

March 16 (wed)

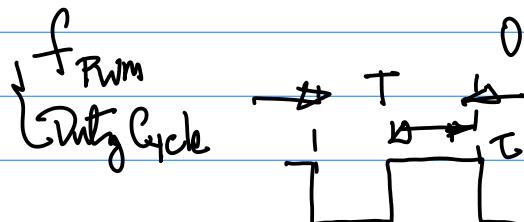
Topics: 1° PWM Architecture/Hardware

Aspect — Waveforms,

Timing Diagrams, SPRs

2° LSM303 Sensor I2C

Example: PWM Discussion.



$$\text{Duty Cycle} = \frac{D}{T} \dots (1)$$

Square Wave: D.C. = 50%.

Architectural Aspects :

2021F-105-#0-cpu-arm11-2018S-29...

PL10b

6410X\_UM

32

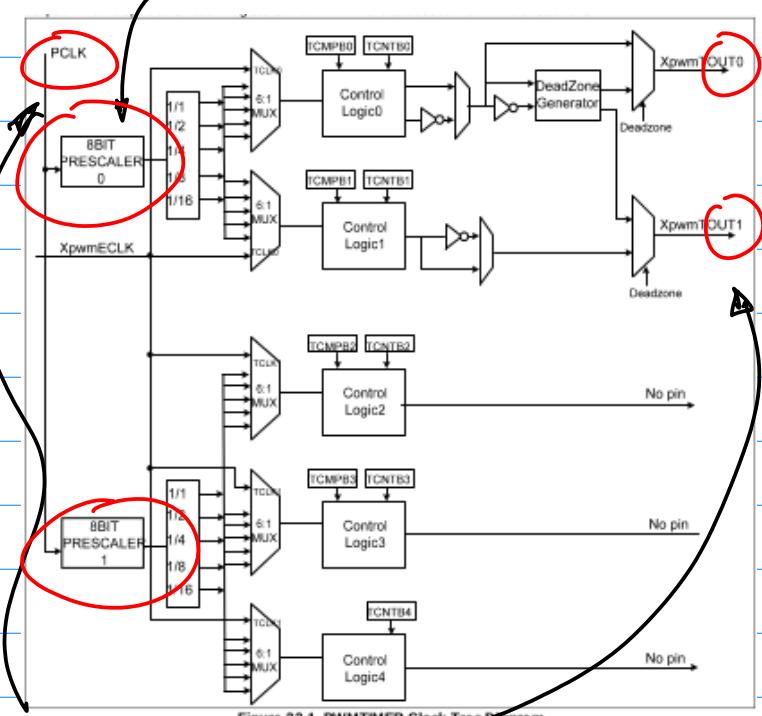
PWM TIMER

This chapter describes the functions and usage of PV

c. Prescaler for  $f_{\text{PWM}}$  Config

8 bit

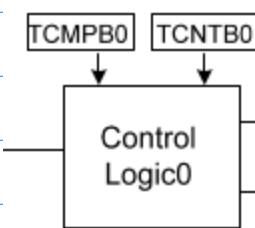
PP1107



a. Input: PCLK peripheral clock.

b. Output (2 outputs)  
 $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$  of the System Clock

d. Special Purpose Registers



CONF (Configuration Register)

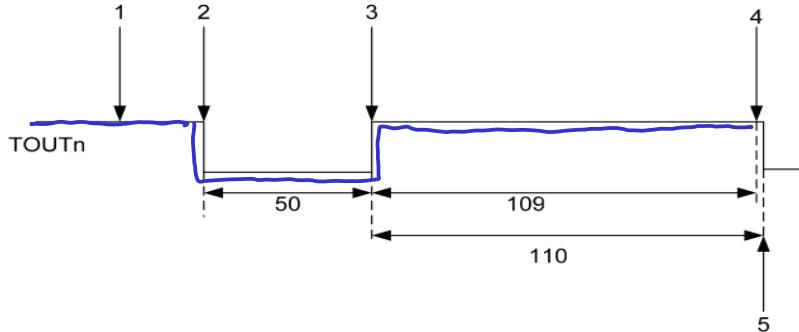
CNT (Control — "Count")

CMP (Comparison)

→ define  $f_{\text{PWM}}$  & Duty Cycle.

- 2022S-107e-pwm-waveform-v3-2018-3-4.jpg
- 2022S-107f-pwm-specialPurposeRegister-v3-2018-3-4.jpg
- 2022S-107g-pwm-calculation-v3-2018-3-4.pdf

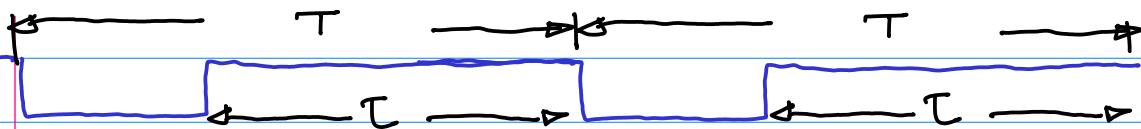
Note: Background on Counters.



Count By 2 ↗

Expand this to A counter for Both integer Number & Fractional Number (in general)  $\Rightarrow f_{PWM}$  (for integer Only)  
then, use Another Counter to get Duty Cycle.

Figure 32-2. Simple Example of PWM Cycle Block Diagram



$$T = \frac{1}{f_{PWM}} \dots (2), \quad f_{PWM} = \frac{PCLK}{(\text{Prescaler}+1)\text{DIV}} \dots (3)$$

Frequency = PCLK / ( {prescaler value + 1} ) / {divider value}

$$\text{Suppose } PCLK = \frac{1}{4} (\text{System Clock}) = \frac{1}{4} (800 \times 10^6) \\ = 200 \times 10^6$$

Defined By Special Purpose Register.

Design Guidelines for PWM:

① CNT ( $T_{CNTB\phi}$ ) Define  $f_{PWM}$   
Control / Count ↑ ↑  
Timer Buffer φ for PWM φ

③ CONF Configuration Register.  
Defines  $f_{PWM}$

② CMP ( $T_{CMPP\phi}$ ) Defines Duty Cycle

Comparison

Example: Suppose  $PCLK = 500 \times 10^6$  (MHz), Find the Counts for  $T_{CNTB\phi}$

Suppose to Drive A Stepper

Motor  $f_{PWM} = 2 \text{ kHz} = 2 \times 10^3$

$$\frac{PCLK}{N} = f_{PWM} \dots (4)$$

$\frac{PCLK}{N} = f_{PWM}$ , Substitute the design Requirements into it.

$$\frac{500 \times 10^6}{N} = 2 \times 10^3$$

$$\therefore N = \frac{500 \times 10^6}{2 \times 10^3} = 250 \times 10^3$$

Verify if TCNTB0 can hold up to that Number

PP1117

### 32.4 SPECIAL FUNCTION REGISTERS

#### 32.4.1 REGISTER MAP

Register	Offset	R/W	Descrip
TCFG0	0x7F006000	R/W	Timer Configuration Register two 8-bit Prescaler and De-
TCFG1	0x7F006004	R/W	Timer Configuration Register and DMA Mode Select Bit
TCON	0x7F006008	R/W	Timer Control Register
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Register
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Register
TCNTO0	0x7F006014	R	Timer 0 Count Observation
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Register
TCMPB1	0x7F00601C	R/W	Timer 1 Compare Buffer Register

#### 32.4.1.4 TCNTB0 (Timer0 Counter Register)

Register	Offset	R/W	
TCNTB0	0x7F00600C	R/W	Timer 0 Count Register

Conclusion: 5-Steps operation of PWM. Can be described as (1) Count By N with Eqn (4), PP3. And deposit

N into TCNTB0; (2) Deposit Count M into TCMPB0, where  $M = (\text{D.C.}) \times N$  ... (5)

(3) The Down Counting will decrement TCNTB0's count by 1 at a time, And a

Comparison is made to TCMPB0, if matched, then trigger the Output to "1", Down Counting continues till the Count in TCNTB0 = 0 One period is reached. Then Repeat this process.

