```
// Lab8.c
// Runs on LM4F120 or TM4C123
// Student names: change this to your names or look very silly
// Last modification date: change this to the last modification date or look very silly
// Last Modified: 4/5/2016

// 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"
uint32_t ADCMail;
uint32_t ADCStatus;
//*****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))
```

```
// Initialize Port F so PF1, PF2 and PF3 are heartbeats
uint32_t input;
void PortF_Init(void){
        SYSCTL_RCGCGPIO_R |= 0x20;
        uint32_t cycle=0;
        cycle=1;
        cycle=2;
        cycle=3;
        GPIO_PORTF_DEN_R |= 0X0E;
        GPIO_PORTF_DIR_R |= 0X0E;

}
uint32_t Data;        // 12-bit ADC
uint32_t Position;    // 32-bit fixed-point 0.001 cm
int main1(void){      // single step this program and look at Data
  TExaS_Init();       // Bus clock is 80 MHz
  ADC_Init();         // turn on ADC, set channel to 1
  while(1){
    Data = ADC_In();  // sample 12-bit channel 1
  }
}

int main2(void){
  TExaS_Init();       // Bus clock is 80 MHz
  ADC_Init();         // turn on ADC, set channel to 1
  ST7735_InitR(INITR_REDTAB);
  PortF_Init();
  while(1){           // use scope to measure execution time for ADC_In and LCD_OutDec
    PF2 = 0x04;       // Profile ADC
    Data = ADC_In();  // sample 12-bit channel 1
```

```
    PF2 = 0x00;       // end of ADC Profile
    ST7735_SetCursor(0,0);
    PF1 = 0x02;       // Profile LCD
    LCD_OutDec(Data);
    ST7735_OutString("    ");  // these spaces are used to coverup characters from last output
    PF1 = 0;          // end of LCD Profile
  }
}

uint32_t Convert(uint32_t number){
  uint32_t position;
          position = (24933*number)/10000-(84665/10);
          return position;
}

void SysTick_Init(void){
          NVIC_ST_CTRL_R = 0;         // disable SysTick during setup
  NVIC_ST_RELOAD_R = 2000000-1;// reload value
  NVIC_ST_CURRENT_R = 0;      // any write to current clears it
          NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF) | 0x40000000; //sets
priority level 2
          NVIC_ST_CTRL_R = 0x07; //enables interrupts
}

void SysTick_Handler(void){
          GPIO_PORTF_DATA_R ^= 0x0E;
          GPIO_PORTF_DATA_R ^= 0x0E;
          ADCMail = ADC_In();
          ADCStatus =1;
          GPIO_PORTF_DATA_R ^= 0x0E;
```

