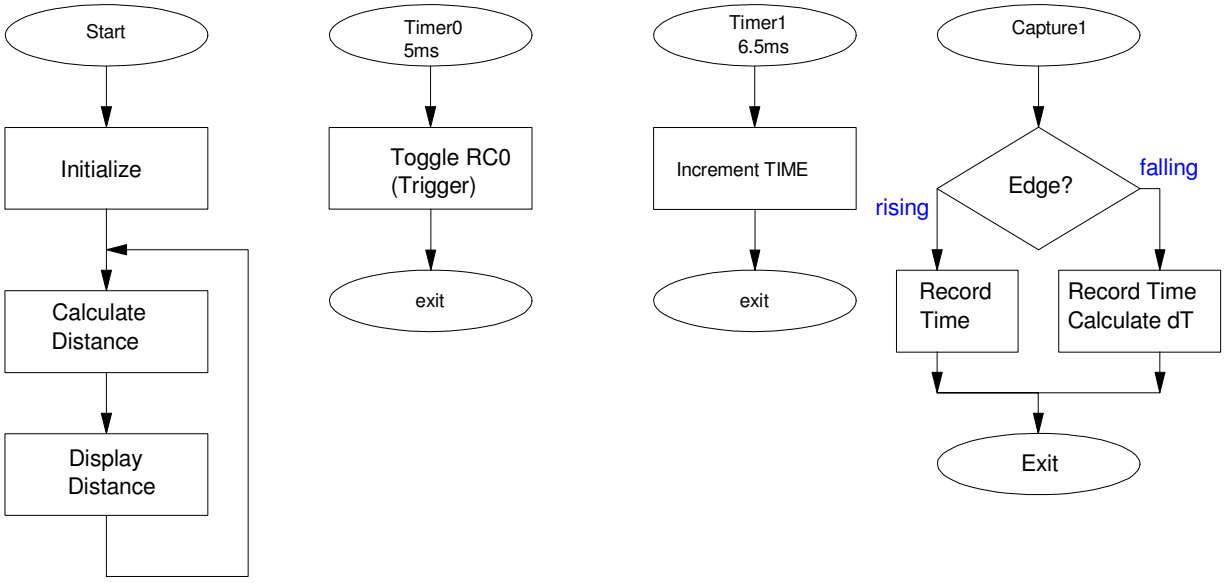
Timer1 Capture

Write a C program which uses Timer1 Capture along with an ultrasonic range sensor to measure the distance to an object that is 10cm to 30cm away from the sensor. The range measurements should be taken every 10ms (100Hz).

1) Give a flow chart for this program

With three interrupts, you have four flow charts

- One for each interrupt
- One for the main routine



2) C- Code

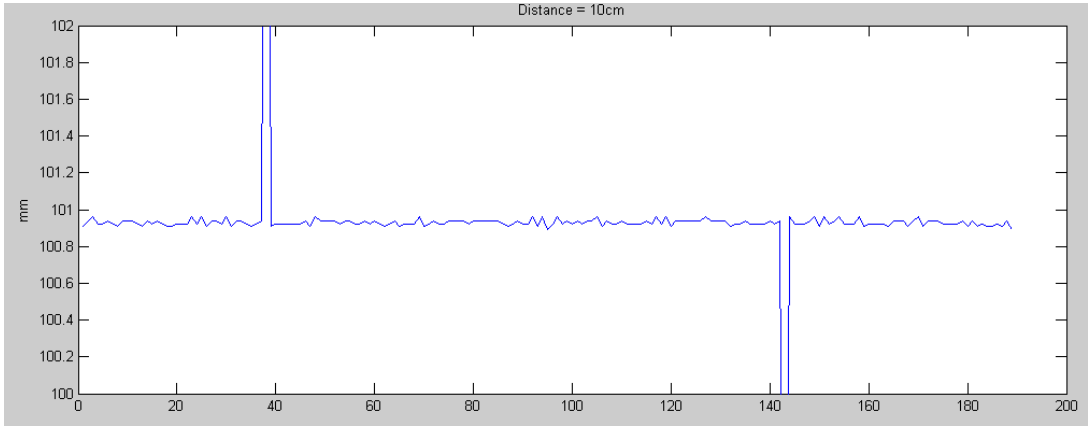
```
// Range.C
//
// Measure distance using an ultrasonic range sensor
//

#include <pic18.h>

// Global Variables
unsigned long int TIME, TIME0, TIME1, dT;

const unsigned char MSG0[21] = "Range.C             ";
const unsigned char MSG1[21] = "Ultrasonic Sensor   ";

// Interrupt Service Routine

void interrupt IntServe(void)
{
   if (TMR0IF) {  // 5ms, PS = 8
      TMR0 = -6250;
      RC0 = !RC0;
      TMR0IF = 0;
      }
   if (TMR1IF) {
      TIME = TIME + 0x10000;
      TMR1IF = 0;
      }
   if (CCP1IF) {
      if (CCP1CON == 0x05) {  // rising edge
         TIME0 = TIME + CCPR1;
         CCP1CON = 0x04;
         }
      else {
         TIME1 = TIME + CCPR1;
         dT = TIME1 - TIME0;
         CCP1CON = 0x05;
      }
      CCP1IF = 0;
      }
   }

// Subroutines
#include        "lcd_portd.c"


// Main Routine

void main(void)
{
   int mm;
   unsigned int i;

   TRISA = 0;
   TRISB = 0xFF;
   TRISC = 0x04;     // capture every rising edge
   TRISD = 0;
   ADCON1 = 0x0F;


   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(1000);
   LCD_Inst(1);
```

