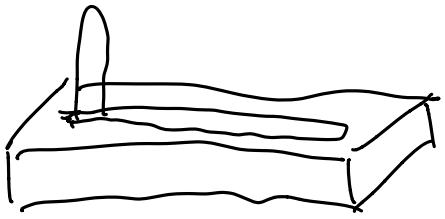
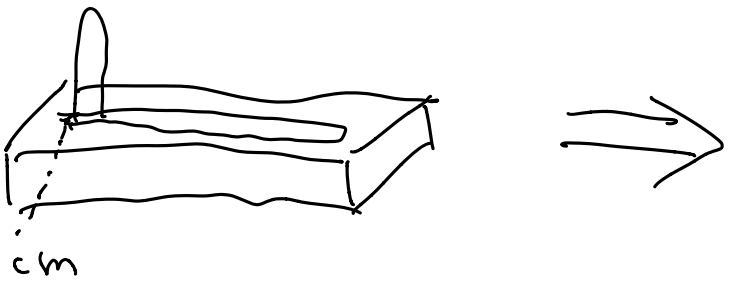
- Overview

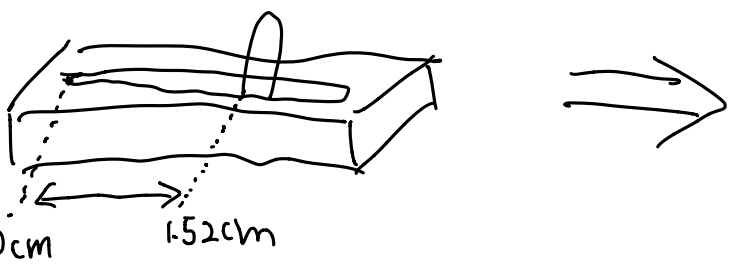
"Slide Pot" → potentiometer
- change resistance





OLED

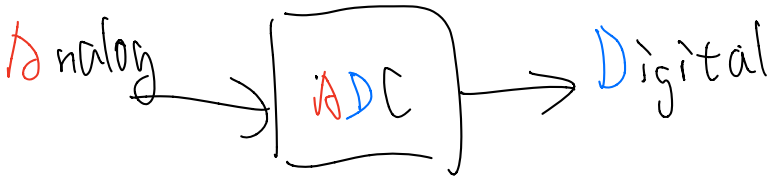
0.00



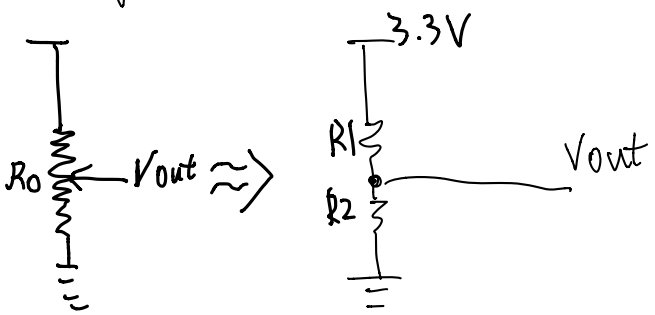
OLED

1.52

– what is ADC?



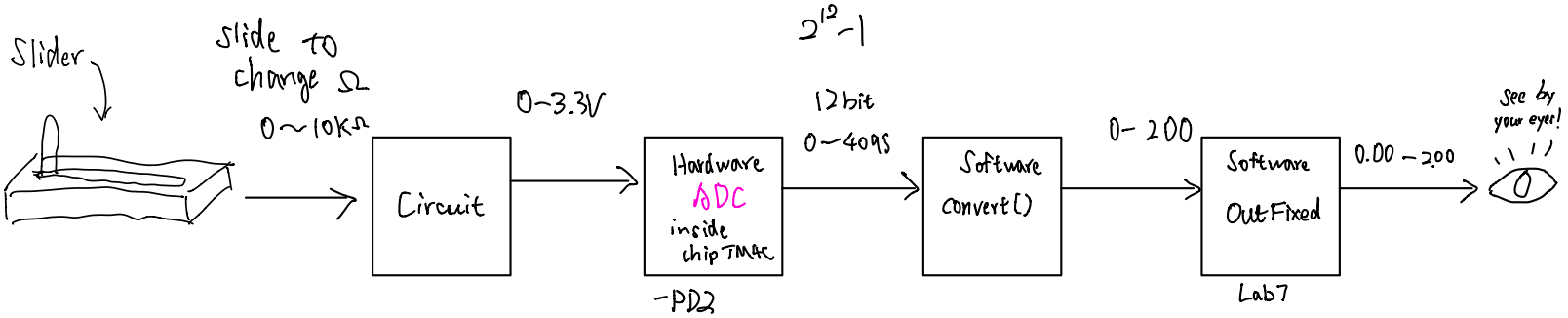
– potentiometer?



