March 16 (Wed)

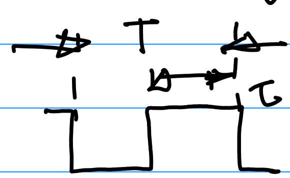
Topics: 1° PWM Architecture/Hardware Aspect — Waveforms, Timing Diagrams, SPRs

2° LSM303 Sensor I2C

Example: PWM Discussion.

{ f_{PWM}
{ Duty Cycle

One period



$$\text{Duty Cycle} = \tau / T \quad \ldots (1).$$

Square Wave: D.C. = 50%.

Architectural Aspects:

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

6410X_UM

P1106

32 PWM TIMER

This chapter describes the functions and usage of PV

C. Prescaler for f_{PWM} Config

8 bit

PP1107

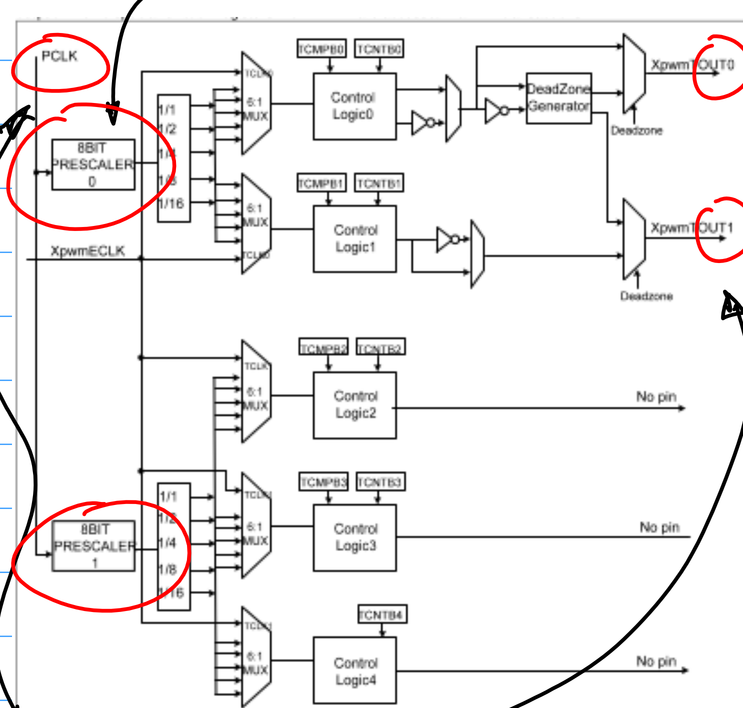


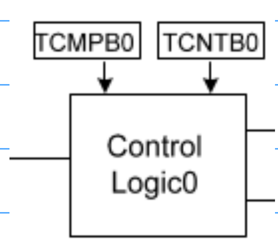
Figure 32-1. PWMTIMER Clock Tree Diagram

a. Input: PCLK peripheral clock.

1/2, 1/4, 1/8, ..., of the System Clock

b. Output (2 outputs)

d. Special Purpose Registers



CONF (Configuration Register)

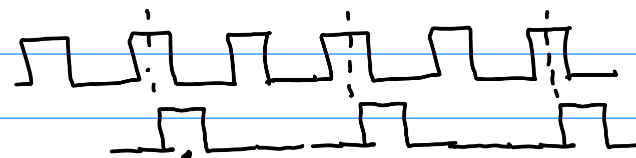
CNT (Control — "Count")

CMP (Comparision)

define f_{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) → f_{PWM} (for interger Only)
then, use Another Counter to get Duty Cycle.