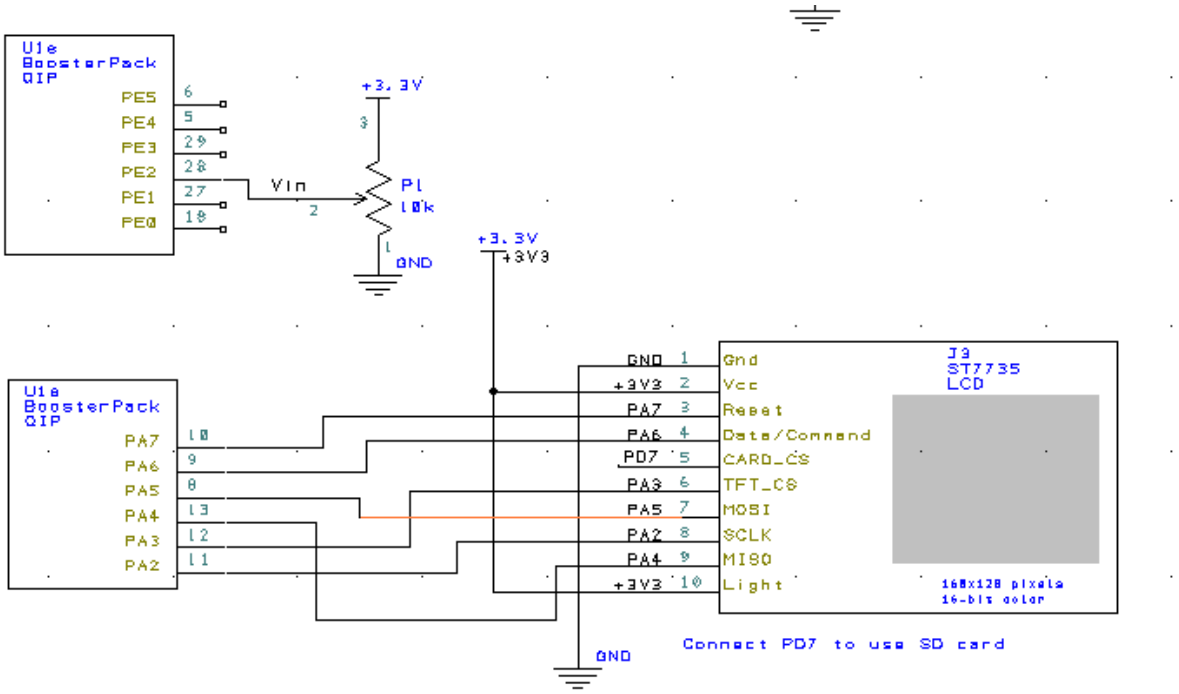
```
}



int main3(void){
  TExaS_Init();         // Bus clock is 80 MHz
  ST7735_InitR(INITR_REDTAB);
  PortF_Init();
  ADC_Init();         // turn on ADC, set channel to 1
  while(1){
    PF2 ^= 0x04;      // Heartbeat
    Data = ADC_In();  // sample 12-bit channel 1
    PF3 = 0x08;       // Profile Convert
    Position = Convert(Data);
    PF3 = 0;          // end of Convert Profile
    PF1 = 0x02;       // Profile LCD
    ST7735_SetCursor(0,0);
    LCD_OutDec(Data); ST7735_OutString("    ");
    ST7735_SetCursor(6,0);
    LCD_OutFix(Position);
    PF1 = 0;          // end of LCD Profile
  }
}
int main(void){
  TExaS_Init();
          PortF_Init();
          ADC_Init();
          ST7735_InitR(INITR_REDTAB);
          EnableInterrupts();
```

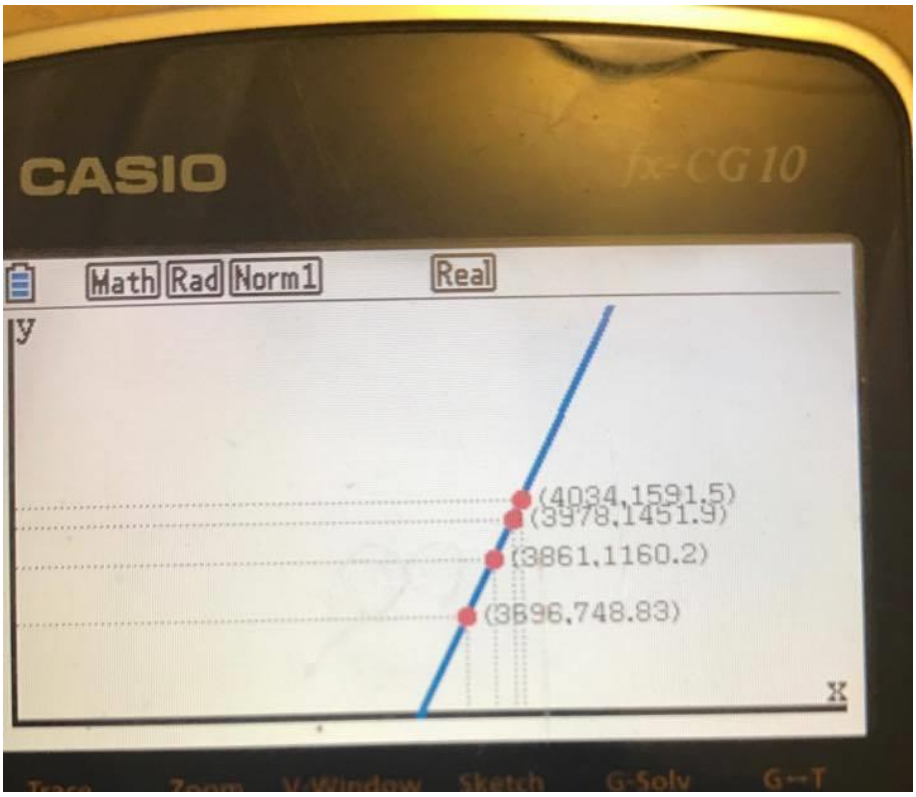
```
        SysTick_Init();
        while(1){
                uint32_t new_input;

                if(ADCStatus ==1){

                        ADCStatus=0;
                        new_input=Convert(ADCMail);
                        ST7735_SetCursor(0,0);
                        LCD_OutFix(new_input);
                        ST7735_OutString("    ");
                }
        }
}
```