R1 & R2 combine always has 10kΩ

$$Vout = \frac{3.3\,R2}{(R1+R2)} = \frac{3.3R2}{10k\Omega}$$

so Vout is approximately linear to R2.



- Steps:

1) baild the circuit (part a)

2) understand software organization (part b)

3) Setup ADC (part c)

---

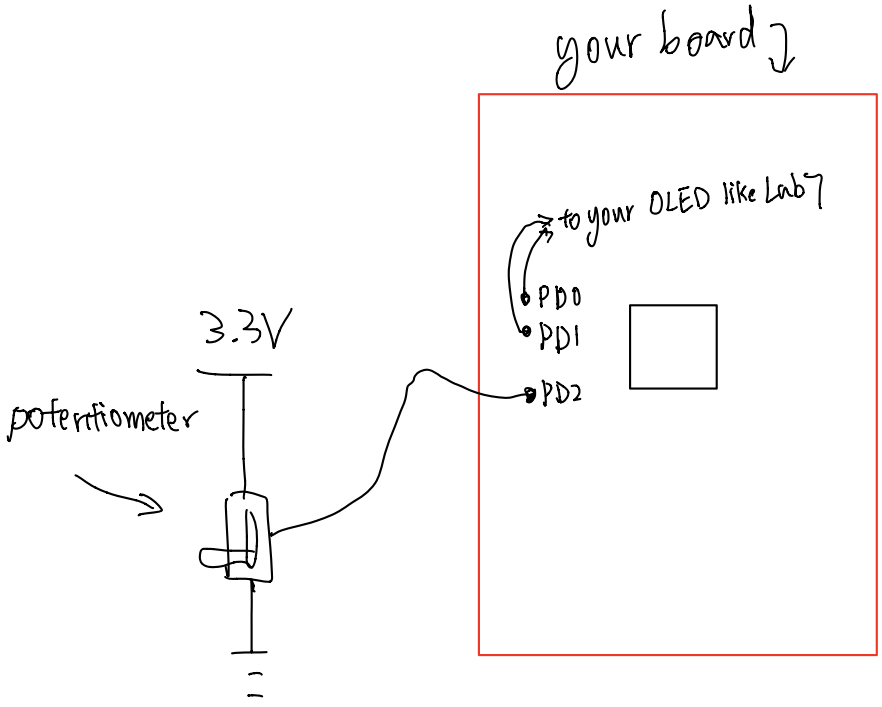
4) convert ADC value to actual length (part c & d)

you are 90% done at this point!

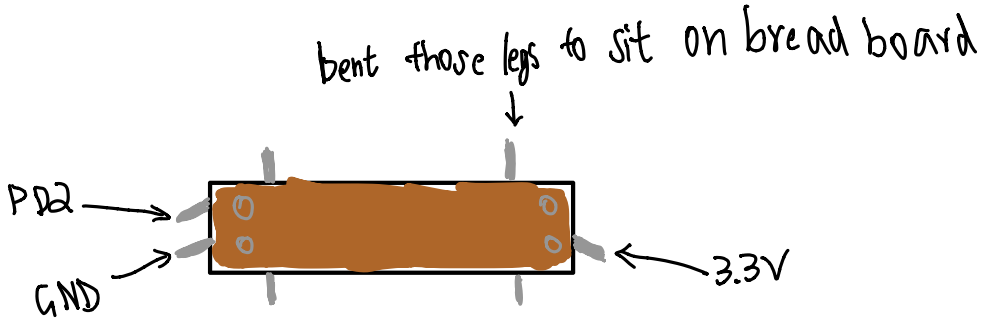
5) Use systick interrupt instead (part f & g & h)