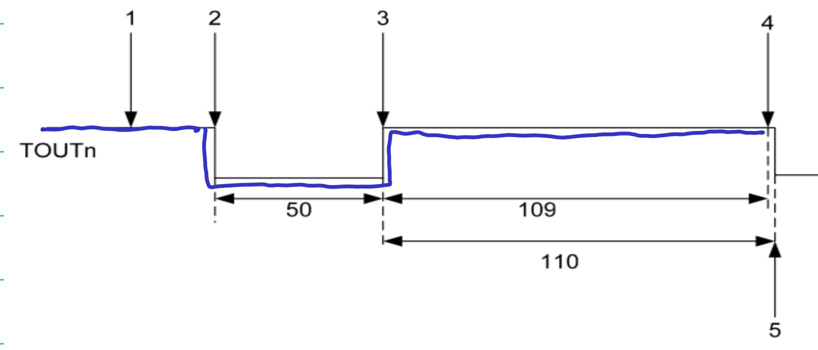
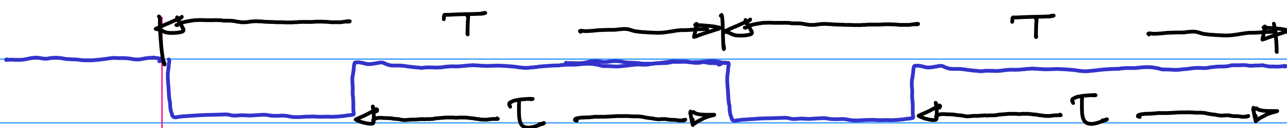


Figure 32-2. Simple Example of PWM Cycle Block Diagram



$T = 1/f_{PWM}$ ... (2), $f_{PWM} = \dfrac{PCLK}{(\text{Prescaler}+1)\,DIV}$ ... (3)

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

Suppose $PCLK = \frac{1}{4}(\text{System Clock}) = \frac{1}{4}(800\times10^6)$

$= 200\times10^6$

Defined By Special Purpose Register.

Design Guidelines for PWM:

① CNT (TCNTB0) Define f_{PWM}

Control Count — Timer — Buffer0 for PWM0

② CMP (TCMPB0) Defines Duty Cycle

Comparision

③ CONF Configuration Register. Defines f_{PWM}

Example: Suppose $PCLK = 500\times10^6$ (MHz), Find the Counts for TCNTB0 (N)

Suppose to Drive A Stepper Motor $f_{PWM} = 2\text{ KHz} = 2\times10^3$

$\dfrac{PCLK}{N} = f_{PWM}$ ... (4)

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

$\frac{500\times10^6}{N} = 2\times10^3$,

$\therefore N = \frac{500\times10^6}{2\times10^3} = 250\times10^3$

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 Registe two 8-bit Prescaler and Dea
TCFG1	0x7F006004	R/W	Timer Configuration Registe and DMA Mode Select Bit
TCON	0x7F006008	R/W	Timer Control Register
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Regis
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Re
TCNTO0	0x7F006014	R	Timer 0 Count Observation
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Regis

2^{32}

32.4.1.4 TCNTB0 (Timer0 Counter Register)

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

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 $M = (D.C.) \times N$ ...(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 $f_{PWM} = 1000$ Hz, to Drive Stepper motor Controller, in addition, Duty Cycle is 30%.

Find: (1) $TCNTB_n = ?$; (2) Find TCMP

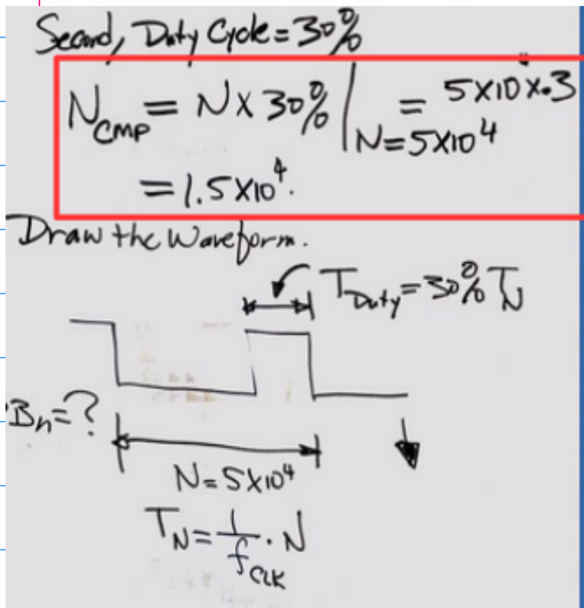
TCMPBn

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

$\frac{50\times10^6}{N} = f_{PWM}$, ...(1) $f_{PWM} = 1000$

$N = \frac{50\times10^6}{1\times10^3} = 5\times10^4 \to$ Hex

TCNTn



March 21 (monday)

Midterm on 23rd (Wed).

1 hr. Exam, + 15 min.

Review on Midterm.