Feb. 21. 21.

Example: Suppose CLK = 50 MHz.  
Design Implementation Technique to produce  $f_{PWM} = 1000 \text{ Hz}$ , to

Drive Stepper Motor Controller, in addition, Duty Cycle is 30%.

Find: (1) TCNTB0 = ? ; (2) Find TCMPB0

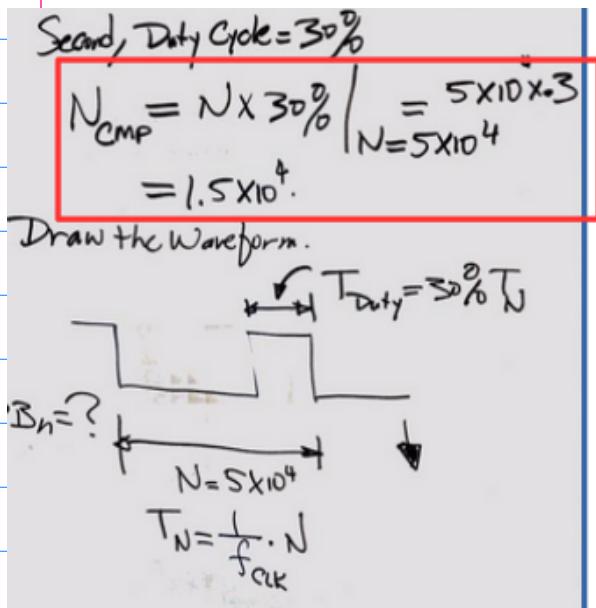
Sol First, find the Counts N Based on the given condition.

$$\frac{50 \times 10^6}{N} = f_{PWM}, \quad \dots (1)$$

$$N = \frac{50 \times 10^6}{1 \times 10^3} = 5 \times 10^4 \rightarrow \text{Hex}$$

$$f_{PWM} = 1000$$

$$TCNTB0 \leftarrow$$



Software Side      { O.S. distribution,  
Toolchain,  
menu config.

### f. SPRs in Driver Code.

GPIO User program.

```
fd = Open("~~~~~");  
path/Driver  
ioctl(~~~~~);
```

### Kernel Space Program Sample

GPECON, etc  $\rightarrow$  CPU

DataSheet

### g. GPIO Testing, I/P Testing, D/P testing.

Ref:

[2022S-101-notes-cmpe242-3-14.pdf](#)

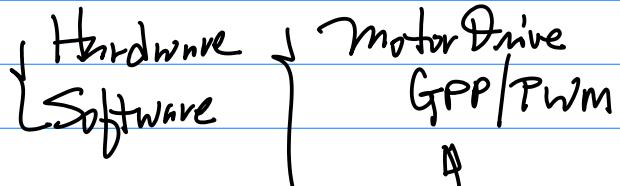
PP.9      CKT  $\rightarrow$  Pin Selection

GPIO 79 (Pin 12)

GPIO 78 (Pin 40)

2.

Question(s) on PIO Controller Design.



Motor Drive Pin Connection Requirements

Sch., Connectivity Table.

### March 21 (Monday)

Midterm on 23rd (Wed).

1 hr. Exam, + 15 min.

Review on Midterm.

3 Questions.

### 1. A question on Basic Concepts.

i. CPU Architecture

32 Bit Architecture

a. Memory Map, Banks,

b. GPP/I/O Peripheral Controller.

c. SPRs. Naming, functions.

GPxCON, GPxDAT

Tech. Spec  $\rightarrow$  Binary Pattern

d. ARM11 Reference Code

User Space, Kernel Space

e. Target Platform, NAND,

"Demo Live Execution of your program.

i. Board is Ready. Take a photo of your Board During the exam.

ii. Stepper motor/motor Drive And the Prototype Board Work together, Take a photo, Screen Capture of program execution.

Motor Operation, Micro Steps, Angular Displacement.

Target Platform. Hardware Configuration to access device driver, Such as GPIO, PWM or I<sub>C</sub>.

3. Theoretical Aspects of the PID Controller Design.

a. PID Block Diagram.

$P_N, P_{IN}, P_D, \dots$  etc.  
Derivative Controller) Shift future.

I (Integration Controller) history

b. Computation. Forward Difference, Backward Difference, Central Difference. Kernels, Computation.

Integration Controller. history  
Back N steps.

c. Sensor Interface Hardware I<sub>C</sub> I/F.

LSM303 I<sub>C</sub> Based Sensor.

Example: An Application of LSM303

Reaching of the vehicle, "skipping" may occur

2022S-108b-AngularSensing-i2c-LSM303- final HL 2017-3-13.pdf

$\alpha$ : from LSM303

$P_A$ : is measurable

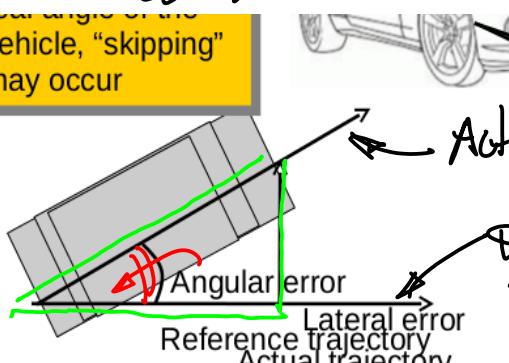


Fig.1.