6) Check if yours solution is accurate. (part i)

1) baild the circuit (part a)

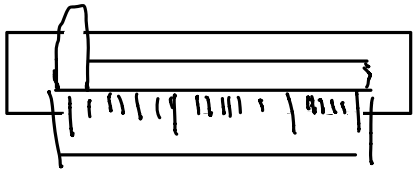


bottom view of slide pot:



*tips: easy to connect with female Jumper wire.

-add a ruler:



2) software

0) All the initialization i.e. ADC_init ; OLED_init ...

1) get your ADC value

i.e. int Data = ADC_In()

2) convert raw data to Fixed point in 0.1 mm

i.e. int position = Convert(Data);

3) OutFixed(Position);

4) go back to step 1 & repeat!

In a nutshell...

a) copy all OLED function you wrote from Lab7

b) you need to write 1. ADC_init() — both in ADC.c
2. ADC_In() — both in ADC.c
3. Convert() — directly in main.c

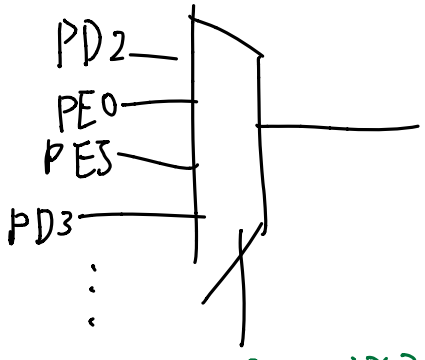
3) Set up ADC

Understand the hardware

- there are 2 ADC in the chip

✓ we use ADC0 these time

- where is the input?



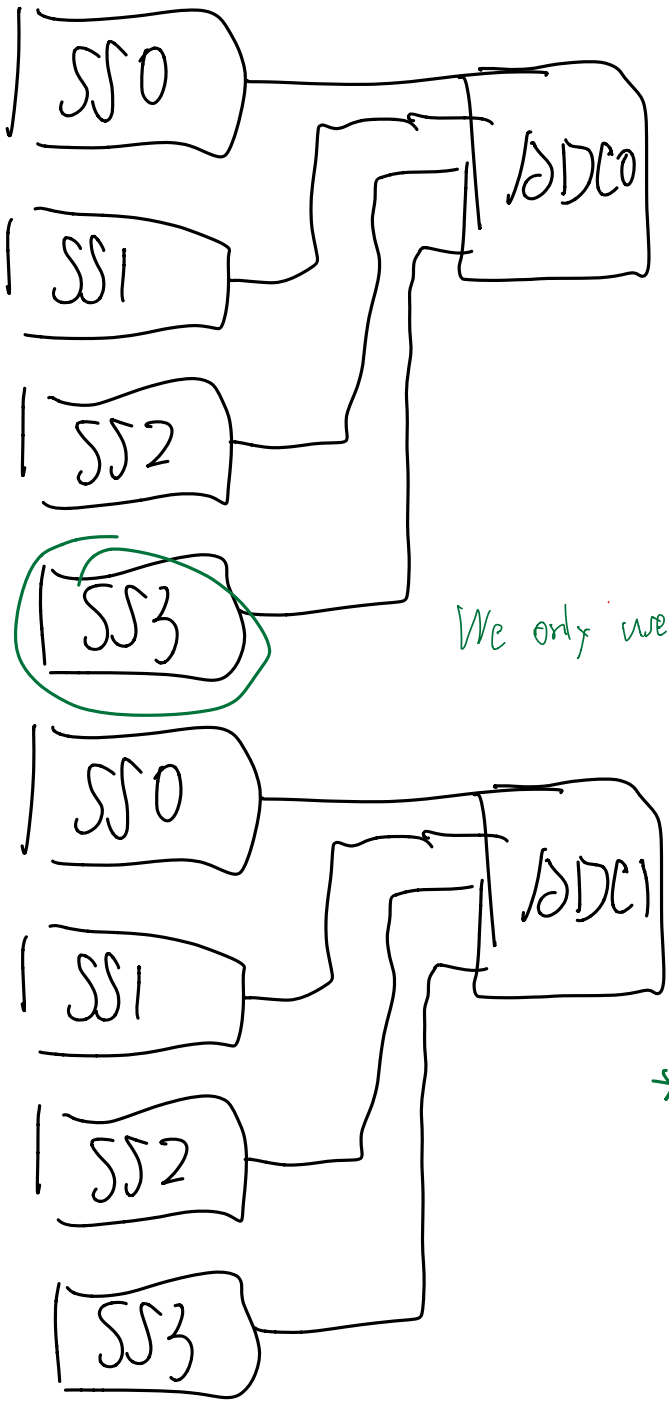
✓ we use PD2 ~~this~~ time, which aka "analog input #5" (ain5)