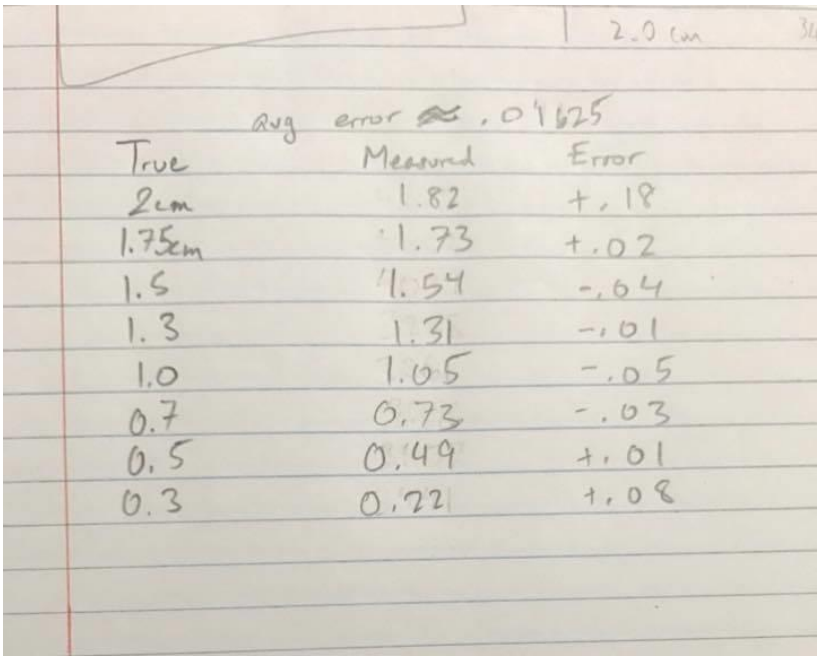
```
/ ADC.c
// Runs on LM4F120/TM4C123
// Provide functions that initialize ADC0
// Last Modified: 3/6/2015
// Student names: change this to your names or look very silly
// Last modification date: change this to the last modification date or look very silly

#include <stdint.h>
#include "tm4c123gh6pm.h"

// ADC initialization function
// Input: none
```

```
// Output: none
void ADC_Init(void){
        SYSCTL_RCGCGPIO_R |= 0x10;
        while((SYSCTL_PRGPIO_R&0x10) == 0){};
        GPIO_PORTE_DIR_R &= ~0x04;
        GPIO_PORTE_AFSEL_R |= 0x04;
        GPIO_PORTE_DEN_R &= ~0x04;
        GPIO_PORTE_AMSEL_R |= 0x04;
        SYSCTL_RCGCADC_R |= 0x01;
        uint32_t cycle=SYSCTL_RCGCADC_R;
        cycle=SYSCTL_RCGCADC_R;
        cycle=SYSCTL_RCGCADC_R;
        cycle=SYSCTL_RCGCADC_R;
        ADC0_PC_R = 0x01;
        ADC0_SSPRI_R = 0x0123;
        ADC0_ACTSS_R &= ~0x0008;
        ADC0_EMUX_R &= ~0xF000;
        ADC0_SSMUX3_R = (ADC0_SSMUX3_R & 0xFFFFFFF0)+1;
        ADC0_SSCTL3_R = 0x0006;
        ADC0_IM_R &= ~0x0008;
        ADC0_ACTSS_R |= 0x0008;



}

//------------ADC_In------------
// Busy-wait Analog to digital conversion
// Input: none
// Output: 12-bit result of ADC conversion
uint32_t ADC_In(void){
```

```
        uint32_t data;
        ADC0_PSSI_R = 0x0008;
  while((ADC0_RIS_R&0x08)==0){};
  data = ADC0_SSFIFO3_R&0xFFF;
  ADC0_ISC_R = 0x0008;
  return data;


}
```