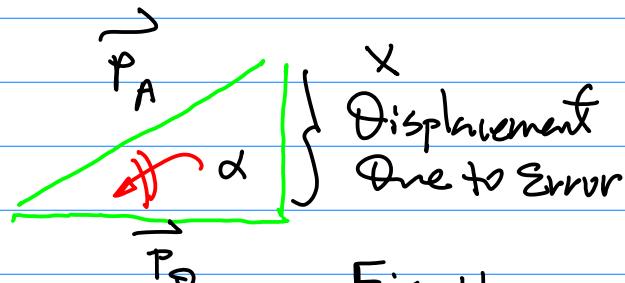
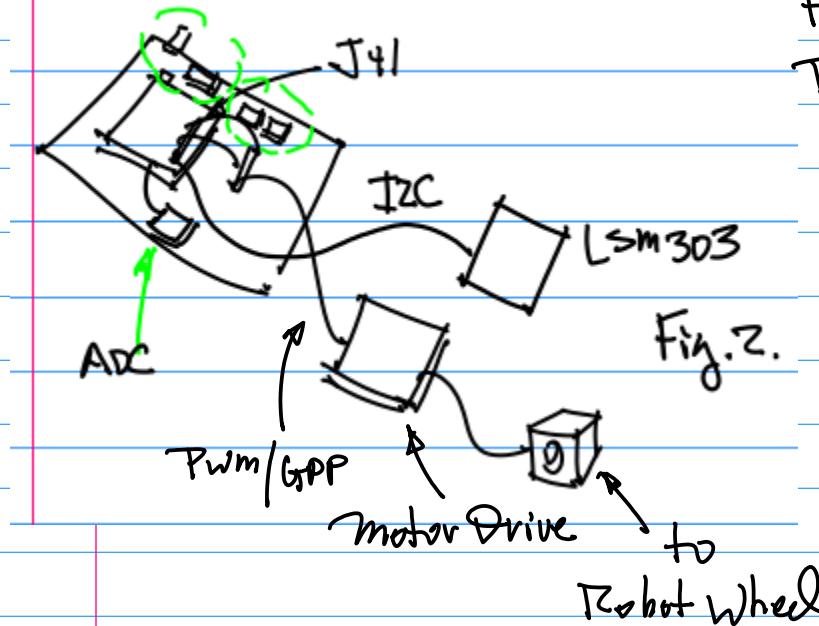
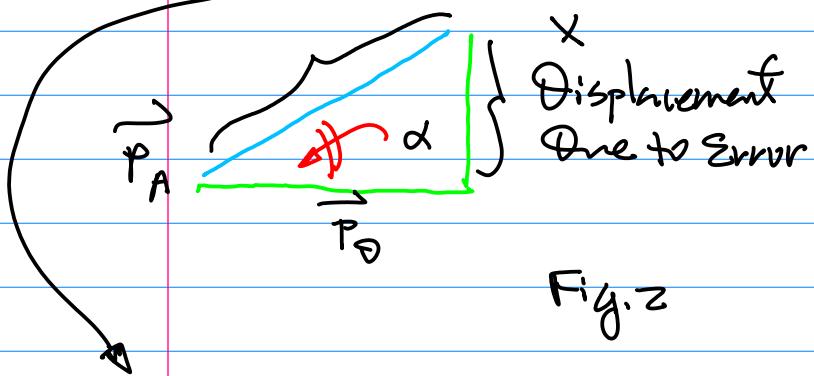


Fig.1b

CmpE442 March 21, 22

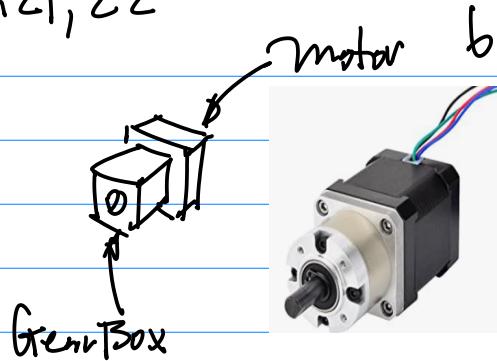
Find  $\|\vec{P}_A\|$  is defined by  
PWM Driven Stepper motor  
action.



$$\frac{1}{R}$$

the wheel of the  
Robot has the  
dimension  $R = 100\text{mm}$

Now, Let's take a look at  
the hardware of the motor  
Combo.



STEPPERONLINE Nema 17 Geared Stepper  
Motor Gear Ratio 5:1 3D Printer Extruder  
Motor DIY CNC Robotics  
Visit the STEPPERONLINE Store  
★★★★★ 25 ratings  
Amazon's Choice in 3D Printer Motors by STEPPERONLINE  
 $\$40^{05}$

Reduction Ratio,  $R_R$

For the purpose of increasing the  
Torque.



Fig. 4a,4b With Reduction Ratio.  
Gear Box.



To find / Establish one-to-one mapping between the actual path and PWM operation.

A point (Monday)

Topics: 1° I2C Interface for LSM303 Sensor Integration.

Ref:

2022S-108b-AngularSensing-i2c-LSM303- final HL 2017-3-13.pdf

### I2C Hardware Features

Protocol Definition

Coding Implementation

Hardware Features:

1. pins
  - SDA : Bi-Directional Data Pin. (Signals SCK: in Both Direction) SerialClock.

Speed:  $\sim 4 \text{ mbps}$ ,  $\leq 10 \text{ mbps}$

2. I2C Bus

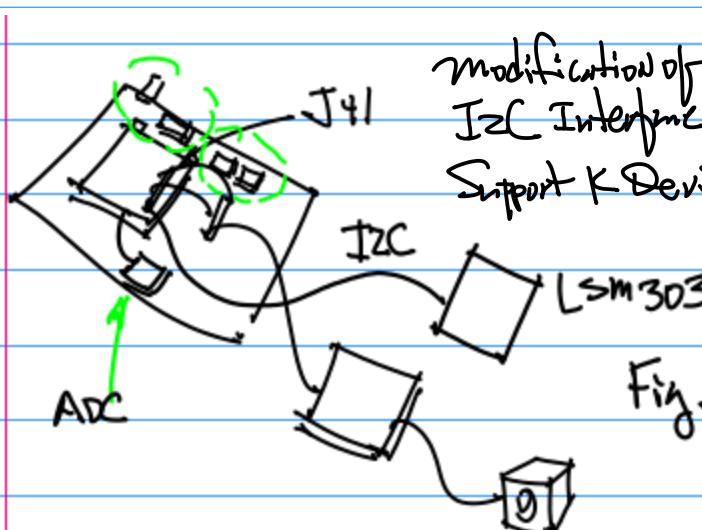
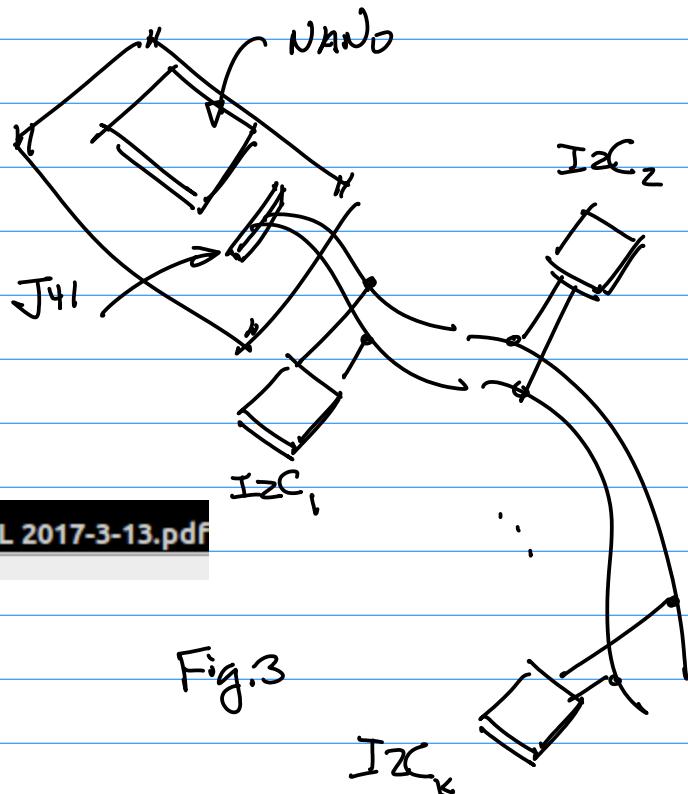


Fig. 3



Sub-Address: 7 bits,  $2^7 = 128 \rightarrow K=128$  Devices, But in Real Engineering Design, "FAN-IN", "FAN-OUT" (e.g. Adequate Electric Current) have to be taken into consideration.