– how many different slide pots can TM4C support?

2? actually no..... because we have sequencer!!

each SSC has 4 sequencer...

so 2 x 4 = 8 slide pots max!



We only use Sequencer 3 in this Lab "SS3"

* each Sequencer will have priority, set in Sample Sequencer Priority hence "SSPRI"

- when to take sample
  - software trigger / GPIO / continous

we use software trigger ✓

Event_MUX_register "EMUX"

- how fast is the adc?

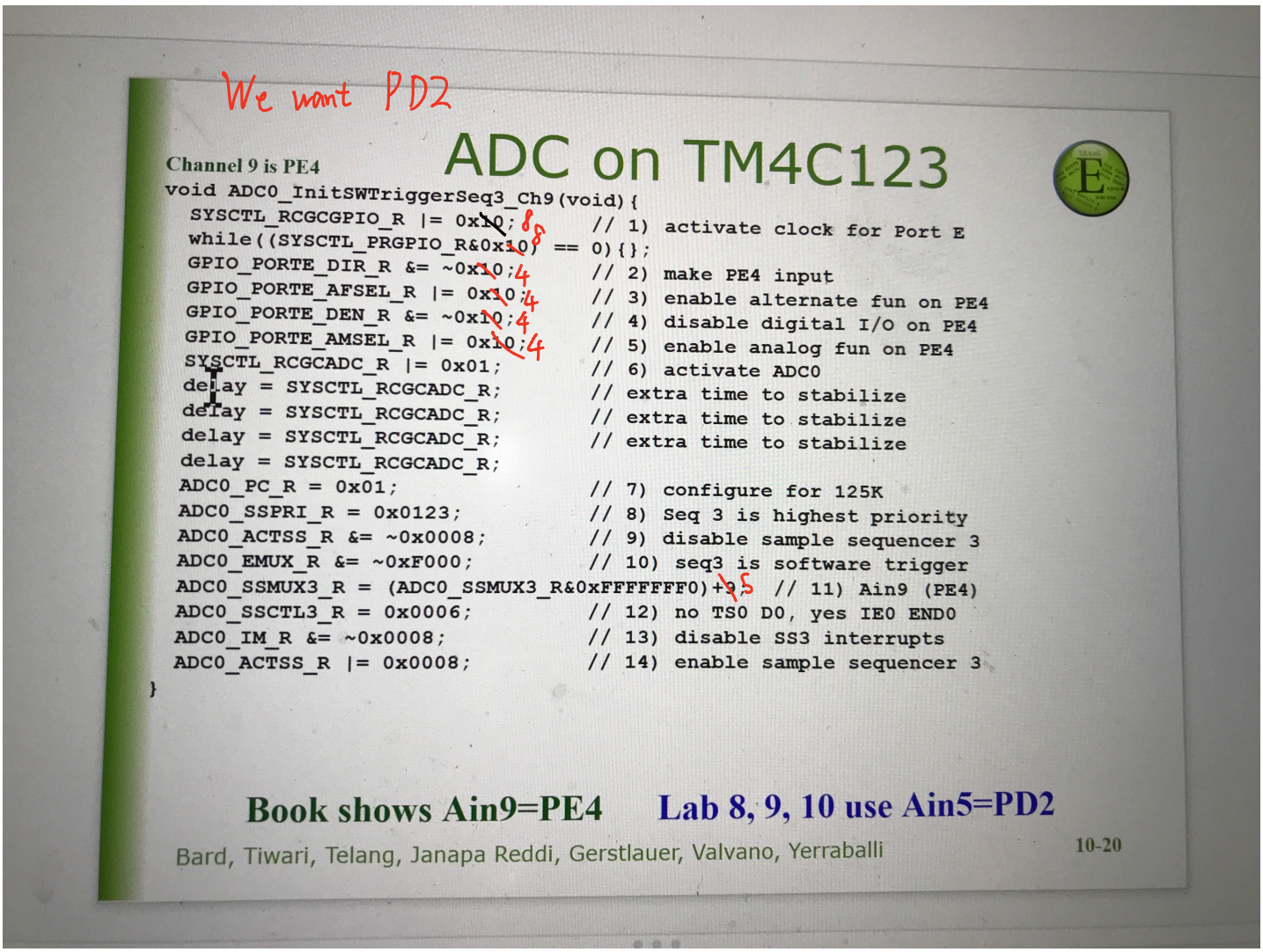
125k/sec in Peripheral configuration "PC"