U1e
BoosterPack
QIP
PE5
PE4
PE3
PE2
PE1
PE0
6
5
29
28
27
18
Vin
2
+3.3V
3
P1
10k
1
GND
+3.3V
+3V3
U1a
BoosterPack
QIP
PA7
PA6
PA5
PA4
PA3
PA2
10
9
8
13
12
11
GND 1
+3V3 2
PA7 3
PA6 4
PD7 5
PA3 6
PA5 7
PA2 8
PA4 9
+3V3 10
J3
ST7735
LCD
Gnd
Vcc
Reset
Data/Command
CARD_CS
TFT_CS
MOSI
SCLK
MISO
Light
168x128 pixels
16-bit color
GND
Connect PD7 to use SD card

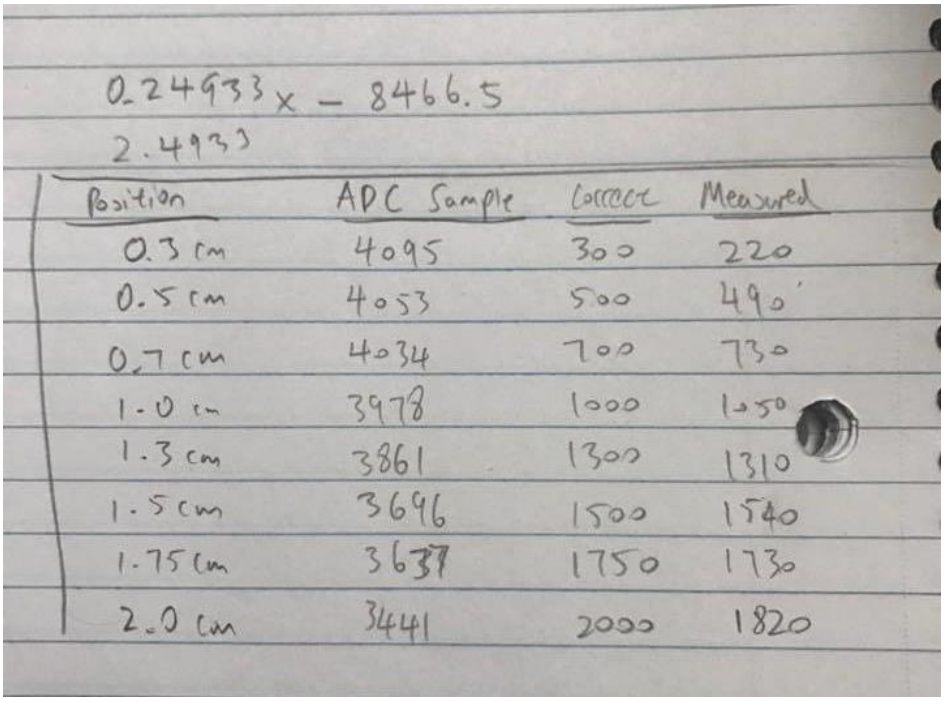

CASIO
fx-CG10
Math
Rad
Norm1
Real
y
(4034,1591.5)
(3978,1451.9)
(3861,1160.2)
(3696,748.83)
x
Trace
Zoom
V-Window
Sketch
G-Solv
G↔T