Bus Communication

Master Device  
Slave Device(s)

Ref:

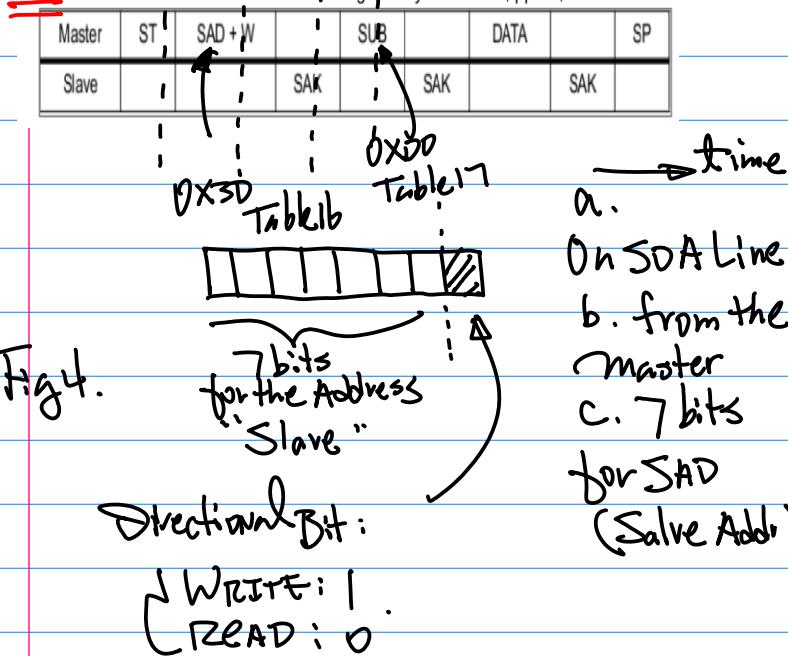
2022S-108-LSM303DLHC.PDF

### I2C Interface

- (1) The transaction started through a START (ST) signal, defined as a high-to-low on the data line while the SCL line is held high.
- (2) After ST, the next byte contains the slave address (the first 7 bit), bit 8 for if the master is receiving or transmitting data.
- (3) When an address sent, each device compares the first seven bits after ST. If they match, the device is addressed.

Note:  $t_0, t_1, t_2, t_3$

Table 11. Transfer when master is writing one byte to slave, pp 20, datasheet



Note: 1° ST Start

2° SAD Slave Address

3° SAK Slave Ack.

Question: SAD for LSm303

Table 14. SAD+Read/Write patterns

Command	SAD[7:1]	R/W	SAD+RW
Read	0011001	1	00110011 (33h)
Write	0011001	0	00110010 (32h)

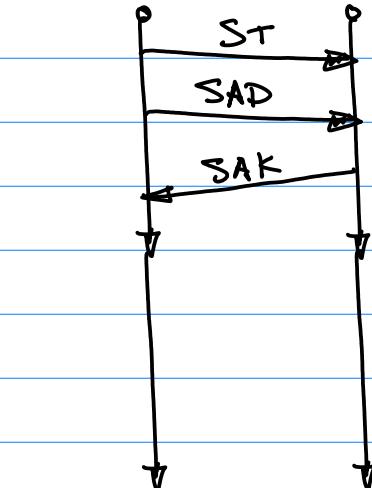
pp.20

0011001x  
3 ① 0011 Read: 3  
② 0010 Write: 2  
0X33 Read  
0X32 Write

Note: Read Op is very often the 1st  
One from the master to get  
Manufacturer's ID & Device ID etc.

Note: Space-Time Diagram.

Space →  
Target(NAND) "Slave"  
I2C Master LSm303  
Spatial

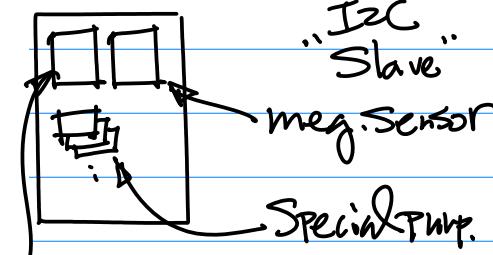


Time

Fig. 5

And at  $t_3$ : SUB Sub-Slave-Add. to address the unit inside the Slave Device. (LSm303)

Fig. 6



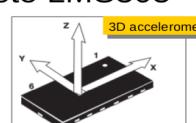
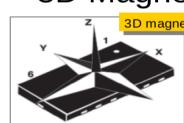
MCU: microcontroller

FSM: Finite State Machine

SPRs: Config & Control Data  
Sensing

Example: Find Sub-Add.

3D Accelerometer and  
3D Magnetometer LMS303



Ref: Slide 7. a. Sub-Addr. for  
magnetic Sensor

2. identify control register(s) for the right sensor

block with the sub-address to set data rate

(1) CRA\_REG\_M register (0x00) to set data rate

Control/Config Register's Address: 0x00  
"Sub"

Table 16. SAD For Magnetic Sensor.

Command	SAD[6:0]	R/W	SAD+R/W
Read	0011110	1	00111101 (3Dh)
Write	0011110	0	00111100 (3Ch)

Table 17. Register address map (continued)

Name	Slave address	Type	Register address		Default
			Hex	Binary	
TIME_LATENCY_A	Table 14	rw	3C	011 1100	00000000
TIME_WINDOW_A	Table 14	rw	3D	011 1101	00000000
Reserved (do not modify)	Table 14		3E-3F	--	--
CRA_REG_M	Table 16	rw	00	00000000	0001000
CRB_REG_M	Table 16	rw	01	00000001	0010000
MR_REG_M	Table 16	rw	02	00000010	00000011
OUT_X_H_M	Table 16	r	03	00000011	output

Example: Tech Spec.

1° Read Angular Information / x-, y-, z-Acceleration  
2° Sample Rate (of Read) : 30 Hz

Find 1° Special Purpose Register Responsible  
to perform Configuration.

2° Find Binary Pattern to initialize  
the SPR.

April 7. Wed

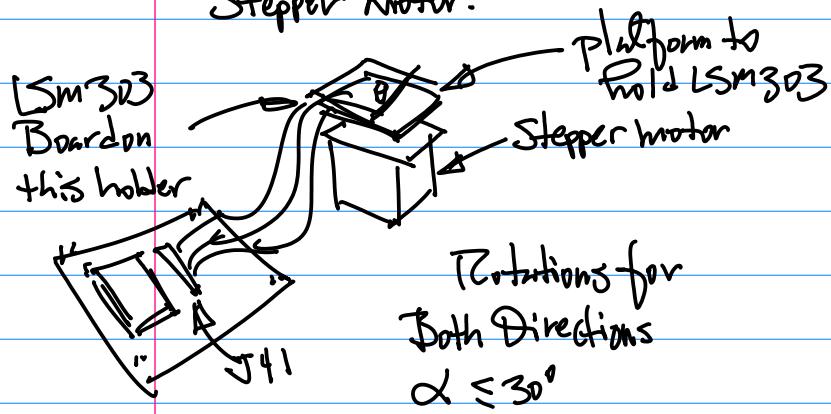
Topics: 1° LSM303 Interface Design.

Project Due April 18 before Class.