- averaging?

Yes 64

Sample Averaging Control → "SAC"

- Bring it all together...



We want PD2

ADC on TM4C123

Channel 9 is PE4

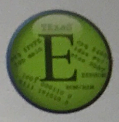
```
void ADC0_InitSWTriggerSeq3_Ch9(void){
  SYSCTL_RCGCGPIO_R |= 0x10;        // 1) activate clock for Port E
  while((SYSCTL_PRGPIO_R&0x10) == 0){};
  GPIO_PORTE_DIR_R &= ~0x10;        // 2) make PE4 input
  GPIO_PORTE_AFSEL_R |= 0x10;       // 3) enable alternate fun on PE4
  GPIO_PORTE_DEN_R &= ~0x10;        // 4) disable digital I/O on PE4
  GPIO_PORTE_AMSEL_R |= 0x10;       // 5) enable analog fun on PE4
  SYSCTL_RCGCADC_R |= 0x01;         // 6) activate ADC0
  delay = SYSCTL_RCGCADC_R;         // extra time to stabilize
  delay = SYSCTL_RCGCADC_R;         // extra time to stabilize
  delay = SYSCTL_RCGCADC_R;         // extra time to stabilize
  delay = SYSCTL_RCGCADC_R;
  ADC0_PC_R = 0x01;                 // 7) configure for 125K
  ADC0_SSPRI_R = 0x0123;            // 8) Seq 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 = (ADC0_SSMUX3_R&0xFFFFFFF0)+9; // 11) Ain9 (PE4)
  ADC0_SSCTL3_R = 0x0006;           // 12) no TS0 D0, yes IE0 END0
  ADC0_IM_R &= ~0x0008;             // 13) disable SS3 interrupts
  ADC0_ACTSS_R |= 0x0008;           // 14) enable sample sequencer 3
}
```

Book shows Ain9=PE4 **Lab 8, 9, 10 use Ain5=PD2**

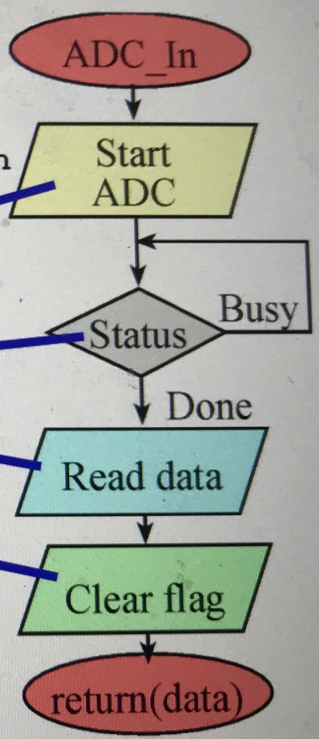
Bard, Tiwari, Telang, Janapa Reddi, Gerstlauer, Valvano, Yerraballi