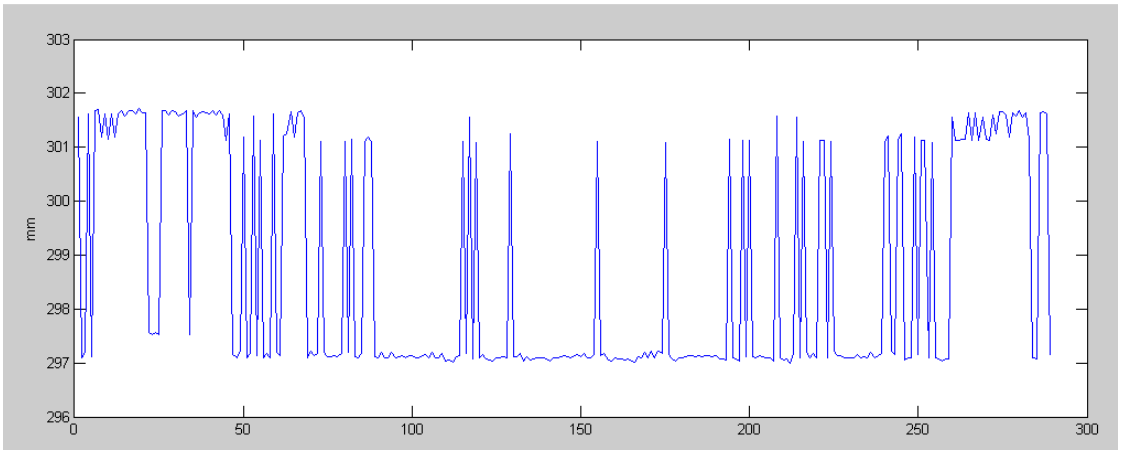
```
   TIME = 0;


// set up Timer0 for PS = 8
   T0CS = 0;
   T0CON = 0x82;
   TMR0ON = 1;
   TMR0IE = 1;
   TMR0IP = 1;
   PEIE = 1;
// set up Timer1 for PS = 8
   TMR1CS = 0;
   T1CON = 0x81;
   TMR1ON = 1;
   TMR1IE = 1;
   TMR1IP = 1;
   PEIE = 1;
// set up Capture1 for rising edges
   TRISC2 = 1;
   CCP1CON = 0x05;
   CCP1IE = 1;
   PEIE = 1;

// Initialize Serial Port to 9600 baud
   TRISC = TRISC | 0xC0;
   TXIE = 0;
   RCIE = 0;
   BRGH = 0;
   BRG16 = 1;
   SYNC = 0;
   SPBRG = 255;
   TXSTA = 0x22;
   RCSTA = 0x90;


// turn on all interrupts
   GIE = 1;

   while(1) {
      mm = dT * 1.715;   // units = 10um

      LCD_Move(0,0);  LCD_Out(dT, 10, 7);
      LCD_Move(1,0);  LCD_Out(mm, 6, 2);
      SCI_Out(mm, 6, 2);
      SCI_CRLF();

      }
   }
```

3) Record and plot 10 seconds (or more) of the distance measurements at

- 10cm
- 30cm

(note: use the serial port to record this data. Just include the plot of your data, not all 1000+ data points)