## Project Requirements:

1. Written Requirements + Rubrics Are to Be posted online, on CANVAS.

2. Integrate LSm303 Sensor to Stepper motor.



3. To Run Stepper motor program to Activate motor Rotation, then Read LSm303 Data to find for each Actual the displacement Value.

4. Test 3 Configuration of the motor, e.g.

- a. Full Step. 1.8 Degree Angle per Step.
- b. Half Step. 0.9 Degree
- c.  $\frac{1}{4}$  Step. 0.45 Degree

5. Form A table to List the Data Read from LSm303

PWM Pulses	LSM303 Input

Note: Try to Run multiple pulses of the PWM, for Example, 5nd Pulses. When collecting Data

b. Create README file, one page to Describe your implementation.

a. Schematics

b. Screen Capture of your code execution

c. Generate 10 Sec. ~ 15 Sec. Video Clips to Show the Working System.

d. 2 photos. { One for the entire System. (Laptop, Target Platform, Stepper motor, Sensor)

One for the Sensor (LSM303 Interface Design)

7. Includes All your Source code for testing/Verification.

8. Put all of the Above into A zip file, Submit the file to CANVAS

Note: please Bring your Implementation Board to the Class for Demo. Next Monday.

Example: Process/Steps to Interface to LSm303.

Step 1. Addr. of the Sensor. See Table 1b For many Sensor.

Note: Table 17, Special Purpose Registers { For Config/Control for Data

## 7.2 Magnetic field sensing register

## 7.2.1 CRA\_REG\_M (00h)

8 bit Register

a.

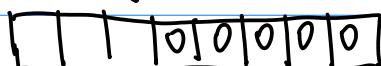


Table 73

CRA\_REG[7:5]

Table 73. CRA\_REG register

GN2	GN1	GN0	0(1)	0(1)	0(1)	0(1)	0(1)
-----	-----	-----	------	------	------	------	------

1. This bit must be set to '0' for correct working of the device.

Table 75. Gain setting

GN2	GN1	GN0	Sensor input field range [Gauss]	Gain X, Y, and Z [LSB/Gauss]	Gain Z [LSB/Gauss]	Output range
0	0	1	$\pm 1.3$	1100	980	
0	1	0	$\pm 1.9$	855	760	
0	1	1	$\pm 2.5$	670	600	
1	0	0	$\pm 4.0$	450	400	0xF800-0x07FF (-2048-2047)
1	0	1	$\pm 4.7$	400	355	
1	1	0	$\pm 5.6$	330	295	
1	1	1	$\pm 8.1$	230	205	

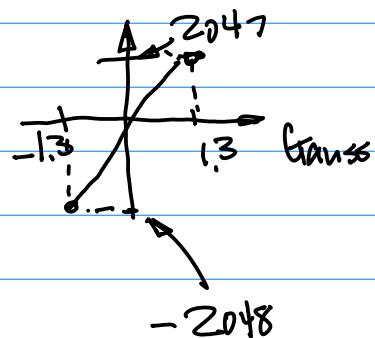
Larger Range/Less Sensitive

b. For Mag Sensor

Smaller Range/more Sensitive

c. Out Range

$$Z'' = 1024 \times Z \\ = 2048$$



$$GNZ=0, GN1=0, \\ GN\phi = 1$$

Note: Start with either default Setting of the Gain, or Set the gain to  $GNZ\text{ }GN1\text{ }GN0 = 100$

Then, Control Register for Operation mode Selection

Only 2 Bits

4. identify the control register responsible for mode of operation, from datasheet, pp 37, table 76, 78 MR\_REG\_M (02h) Sub-Addr. for this Register

0(1)	0(1)	0(1)	0(1)	0(1)	0(1)	MD1	MD0
------	------	------	------	------	------	-----	-----

Use Continuous mode

MD1	MD0	Mode
0	0	Continuous-conversion mode
0	1	Single-conversion mode

To Config/Access to Special Purpose Registers), follow this "Write" Pattern:

SPR's Address: 0x01 for GRAN Setting, CRB-REG-M  
0x02 for mode MR-REG-M,



Binary Pattern

For Reading After the Configuration, we will need find SADR (Sub Addr. for Read), Example is given in PPT, from Data sheet, Table 5, pp.20.

Coding Implementation : C/C++ Option  
Python Option.

This Capability is to be implemented in this Class as a part of the project Part of the

April 11 (Monday). ADC

Analog To Digital Conversion.

Industrial IOT Application.

Analog Sensor Interface

Fig.1

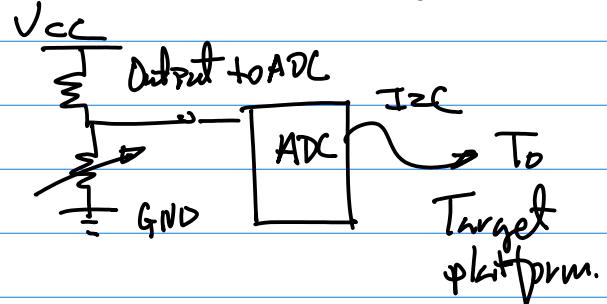
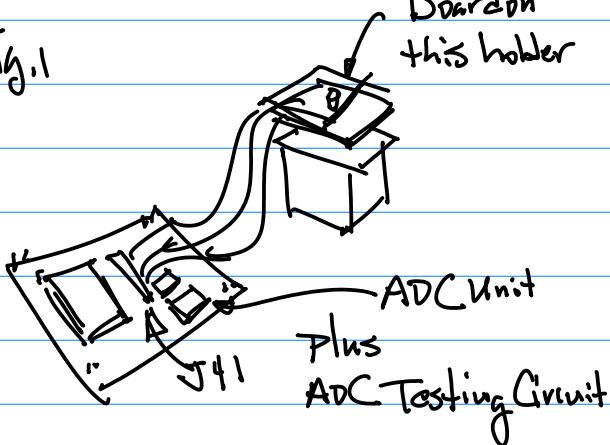


Fig.2.

Example: Background of Analog to Digital Conversion.

