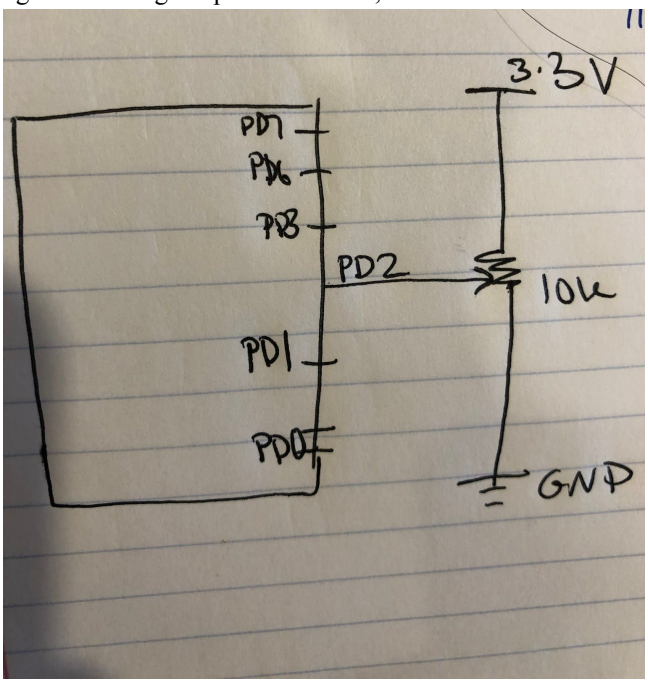
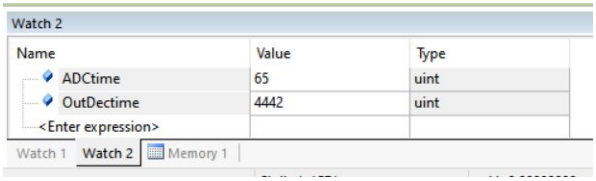
1. Circuit diagram showing the position sensor, hand-drawn or PCB Artist, like Figure 8.1 (part a),
   a. 

2. Time measurements ~~and a photo~~ showing the ADC/LCD execution time profile (part d)
   a. 

3. Calibration data, like the first three columns of Table 8.1 (part d)
   a.

Position	Analog Input	ADC Sample
0.10	0.07	82
0.50	0.79	998
0.90	1.64	2050
1.10	2.02	2522
1.50	2.87	3589

   b.
4. Final version of distance meter with SysTick, ADC, convert, and main (your code for parts c, e, f, g and h)
   a.

```
52 void SysTick_Init(void){
53         NVIC_ST_CTRL_R = 0;
54         NVIC_ST_CTRL_R = 0x07;
55         NVIC_ST_RELOAD_R = 8000000;
56         NVIC_ST_CURRENT_R = 0;
57
58 }
```

b.

```
void ADC_Init(void){
        volatile unsigned long delay;
        SYSCTL_RCGCGPIO_R |= 0x08;  // 1) activate clock for Port D
        delay = SYSCTL_RCGCGPIO_R;          //    allow time for clock to stabilize
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        delay = SYSCTL_RCGCGPIO_R;
        GPIO_PORTD_DIR_R &= ~0x04;       // 2) make PD2 input
        GPIO_PORTD_AFSEL_R |= 0x04;      // 3) enable alternate function on PD2
        GPIO_PORTD_DEN_R &= ~0x04;       // 4) disable digital I/O on PD2
        GPIO_PORTD_AMSEL_R |= 0x04;      // 5) enable analog function on PD2
        SYSCTL_RCGCADC_R |= 0x01;   // 6) activate ADC0
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
        delay = SYSCTL_RCGCADC_R;
  ADC0_PC_R = 0x03;              //edit   // 7) configure for 125K ?
  ADC0_SSPRI_R = 0x0123;           // 8) Sequencer 3 is highest priority
  ADC0_ACTSS_R &= ~0x0008;         // 9) disable sample sequencer 3
  ADC0_EMUX_R &= ~0xF000;          // 10) seq3 is software trigger
  ADC0_SSMUX3_R=(ADC1_SSMUX3_R & ~0xF)+5; // channel Ain5 (PD2)
  ADC0_SSCTL3_R = 0x0006; // 12) no TS0 D0, yes IE0 END0
  ADC0_IM_R &= ~0x0008;
  ADC0_ACTSS_R |= 0x0008;
}
```

c.

```
uint32_t ADC_In(void){
        //uint32_t result =0;
        unsigned long result; //edit
        ADC0_PSSI_R = 0x008; //start sequencer 3
        while ((ADC0_RIS_R & 0x08)==0) {}; //busy-wait
        result = ADC0_SSFIFO3_R & 0xFFF; //read data from FIFO buffer
        ADC0_ISC_R = 0x0008; //clear flag in RIS
        return result;
}
```

d.

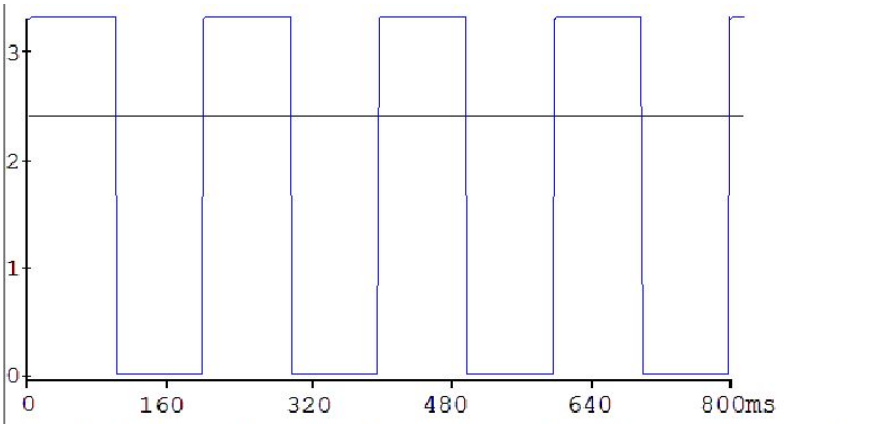
```
// your function to convert ADC sample to distance (0.01cm)
uint32_t Convert(uint32_t data){
  return  162*data/4096+8; // replace this line with your Lab 8 solution
  //data = 162*data/4096+8
}
```

e.

```
int main(void){
  TExaS_Init();
  ST7735_InitR (INITR_REDTAB);// your Lab 8
  PortF_Init();
  ADC_Init();
  SysTick_Init ();
  EnableInterrupts();
  while(1){
      if (ADCstatus == 1){ //check if one or zero
        uint32_t x = ADCmail;
        ADCstatus = 0;
        x = Convert(x);
        ST7735_SetCursor(1,1);
        LCD_OutFix(x);
        ST7735_OutString("cm ");

    }
  }
}
```

5. A photo or screenshot verifying the sampling rate is 10 Hz, like Figure 8.8 (part h)



a.

6. Accuracy data and accuracy calculation, Table 8.2 (part i)

a.

True Position	Measured Position	True-Measured
0.10	0.13	-0.03
0.50	0.52	-0.02
0.90	0.90	0.00
1.10	1.13	-0.03
1.50	1.51	-0.01