10-20

ADC on TM4C123



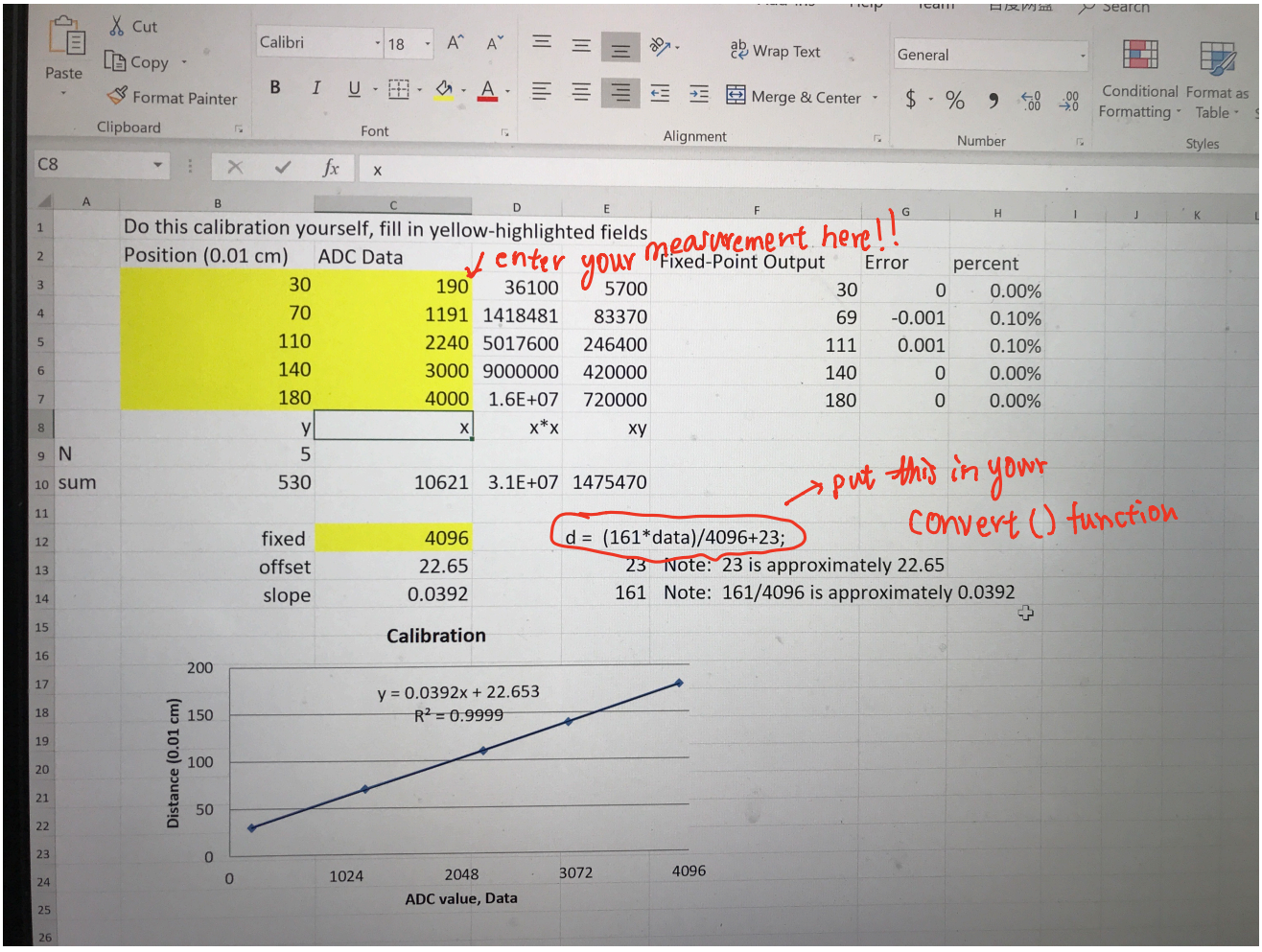
```
//------------ADC_InSeq3------------
// Busy-wait analog to digital conversion
// Input: none
// Output: 12-bit result of ADC conversion
uint32_t ADC0_InSeq3(void){
uint32_t data;
  ADC0_PSSI_R = 0x0008;
  while((ADC0_RIS_R&0x08)==0){};
  data = ADC0_SSFIFO3_R&0xFFF;
  ADC0_ISC_R = 0x0008;
  return data;
}
```