Sensor (Analog) Input, Analog Current



Potentiometer



Note: a. Analog Charge Output

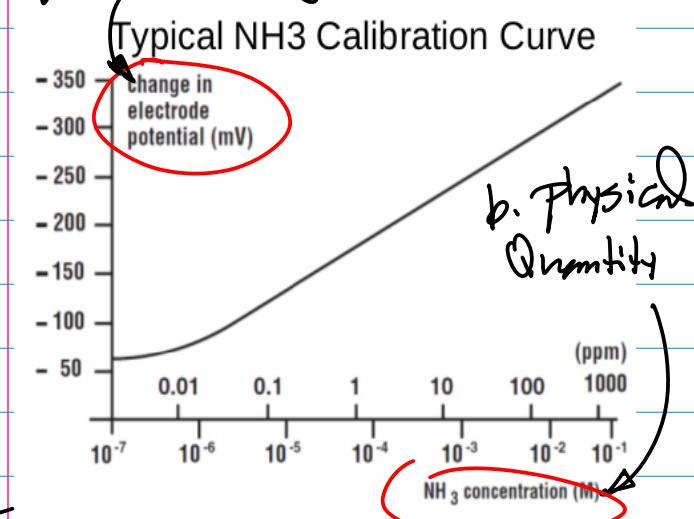
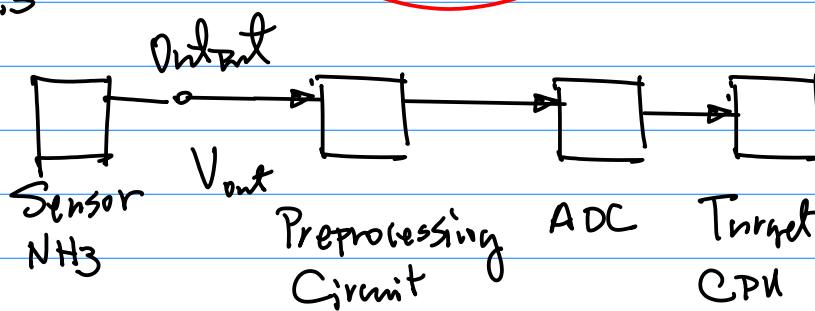
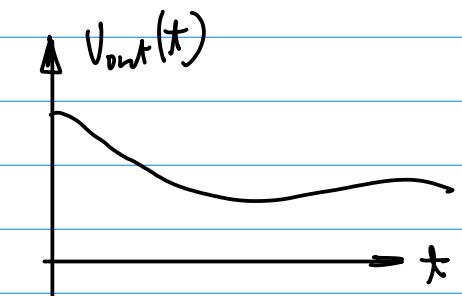
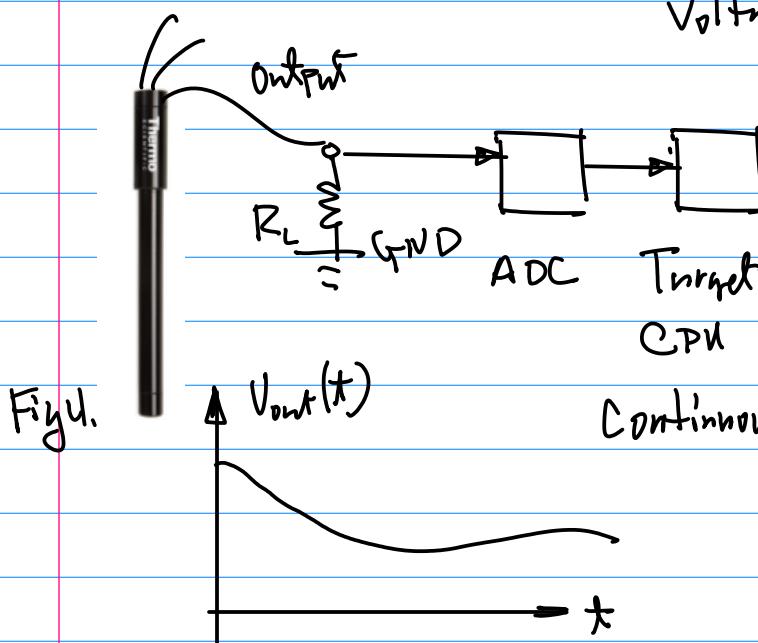


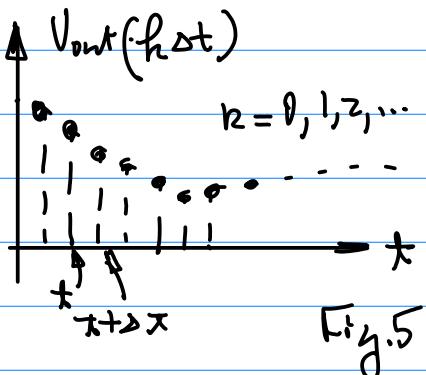
Fig.3



$R_L$  Conversion:  
Current i to  
Voltage v.



i Sampling  
ADC.



2<sup>o</sup> Quantization

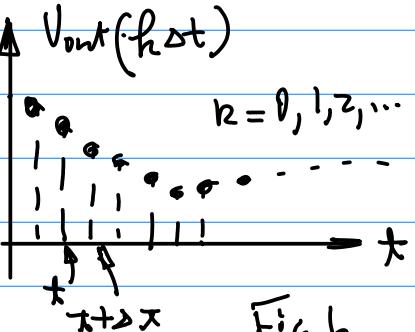


Fig.6

Note: the magnitude after the quantization becomes one of  $2^X$  Possible Levels.

(10 bits ADC,  
 $2^{10} = 1024$  levels,  
12 bit ADC,  
 $2^{12} \cdot 2^2 = 4096$  levels)

Performance of ADC:

### 1. Sampling Rate:

(Common) 500 KSPS  
 (Samples per Second)

### Sampling Theorem (Nyquist ~)

$$f_{\text{Sampling}} \geq 2f_{\text{max}} \quad \dots (1)$$

The Sampling Speed has to be greater than or equal to Twice of the maximum Speed of a given Signal.

### 2. Measurement of ADC. No. of Quantization

Level. 10 bits ( $2^{10} = 1024$ ), 8 bits ( $2^8 = 256$ )  
 12 bits ( $2^{12} = 4096$ ).

### 3. Linear Characteristic.

ADC Output (10 bits)

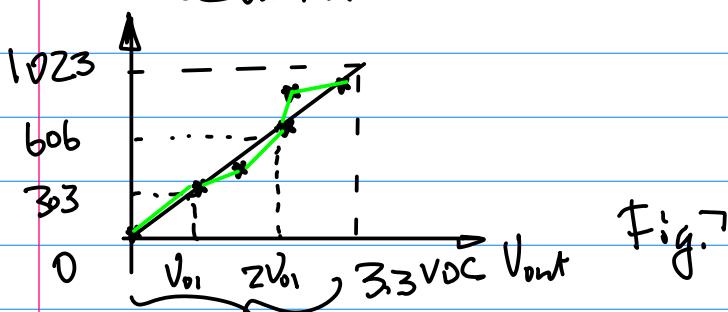


Fig. 7

Fig. 7 ADC Input Range  
 (Dynamic ~)

$$\frac{\Delta \text{Digital}}{\Delta V_{\text{out}}} = k \left( \text{Slope of the Line in Fig. 7} \right)$$

We have to measure Linearity of a given ADC.

Homework (After the project of LM353)

Formal Due Date to Be Announced in the next lecture.

- 1° Identify & Purchase your ADC module.



ADC with I2C Interface, ADS1115.

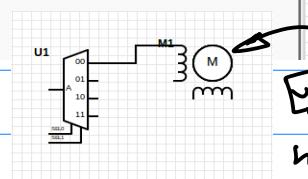
- 2° Generate Schematics.

<https://www.digikey.com/schemedit/project>

Scheme-it | Free Online Schematic

A screenshot of the Scheme-it online schematic editor interface. The top navigation bar includes 'HOME', 'DESIGN STARTERS', 'MAIN' (which is selected), 'BOM', and other options like 'New', 'Open', 'Save', 'Notes', 'Export', and 'Share'. A search bar is at the top right. The main area shows a sidebar with categories: 'Open Design Starters' (highlighted with a red arrow), 'Build a Symbol' (with a red circle around it), 'Digi-Key Catalog', 'Manufacturers', 'Schematic Symbols' (highlighted with a red circle), 'Diagram Symbols', 'System Blocks', 'System Icons', 'Custom Symbols', and 'Import BOM'. Below the sidebar is a workspace where a simple circuit diagram is being drawn, showing a component U1 connected to a motor M.