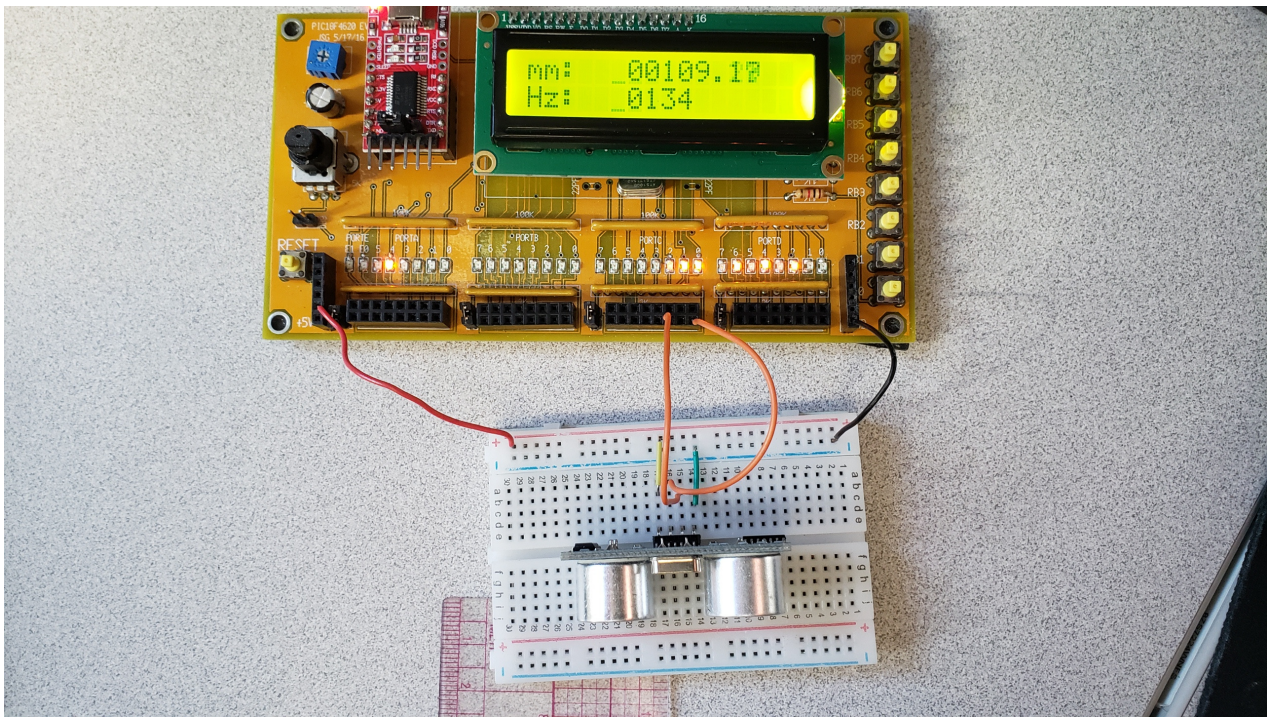
4) From your data, determine

- The mean and standard deviation at 10cm
- The mean and standard deviation at 30cm

Note: Keep your data. You'll need it for homework set #11

	mean (mm)	st dev (mm)	n (# data points)
R = 10cm	100.9293	0.4665	189
R = 30cm	298.6829	2.0730	289

5) Demo: in person or on video. Demo can be combined with problem #10.



Timer1 Compare

Use Timer1 Compare iterrupts to output

- 130.81Hz at 10cm (C3)
- 261.63Hz at 30cm (C4)

6) Determine the relationship

$$N_2 = f(N_1)$$

where

- N2 is the number of clocks between interrupts to generate notes C3 to C4, and
- N1 is the number of clocks for an object that is 10cm to 30cm away (problem 1-4)

