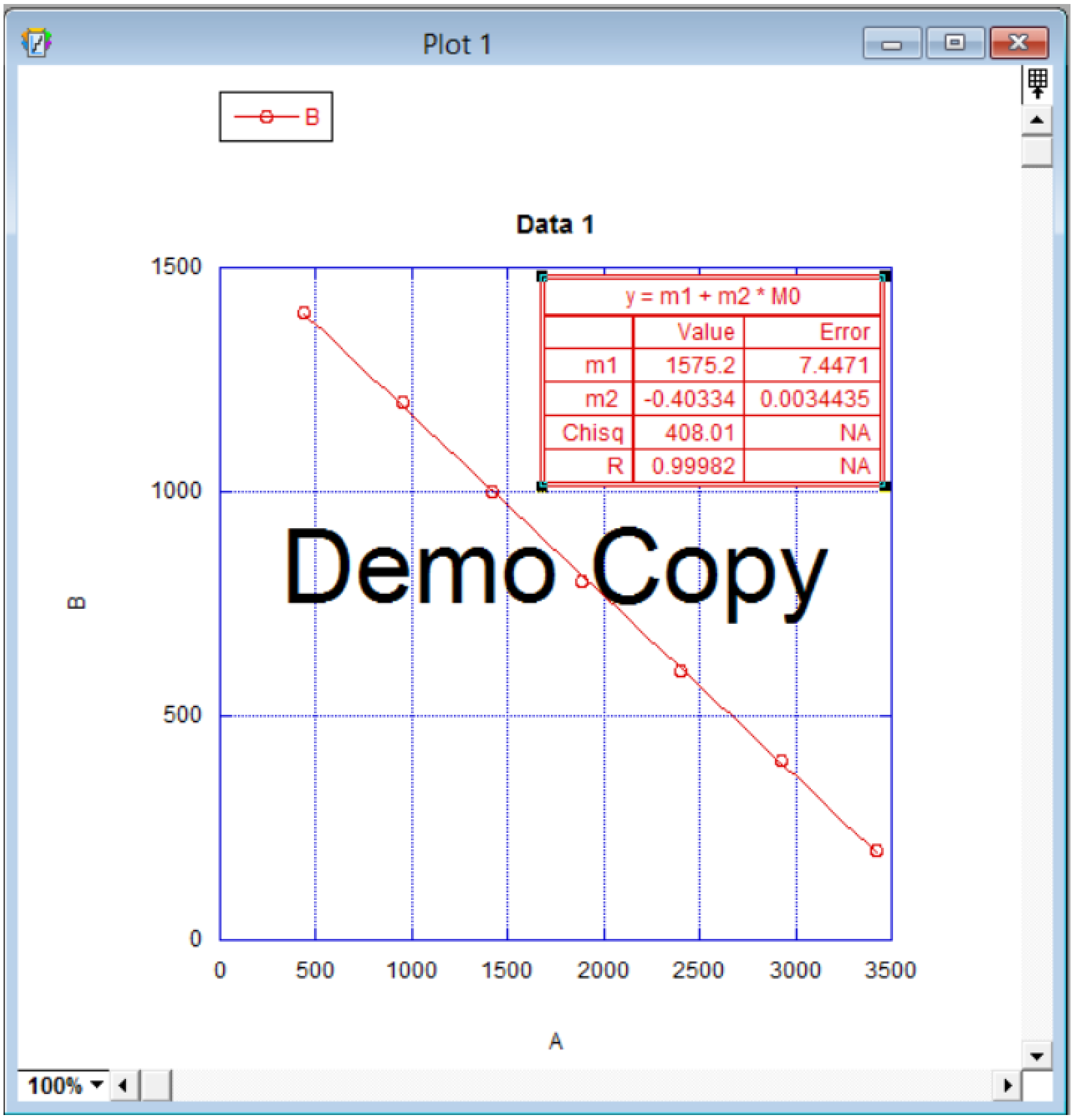
Lab 8 Deliverables

Calibration Data

Position - x	Analog Input	ADC Input	Correct Fixed P	Measured Fixed P
0	3681	3871	0	13
0.20	3140	3422	200	194
0.40	2814	2931	400	393
0.60	2237	2399	600	607
0.80	1747	1889	800	813
1.00	1212	1419	1000	1003
1.20	812	953	1200	1191
1.40	357	441	1400	1397



y = m1 + m2 * M0		
	Value	Error
m1	1575.2	7.4471
m2	-0.40334	0.0034435
Chisq	408.01	NA
R	0.99982	NA