Fig. 8



Example here with a Stepper motor

3. Use your Built Circuit Below, to collect 10 pairs of Data Points.

$$(V_{o_1}, D_1), (V_{o_2}, D_2), \dots (V_{o_{10}}, D_{10}),$$

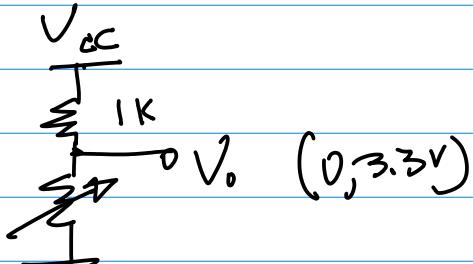


Fig. 4  
470 k $\Omega$  Potential meter

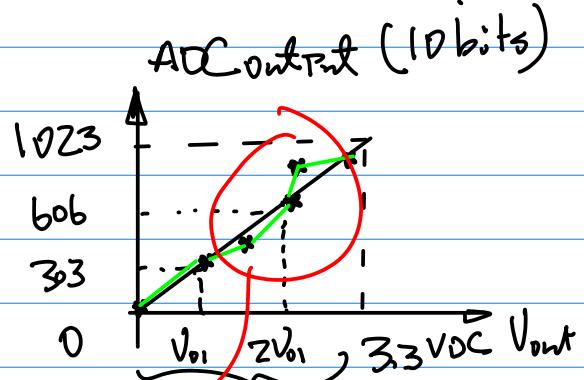
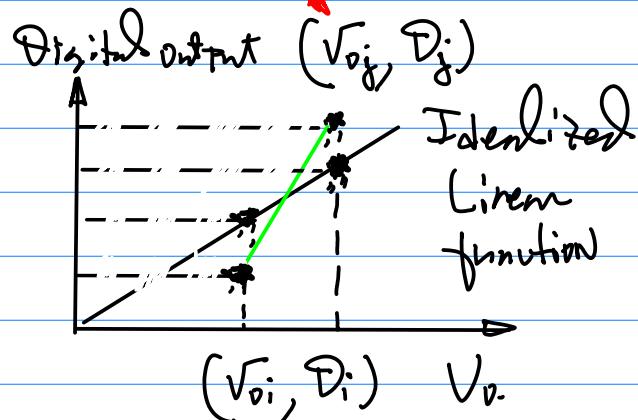


Fig. 11 ADC Input Range (Dynamic ~)

4. In order to do 3, you will have to have Python or C/C++ interface to your ADC.

5. plot your data, as in the

Fig. here



Actual measurement

Develop A Correction function to correct Non-Linear Behavior By Developing A function.

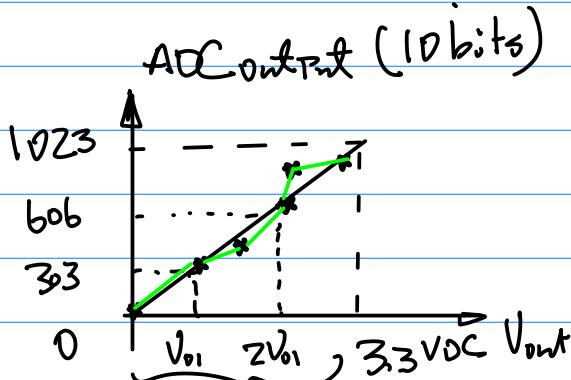


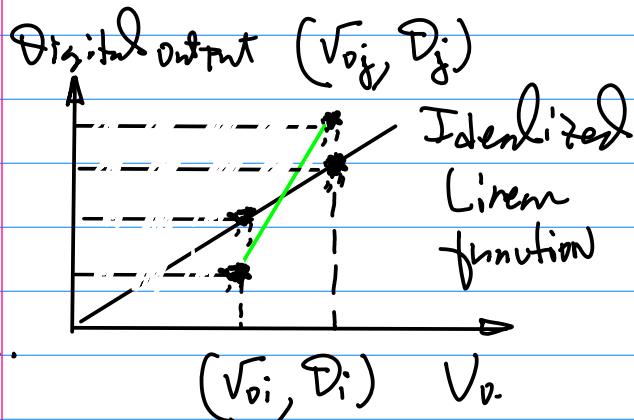
Fig. 10 ADC Input Range (Dynamic ~)

6. Correction of Non-Linear Behavior

April 13.

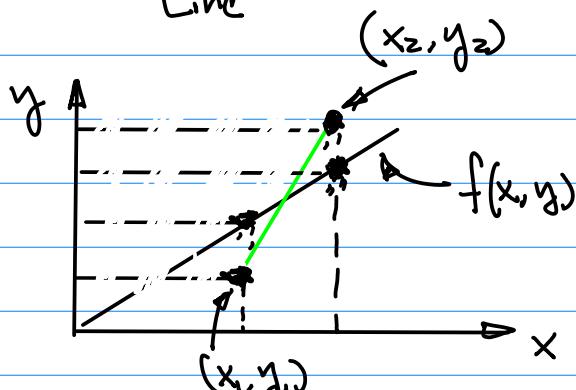
Topics: 1<sup>o</sup> ADC Linear Compensation

Example: To correct/compensate the NonLinear Deviation (green Line) to make it as linear function (Black Line).



Define  $g(x, y)$  as the "green" Line.

$f(x, y)$  as the desired Linear Characteristic Line



Objective: To Develop Post Processing Technique to Correct/Compensate the green Line (Actual Data).

Step 1. Desired Linear Characteristic Line.

Take A target form with An ADC  
ADC has: 10 bits ADC

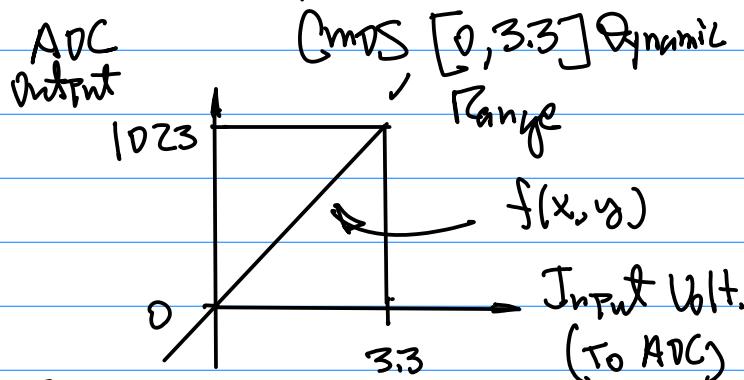


Fig 2

$$10 \text{ bits} = 2^{10} \text{ bits} \Rightarrow 1024 \text{ Levels}$$

Since  $y = ax + b \dots (1)$

where  $b=0$  B/C the Line passes the origin.