3 Questions.

1. A question on Basic Concepts.

1. CPU Architecture
   32 Bit Architecture

a. Memory map. Banks,

b. GPP/IO Peripheral Controller.

c. SPRs. Naming, functions.
   GPxCON, GPxDAT
   Tech. Spec ⇄ Binary Pattern

d. ARM11 Reference Code
   User Space, Kernel Space

e. Target Platform, NAND,

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 → CPU Datasheet

g. GPIO Testing. { I/P Testing, O/P testing.

Ref: 2022S-101-notes-cmpe242-3-14.pdf

PP.9 CKT → Pin Selection

gpio79 (pin 12)

gpio78 (pin 40)

2. Question(s) on PID Controller Design.

{ Hardware, Software } Motor Drive GPP/PWM

Motor Drive Pin Connection Requirements

SCH., Connectivity Table.

"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 I2C.

3. Theoretical Aspects of the PID Controller Design.

a. PID Block Diagram.
P~, PI~, PD, ... etc.
D (erivative Controller) Short Future.
I (ntegration Controller) History

b. Computation. Forward Difference, Backward Difference, Central Difference. Kernels, Computation.
Integration Controller. History Back N Steps.

c. Sensor Interface Hardware I2C I/F.

LSM303 I2C Based Sensor.

Example: An Application of LSM303

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

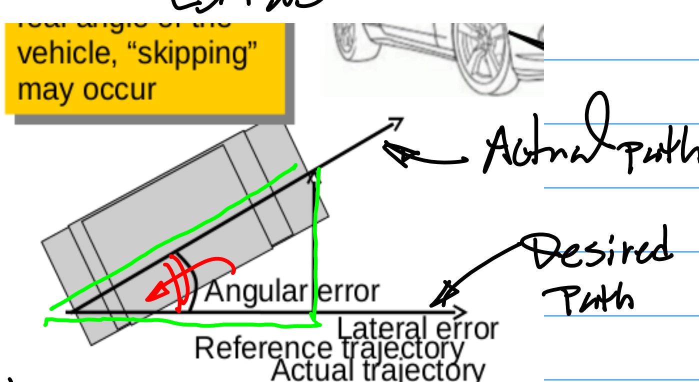


Fig 1.

α : from LSM303
$\vec{P_A}$: is measurable

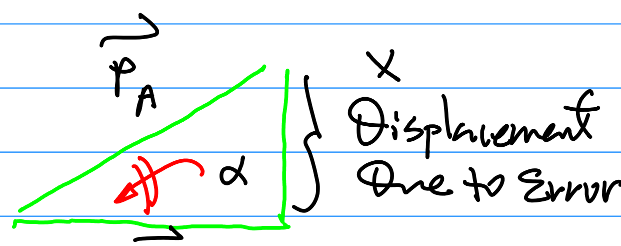
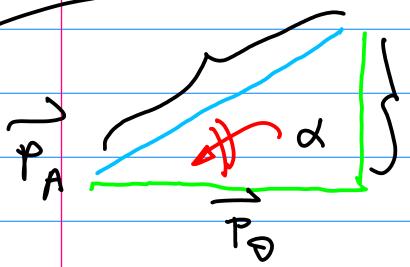


Fig. 1b

CmpE242 March 21, 22

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



Displacement Due to Error

Fig. 2

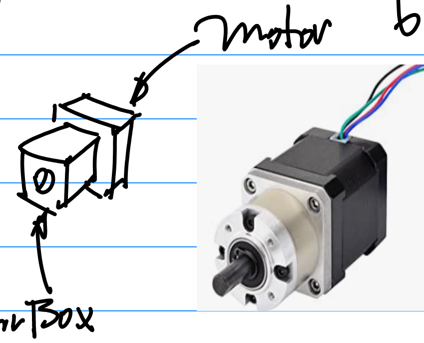


Fig 3.

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.