At 10cm

$$N_1 = 5830.9$$

f = 130.81Hz

$$N_2 = \left(\frac{10{,}000{,}000}{2 \cdot 130.81 Hz}\right) = 38,223.37$$

At 30cm

$$N_1 = 17492.7$$

f = 261.73Hz

$$N_2 = \left(\frac{10{,}000{,}000}{2 \cdot 261.63 Hz}\right) = 19,111.69$$

Doing a linear cirve fit

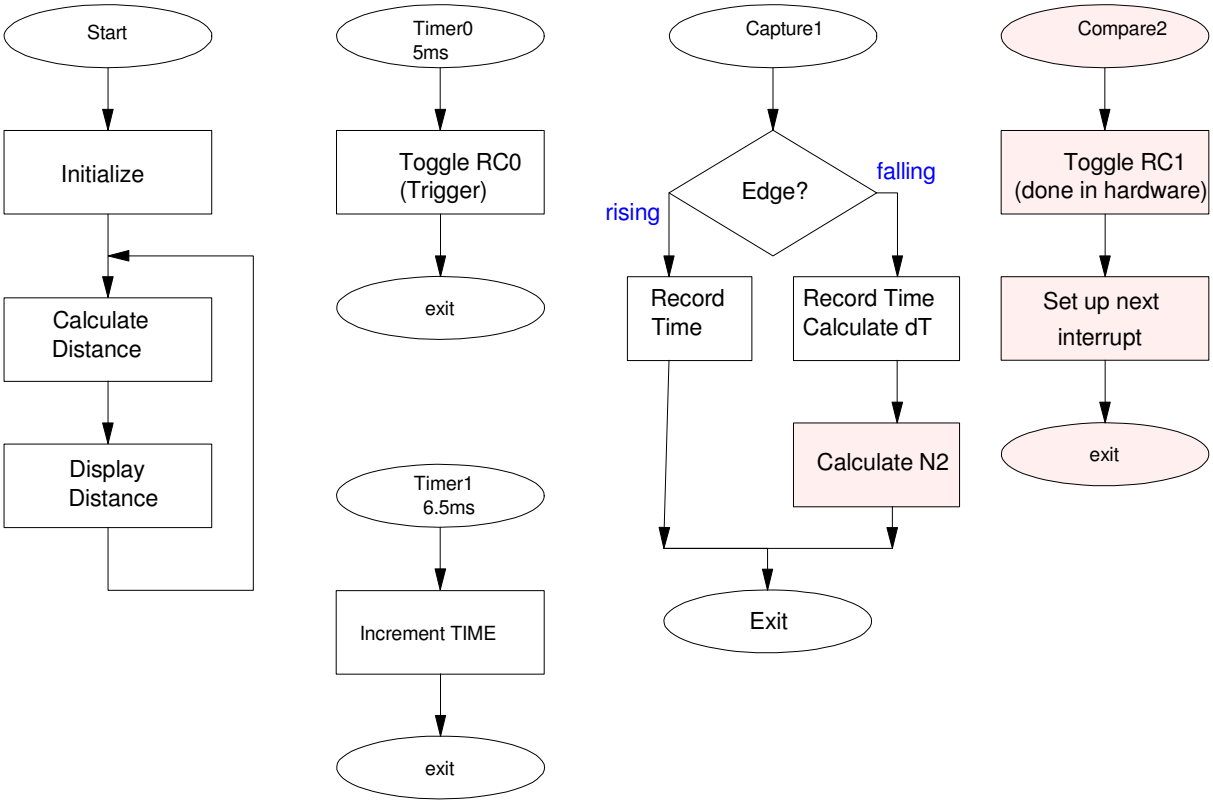
$$N_2 = 47779 - 1.6388 \cdot N_1$$

7) Give a flow chart for your resulting code

- Measure distance using Timer1 Capture1
- Generating frequency using Timer1 Compare2

Note:

- Previous code is used
- Additions are shown in shaded blocks



8) C - Code

```
// Timer1 Compare.C
//
// Output a square wave with a frequency of 349.23Hz
// on RC0 using Timer1 Compare interrupts

#include <pic18.h>

// Global Variables
unsigned long int TIME;
unsigned int TIME0, TIME1, N1, N2;

const unsigned char MSG0[21] = "mm:               ";
const unsigned char MSG1[21] = "Hz:               ";


// Interrupt Service Routine

void interrupt IntServe(void)
{
   if (TMR0IF) {  // 5ms, PS = 8
      TMR0 = -6250;
      RC0 = !RC0;
      TMR0IF = 0;
      }
   if (TMR1IF) {
      TIME = TIME + 0x10000;
      TMR1IF = 0;
      }
   if (CCP1IF) {
      if (CCP1CON == 0x05) {  // rising edge
         TIME0 = TIME + CCPR1;
         CCP1CON = 0x04;
         }
      else {
         TIME1 = TIME + CCPR1;
         N1 = TIME1 - TIME0;
         CCP1CON = 0x05;
      }
      CCP1IF = 0;
      }
   if (CCP2IF) {
      CCP2CON ^= 1;
      CCPR2 += N2;
      CCP2IF = 0;
      }
   }

// Subroutines
#include        "lcd_portd.c"



// Main Routine

void main(void)
{
   unsigned int mm;
   unsigned int i, X, Hz;

   TRISA = 0;
   TRISB = 0xFF;
   TRISC = 0x04;     // capture every rising edge
   TRISD = 0;
```

```
   ADCON1 = 0x0F;


   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(100);


   TIME = 0;

// 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 Capture1 for rising edges
   TRISC2 = 1;
   CCP1CON = 0x05;
   CCP1IE = 1;
   PEIE = 1;
// set up Compare2
   TRISC1 = 0;
   CCP2CON = 0x08;
   CCP2IE = 1;
   PEIE = 1;


// turn on all interrupts
   GIE = 1;

   while(1) {
      mm = N1 * 1.715;   // units = 10um
      N2 = 47779 - 1.68*N1;
      Hz = 5000000.0 / N2;
      LCD_Move(0,5);  LCD_Out(mm, 7, 2);
      LCD_Move(1,5);  LCD_Out(Hz, 4, 0);
      }
   }
```

9) Test & Validation: Collect data to verify

- Distance is measured correctly from 10cm to 30cm
- The frequency output is correct from 10cm to 30cm

Distance (mm)		Frequency (Hz)	
ruler	code	Freq Counter	Code
100	103.80	132.5	132
150	149.95	150.9	151
200	195.11	173.8	174
250	241.71	213.8	213
300	297.00	256.1	257

10) Demo