```
/*
 Moses Peace McCabe

 this project creates a position measurment software that outputs the result on LCD
 The software samples analog input signal and use calibration to output the position
 Runs on LM4F120 or TM4C123
 Use the SysTick timer to request interrupts at a particular period.

*/


#include "Nokia5110.h"
#include "pll.h"
#include "ADC.h"
#include "tm4c123gh6pm.h"


void DisableInterrupts(void);                   // Disable interrupts
void EnableInterrupts(void);                  // Enable interrupts
long StartCritical (void);              // previous I bit, disable interrupts
void EndCritical(long sr);              // restore I bit to previous value
void WaitForInterrupt(void);            // low power mode
unsigned long Convert(unsigned long );  // conversion function
void PORTF_Init(void);                  // Post F initialization
void SysTick_Init(void);                // SysTick initialization


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



unsigned short Data;      // 12-bit ADC
unsigned long Position;  // 32-bit fixed-point 0.001 cm
unsigned long data, data1, data2, temp, ADCMail, ADCstatus = 0;



//int main(void){      // single step this program and look at Data
//  PLL_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
//  }
//}


//  PART 3 #C
int main(void){
  PLL_Init();         // Bus clock is 80 MHz
  ADC_Init();         // turn on ADC, set channel to 1
  SysTick_Init();     // call systick initialization

  Nokia5110_Init();   // call nokia initialization

  PORTF_Init();                 //initializes port PF2 for debugging
```

```
  EnableInterrupts();   // enable all interrupts

  while(1){           // use scope to measure execution time for ADC_In and LCD_OutDec
                        if(ADCstatus == 1)
                        {
                                ADCstatus = 0;         // set status to zero indicating data has been read
                                Data = ADCMail;        // pass scam value to Data
                                data = Convert(Data);  // get calculated value
                                data1 = data/1000;     // get 1's place digit
                                data2 = data%1000;     // get remainder
                          Nokia5110_Clear();       // clear cursor
                                Nokia5110_OutUDec(data1);  // display digit
                                Nokia5110_OutChar('.');    // display the decimal point
                                Nokia5110_OutUDec(data2);  // display remainder.
                                Nokia5110_OutString("cm"); // display cm
                        }
                        else{} // else do nothing
}
}

unsigned long Convert(unsigned long input){
        Position = ( (0.4196*input)*(256) + (20.515)*(256) )/256;  // converted ADC data into fixed-point
  return Position;
}



void SysTick_Init(void)
{
        NVIC_ST_CTRL_R = 0x00;                    // disable SysTick during setup
  NVIC_ST_RELOAD_R = (2000000 -1);       // reload value for 500us
  NVIC_ST_CURRENT_R = 0;                  // any write to current clears it
        NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000; // priority 2
        NVIC_ST_CTRL_R = 0x00000007;           // enable with core clock and interrupts
}


void SysTick_Handler(void)
{
        PF2 ^= 0x04;            // set PF2 to high
        ADCMail = ADC_In();     // Read scan data into ADCMAIL
        ADCstatus = 1;          // ADCstatus to one indicating data has been read
        PF2 ^= 0x00;           // set PF2 to low
}
```

```
Error using dbstatus
Error: File: /Users/mosesmccabe/Documents/Embedded System/embedded sys/ADC/main_c.m Line: 1 Column: 1
Invalid use of operator.
```

Published with MATLAB® R2019b