4) ADC Data to actual length

Use Calibration.xls in your Lab 8 folder! (part C)

- record your ADC raw Data in 3mm, 7mm, 1.1cm, 1.4cm, 1.8cm & then



Do this calibration yourself, fill in yellow-highlighted fields

Position (0.01 cm)	ADC Data			Fixed-Point Output	Error	percent
30	190	36100	5700	30	0	0.00%
70	1191	1418481	83370	69	-0.001	0.10%
110	2240	5017600	246400	111	0.001	0.10%
140	3000	9000000	420000	140	0	0.00%
180	4000	1.6E+07	720000	180	0	0.00%
y	x	x*x	xy			
N 5						
sum 530	10621	3.1E+07	1475470			

enter your measurement here!!

put this in your convert() function

fixed	4096	d = (161*data)/4096+23;
offset	22.65	23 Note: 23 is approximately 22.65
slope	0.0392	161 Note: 161/4096 is approximately 0.0392

5) write systick interrupt so that ADC_in will be called inside handler instead of being called in main directly

Why? - more accurate timing

- release CPU from doing useless while Loop

How often?

- 10Hz because of Nyquist Theorem

How?

- initialize: Look at your Lab 6 code!
- in your systick_handler:

```
PF1 ^= 0x02; //toggle LED for Texas Display
MailValue = ADC_in(); //MailValue is a global variable
MailStatus = 1; //MailStatus is a global variable for main
```

- in your main, instead of calling main:

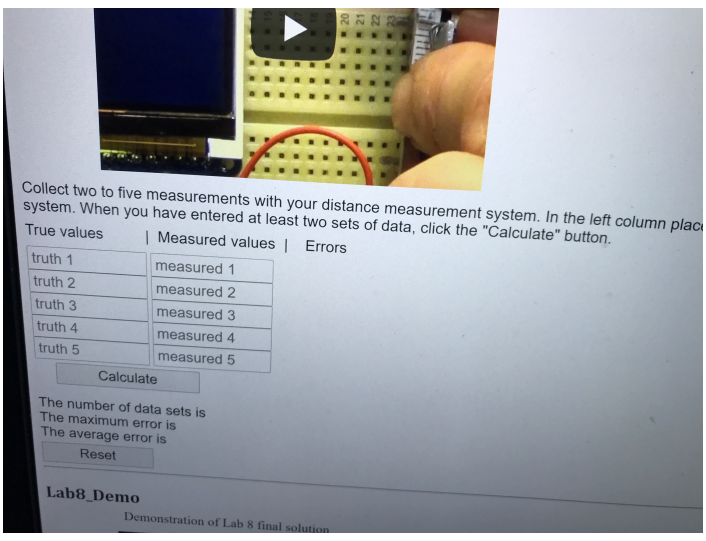
```
while(1) {
    while (MailStatus == 0) {};
        PF3 ^= 0x08;
        mailstatus = 0; //indicate mail is read
        Position = Convert(MailValue);

    LCD_OutFix(Position);
}
```

Verify: connect your PF1 to PD3 & use Texas_Display!

6) measure accuracy:

- Measure couple location & record actual position & measured position by ADC

- cacalate error & average accuracy in Ch14 ebook:



DONE!!