$a$  is a slope.

$$a = \frac{\Delta y}{\Delta x} \quad \begin{cases} \Delta y = 1024 \\ \Delta x = 3.3 \end{cases}$$

$$= 1024 / 3.3 = 310.3$$

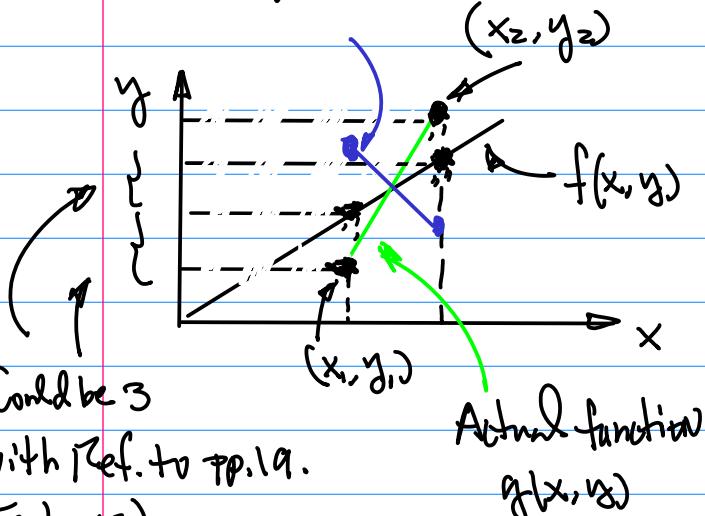
Step 2.

$$g(x, y) + h_{\text{Comp}}(x, y) = f(x, y) \dots (2)$$

we have

$$h_{\text{Comp}}(x, y) = f(x, y) - g(x, y) \dots (2-b)$$

Compensation function  $f(x, y)$



Step 3.  $g(x, y)$  from the Actual Data, we have from the Calibration Process.

From Fig. 1-b.

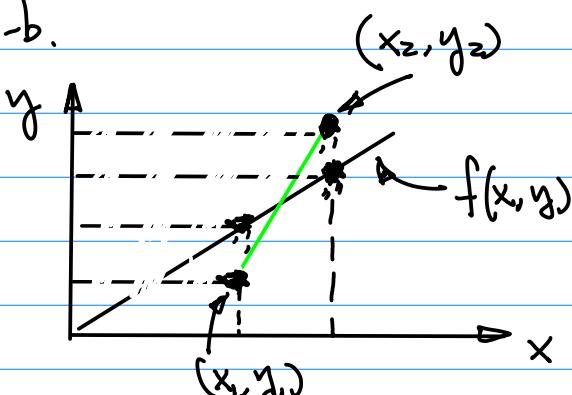


Fig 1-b

$$\frac{x-x_2}{y-y_2} = \frac{x_1-x_2}{y_1-y_2} = \frac{x_2-x_1}{y_2-y_1}$$

... (3)

$$y = ax + b.$$

$$x-x_2 = \frac{x_1-x_2}{y_1-y_2} (y-y_2)$$

$$\frac{y_1-y_2}{x_1-x_2} (x-x_2) = y - y_2$$

$$\frac{y_1-y_2}{x_1-x_2} (x-x_2) = y - y_2$$

$$\frac{y_1-y_2}{x_1-x_2} x - \frac{y_1-y_2}{x_1-x_2} x_2 + y_2 = y$$

$$y = \underbrace{\frac{y_1-y_2}{x_1-x_2}}_b x - \underbrace{\frac{y_1-y_2}{x_1-x_2} x_2 + y_2}_c$$

hence

$$y = bx + c \quad \dots (4)$$

where

$$b = \frac{y_2-y_1}{x_2-x_1} \quad \dots (4-b)$$

$$c = -\frac{y_2-y_1}{x_2-x_1} x_2 + y_2 \quad \dots (4-c)$$

From Step 2, Eqn (2-b), find the Compensation.

$$h(x, y) = f(x, y) - g(x, y)$$

where  $f(x, y)$ :  $y = ax$

$$f(x, y) = y - ax \quad \dots (5)$$

$$g(x, y) : y = \cancel{a'} x + \cancel{c'}$$

$$g(x, y) = y - (a' x + b') \quad \dots (b)$$

Therefore

$$\begin{aligned}
 h(x, y) &= y - ax - [y - (b'x + c')] \\
 &= -ax + b'x + c' \\
 &= (a' - a)x + b'c' \\
 &= (b' - a)x + c' \quad \dots (7)
 \end{aligned}$$

## Hardware, Architectural Aspects of ADC

2021F-105-#0-cpu-arm11-2018S-29...

github, in Z44 folder

6410X\_UM

TP1209

# 39

## ADC & TOUCH SCRE

This chapter describes the functions and usage of ADC & Touch

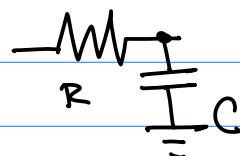
### 39.1 OVERVIEW

The 10-bit/12-bit CMOS ADC (Analog to Digital Converter) is a recycling type device with 8-channel analog inputs. It converts the analog input signal into 10-bit/12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. The power down mode is supported.

Touch Screen Interface can control input pads (XP, XM, YP, and YM) to obtain X/Y-position on the external touch screen device. Touch Screen Interface contains three main blocks; these are touch screen pads control logic, ADC interface logic and interrupt generation logic.

$$\text{Sampling,}_{\text{max}} = 1 \times 10^6 \text{ Hz}, f_{\text{ADC}} = 5 \times 10^6 \text{ Hz}$$

e. Sample-And-Hold



) Inside ADC

### Schematic Symbols

Amplifiers

Circuit Protection

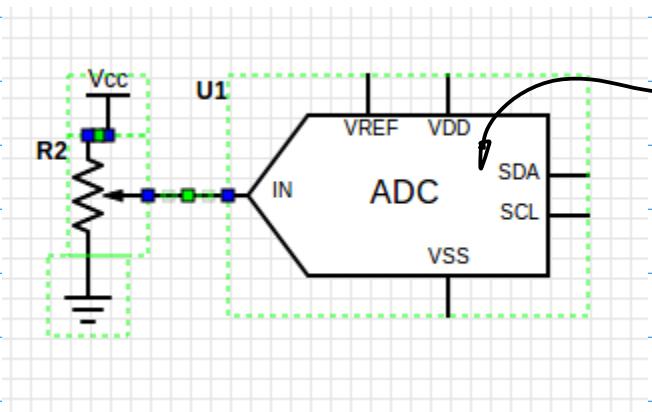
Clock Management

Connectors

Converters

Analog to Digital

- Resolution: 10-bit/12-bit
- Differential Nonlinearity:  $\pm 2.0$  LSB
- Integral Nonlinearity:  $\pm 4.0$  LSB
- Maximum Conversion Rate: 1MSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Analog Input Range: 0 – 3.3V
- On-chip sample-and-hold function
- Normal Conversion Mode
- Separate X/Y position conversion mode
- Auto(Sequential) X/Y Position Conversion Mode
- Waiting for Interrupt Mode
- STOP mode wakeup source



Suppose for 10bits ADC Configuration

- Resolution: 10-bit/12-bit
- Differential Nonlinearity:  $\pm 2.0$  LSB
- Integral Nonlinearity:  $\pm 4.0$  LSB
- Maximum Conversion Rate: 1MSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Analog Input Range: 0 ~ 3.3V
- On-chip sample-and-hold function
- Normal Conversion Mode
- Separate X/Y position conversion Mode
- Auto(Sequential) X/Y Position Conversion Mode
- Waiting for Interrupt Mode
- STOP mode wakeup source



$$b_9 b_8 \dots | b_1, b_0$$

6      { +3 (=11)  
 -3

$f_{Sampling} \geq 2f_{max}$

error due to Nonlinear Behavior.

"Sampling"  
Nyquist Theorem

$$f_{Sampling} \geq 2f_{max} \dots (8)$$