2.0 cm

Avg error ≈ .01625

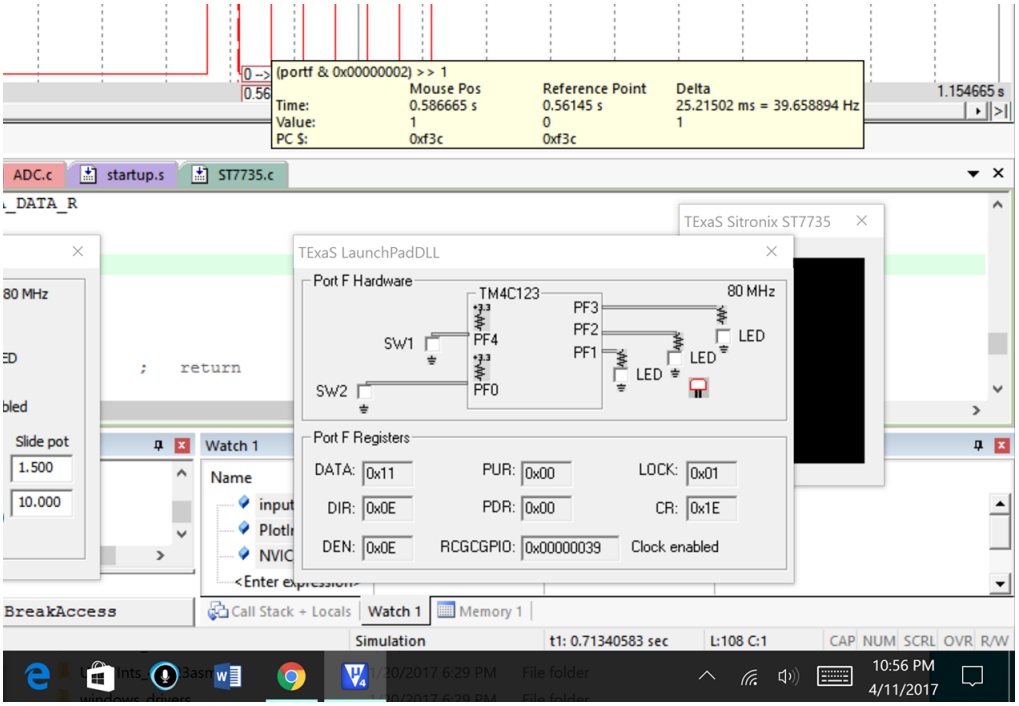
True	Measured	Error
2cm	1.82	+.18
1.75cm	1.73	+.02
1.5	1.54	-.04
1.3	1.31	-.01
1.0	1.05	-.05
0.7	0.73	-.03
0.5	0.49	+.01
0.3	0.22	+.08



0.24933x − 8466.5

2.4933

Position	ADC Sample	Correct	Measured
0.3 cm	4095	300	220
0.5 cm	4053	500	490
0.7 cm	4034	700	730
1.0 cm	3978	1000	1050
1.3 cm	3861	1300	1310
1.5 cm	3696	1500	1540
1.75 cm	3637	1750	1730
2.0 cm	3441	2000	1820


(portf & 0x00000002) >> 1
Mouse Pos
Reference Point
Delta
Time:
0.586665 s
0.56145 s
25.21502 ms = 39.658894 Hz
Value:
1
0
1
PC $:
0xf3c
0xf3c
1.154665 s
ADC.c
startup.s
ST7735.c
_DATA_R
; return
TExaS Sitronix ST7735
TExaS LaunchPadDLL
Port F Hardware
TM4C123
80 MHz
PF3
PF2
PF1
PF4
PF0
SW1
SW2
LED
LED
LED
Port F Registers
DATA: 0x11
PUR: 0x00
LOCK: 0x01
DIR: 0x0E
PDR: 0x00
CR: 0x1E
DEN: 0x0E
RCGCGPIO: 0x00000039
Clock enabled
80 MHz
Slide pot
1.500
10.000
Watch 1
Name
input
Plotl
NVIC
<Enter expression>
BreakAccess
Call Stack + Locals
Watch 1
Memory 1
Simulation
t1: 0.71340583 sec
L:108 C:1
CAP NUM SCRL OVR R/W
10:56 PM
4/11/2017