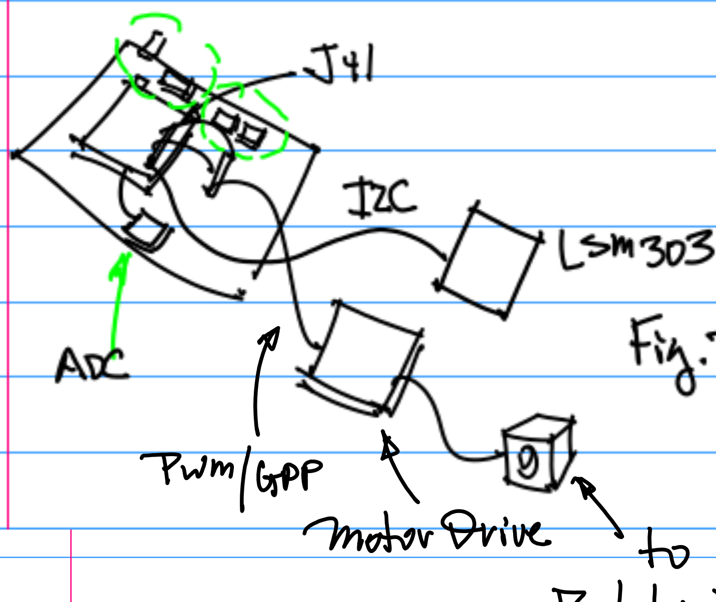


Fig. 2.

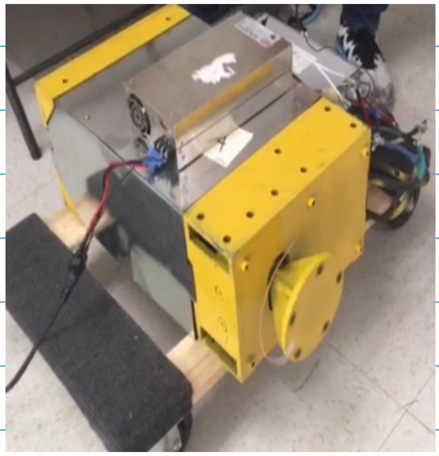
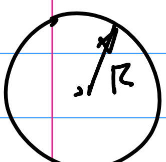


Fig. 4a, 4b With Reduction Ratio Gear Box.



the wheel of the Robot has the dimension $R = 100mm$

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

Cmpe242 March21,22

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

April4 (monday)

Topics: 1° I2C Interface for LSM303 Sensor Integration.

Ref:

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

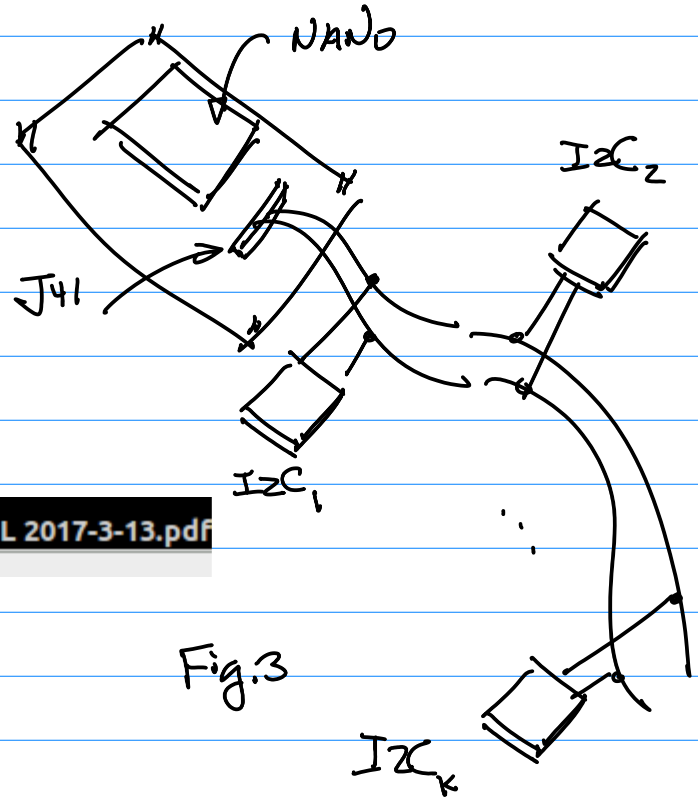


Fig.3

I2C Hardware Features
- Protocol Definition
- Coding Implementation

Hardware Features:

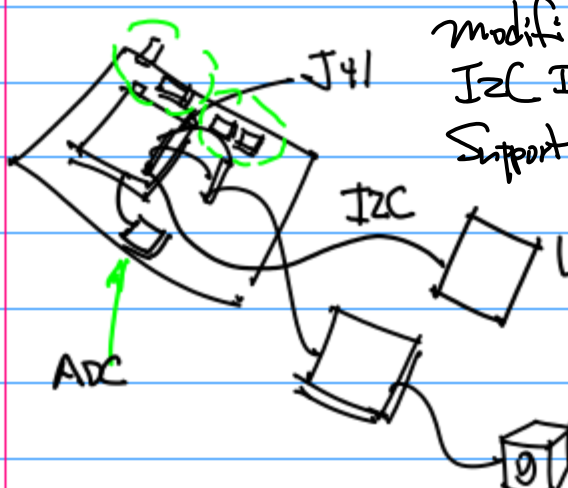
1. Pins
   - SDA: Bi-Directional Data Pin. (Signals in Both Direction)
   - SCK: Serial Clock.

   Speed: ~4 mbps, ≤ 10 mbps

Sub-Address: 7 bits, $2^7 = 128$ → 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)

2. I2C Bus



Modification of I2C Interface to Support K Devices

LSM303

Fig.2.

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

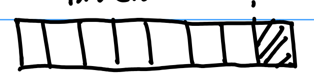
Table 11. Transfer when master is writing one byte to slave, pp 20, datasheet

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

0X3D Table 16

0X3D Table 17

Fig 4.



7 bits for the Address "Slave"

Directional Bit:

{ WRITE: 1
{ READ: 0

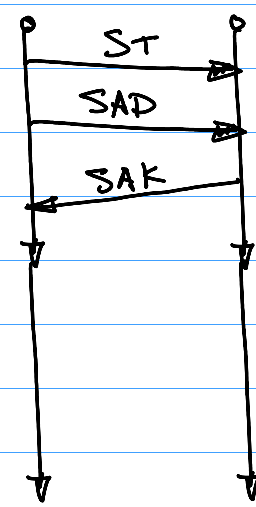
→ time

a. On SDA Line
b. from the master
c. 7 bits for SAD (Salve Addr)

Note: Space-Time Diagram.

→ Space Spatial

Target (NAND) "Slave"
I2C Master LSM303



Time

Fig. 5

Note: 1° ST Start
2° SAD Slave Address
3° SAK Slave Ack.

And at t_3: SUB Sub-Slave-Addr.
to Address the unit inside the Slave Device.

Question: SAD for LSM303
a. b. SAD[7:1]

Pp. 20

Table 14. SAD+Read/Write patterns

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

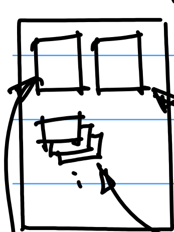
0011001X

3 { (1) 0011 Read : 3
{ (2) 0010 Write : 2

0X33 Read
0X32 Write

Fig 6

LSM303



"I2C Slave"

meg. Sensor

Special Purp. Registers

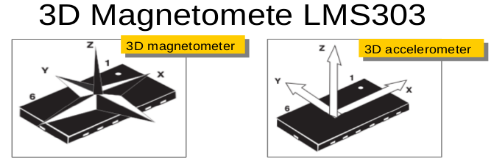
MCU: microprocessor
FSM: Finit State machine

SPRs: Config & Control, Data Sensing

Example: Find Sub-Addr.

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

3D Accelerometer and
3D Magnetomete 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"

For magnetic Sensor.

Table 16. SAD

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): 30Hz
Find 1° Special Purpose Register Responsible to perform Configuration.
2° Find Binary Pattern to initialize the SPR.

April 7. Wed

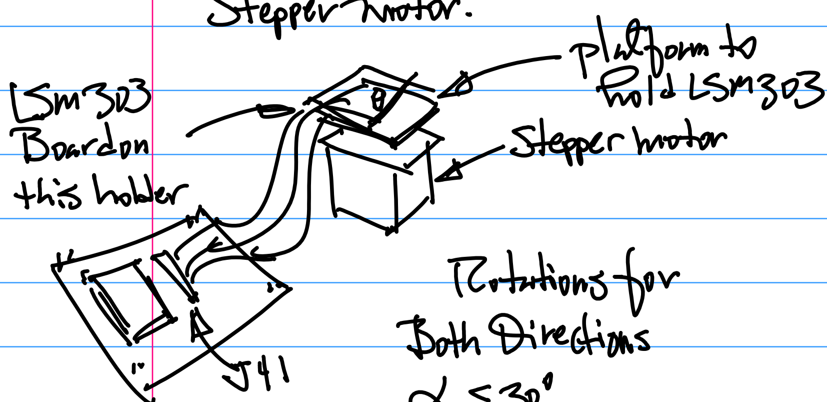
Topics: 1° LSM303 Interface Design.
Project Due April 18 before Class.