Accuracy Data

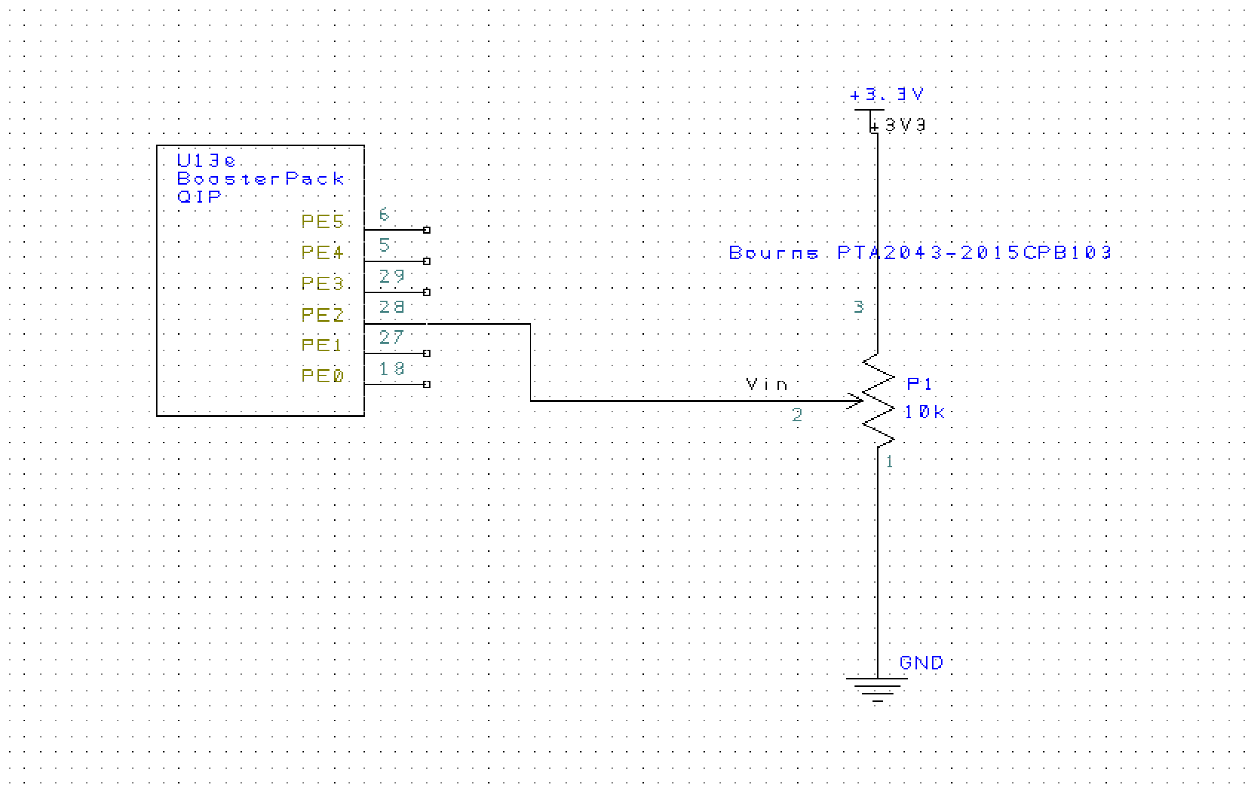
True position	Measured position	Error
0.1	0.109	-0.009
0.3	0.301	0.004
0.5	0.512	-0.012
0.7	0.721	-0.004
0.9	0.897	0.003

Data Set: 5 points

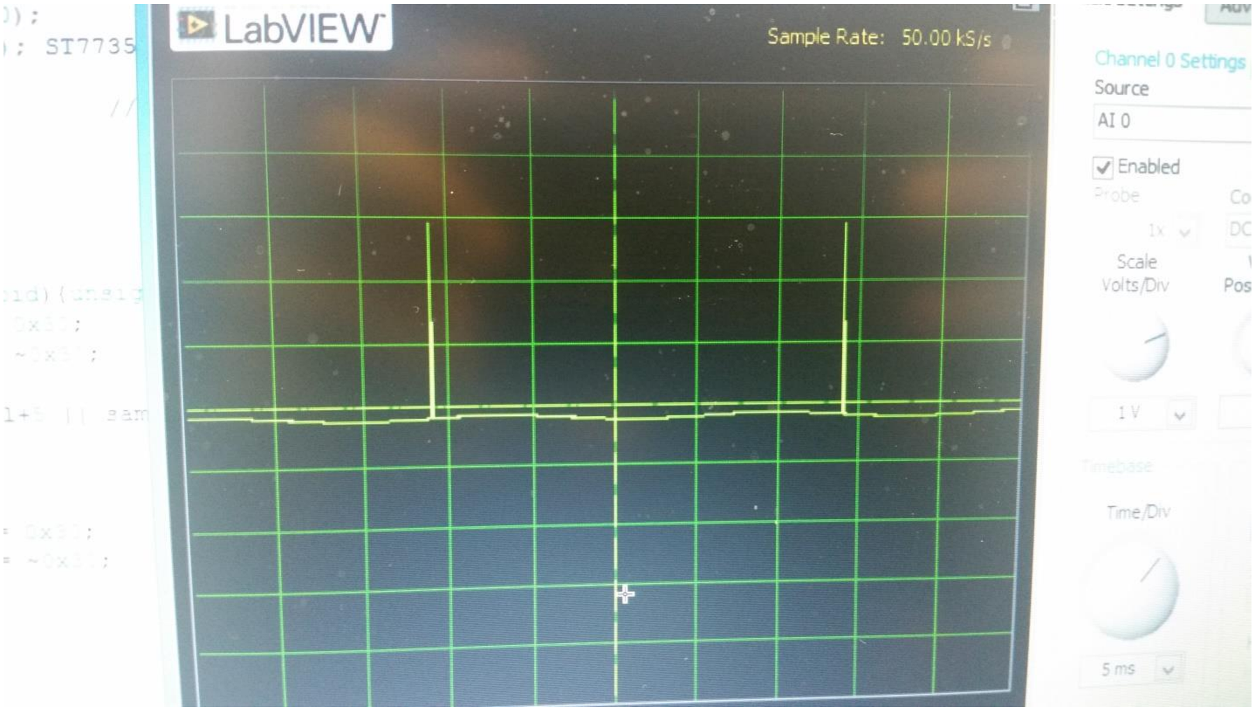
Maximum Error: 0.012

Average Error: 0.006

PCB Artist Diagram



ACD/LCD Execution Time



LabVIEW

Sample Rate: 50.00 kS/s

Channel 0 Settings

Source: AI 0

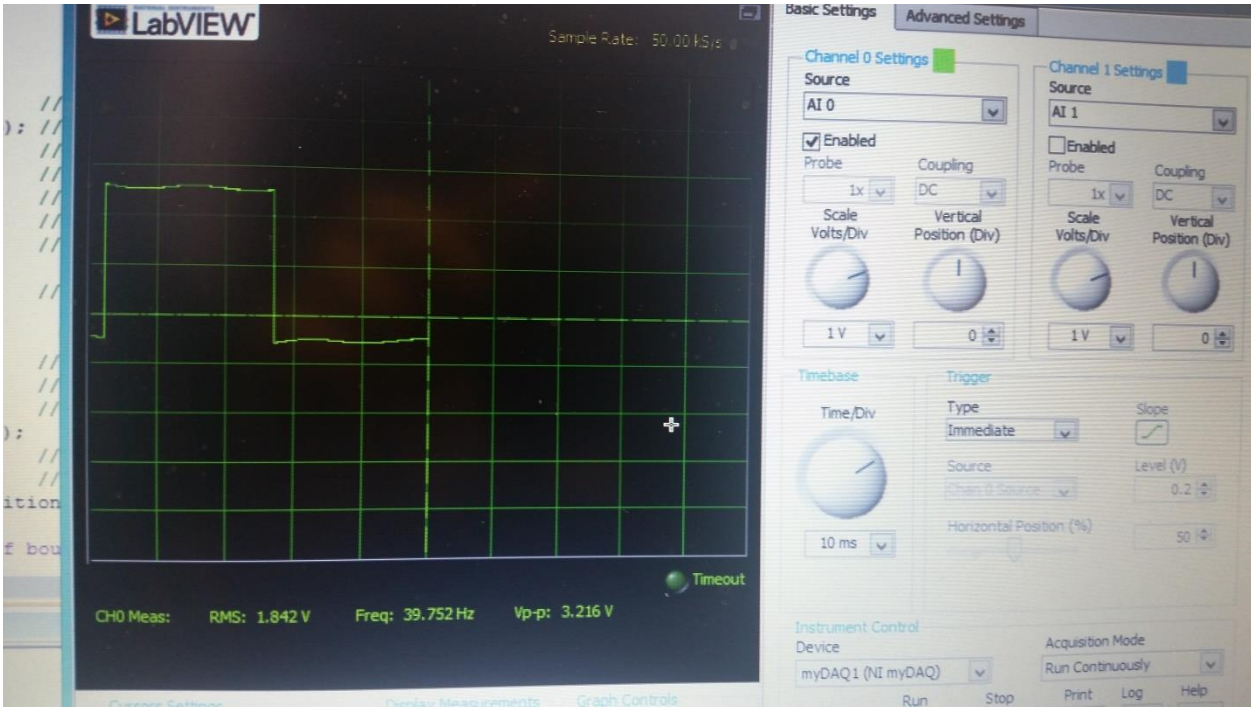
☑ Enabled

Scale Volts/Div: 1 V

Time/Div: 5 ms

ADC = 11 microseconds   Output = 1120 microseconds

Sampling Rate



LabVIEW

Sample Rate: 50.00 kS/s

Basic Settings | Advanced Settings

Channel 0 Settings

Source: AI 0

☑ Enabled

Probe: 1x   Coupling: DC

Scale Volts/Div: 1 V   Vertical Position (Div): 0

Channel 1 Settings

Source: AI 1

☐ Enabled

Probe: 1x   Coupling: DC

Scale Volts/Div: 1 V   Vertical Position (Div): 0

Timebase

Time/Div: 10 ms

Trigger

Type: Immediate

Source: Chan 0 Source

Slope

Level (V): 0.2

Horizontal Position (%): 50

Timeout

CH0 Meas: RMS: 1.842 V   Freq: 39.752 Hz   Vp-p: 3.216 V

Instrument Control

Device: myDAQ1 (NI myDAQ)

Acquisition Mode: Run Continuously

Final Code

```
// Lab8.c
// Runs on LM4F120 or TM4C123
// Student names: Eduardo Zueck Garces
// Last Modified: 4/14/2015

// Analog Input connected to PE2=ADC1
// displays on Sitronox ST7735
// PF3, PF2, PF1 are heartbeats

#include <stdint.h>

#include "ST7735.h"
#include "TExaS.h"
#include "ADC.h"
#include "print.h"
#include "tm4c123gh6pm.h"


//*****the first three main programs are for debugging *****
// main1 tests just the ADC and slide pot, use debugger to see data
// main2 adds the LCD to the ADC and slide pot, ADC data is on Nokia
// main3 adds your convert function, position data is no Nokia

void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts

#define PF1       (*((volatile uint32_t *)0x40025008))
#define PF2       (*((volatile uint32_t *)0x40025010))
#define PF3       (*((volatile uint32_t *)0x40025020))

int32_t Data;        // 12-bit ADC
int32_t Position;    // 32-bit fixed-point 0.001 cm
uint32_t ADCStatus;
uint32_t ADCMail;

// Initialize Port F so PF1, PF2 and PF3 are heartbeats
void PortF_Init(void){
	volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x035; //  We activate clock PE
  delay = SYSCTL_RCGC2_R;       //
  GPIO_PORTF_AMSEL_R &= ~0x0F;      // no analog
        GPIO_PORTF_PCTL_R &= ~0x00000FFF; // regular function
        GPIO_PORTF_DIR_R |= 0x0E;      // Make PF1-3 input
  GPIO_PORTF_AFSEL_R &= ~0x0F;   // disable alt on PE2-0
  GPIO_PORTF_DEN_R |= 0x0E;      // Enable Digital IO
        ADCMail=0;
        ADCStatus=0;
}

//Initialize our Heartbeat registers
void HeartBeat(void){
        unsigned long volatile delayE;
        GPIO_PORTC_AMSEL_R &= ~0x30;      // no analog
  GPIO_PORTC_DIR_R |= 0x30;      // make PC4-5 out
  GPIO_PORTC_AFSEL_R &= ~0x30;   // disable alt on PC4-5
  GPIO_PORTC_DEN_R |= 0x30;      // enable digitalon PE4-5
}

//Initialize our Systick timer with 40HZ
void Systick_Init(void){
        NVIC_ST_CTRL_R = 0;
        NVIC_ST_RELOAD_R = 2000000-1;
        NVIC_ST_CURRENT_R = 0;
        NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x20000000;
        NVIC_ST_CTRL_R = 0x0007;
}

//Function to convert our data
uint32_t Convert(uint32_t input){
        signed long output;
        output = 1575-(0.40344*input);
  return output;
}
//Our final main program
int main(void){
  TExaS_Init();                              //Bus clock is 80 MHz
  ST7735_InitR(INITR_REDTAB); //Initialize the screen
  PortF_Init();                                                     //Initialize PortF
  ADC_Init();                                //Turn on ADC, set channel to 1
        HeartBeat();                                                        //Initialize PortC for
heartbeat
        Systick_Init();                                               //Initialize our SysTick timer
```

```
 EnableInterrupts();                                  //We enable interrupts and run with it
 while(1){
              while(ADCStatus==0){                    //Wait for new data
        }
              DisableInterrupts();
              PF2 ^= 0x04;                             //Heartbeat
  Data = ADCMail;                                     //Get Data from mail
  PF3 = 0x08;                                   //Profile Convert
  Position = Convert(Data);
  PF3 = 0;                              //End of Convert Profile
  PF1 = 0x02;                                   //Profile LCD
              if(Position>1500 || Position<0){
              ST7735_SetCursor(0,0);
  ST7735_OutString("Out of bounds!");
  ST7735_SetCursor(6,0);
              }
              else{
  ST7735_SetCursor(0,0);
  LCD_OutDec(Position); ST7735_OutString("    ");
  ST7735_SetCursor(6,0);
  LCD_OutFix(Position); ST7735_OutString(" cm");
              }
  PF1 = 0;                              //End of LCD Profile
              ADCStatus=0;
              EnableInterrupts();
 }
}

void SysTick_Handler(void){unsigned long sampling;
       GPIO_PORTC_DATA_R ^= 0x30;     //Toggle HB
       sampling = ADC_In();                                           //Get our new sample
       if(sampling > ADCMail+5 || sampling < ADCMail-5){                            //Check if new
one for MB
              ADCStatus = 1;                                                        //Set mailbox
       }
       ADCMail = sampling;                                                   //MAAAAIL
}
```