CMPE242 April 7, 22

Project Requirements:

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

2. Integrate LSM303 Sensor to Stepper motor.



Rotations for Both Directions
$\alpha \leq 30°$

3. to Run Stepper motor program to Actuate 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. 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, 5 ~ 10 pulses. when collecting Data

6. 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 16 For mag. 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.

			0	0	0	0	0

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.

b. For Mag. Sensor

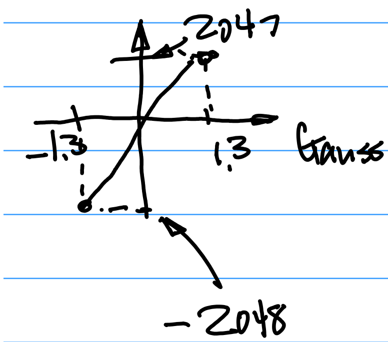
Smaller Range / more Sensitive

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	±1.3	1100	980	0xF800–0x07FF (-2048–2047)
0	1	0	±1.9	855	760	
0	1	1	±2.5	670	600	
1	0	0	±4.0	450	400	
1	0	1	±4.7	400	355	
1	1	0	±5.6	330	295	
1	1	1	±8.1	230	205	

c. Out Range

$Z'' = 1024 \times 2$
$= 2048$



Larger Range / Less Sensitive

GN2=0, GN1=0, GNϕ=1

Note: Start with either default setting of the Gain, or Set the gain to GN2 GN1 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 Register(s), follow this "Write" Pattern:

SPR's Address: 0x01 for GAIN Setting, CRB_REG_M
0x02 for mode MR_REG_M,

Binary Pattern

For Reading After the Configuration, we will need find SAD+R (Sub Addr. for Read), Example is given in PPT, from Datasheet, Table 15, pp:20.

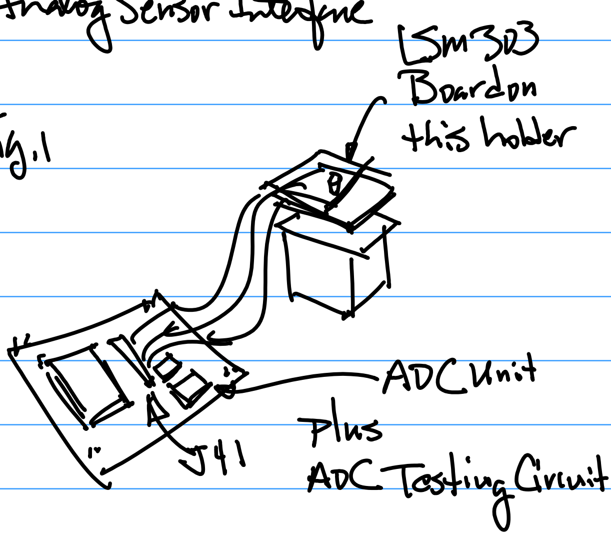
Coding Implementation: C/C++ Option
Python Option.

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

April 11 (monday). ADC

Analog To Digital Conversion.
Industrial IOT Application.
Analog Sensor Interface

Fig. 1



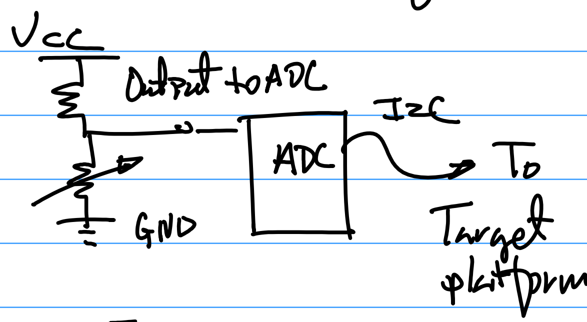
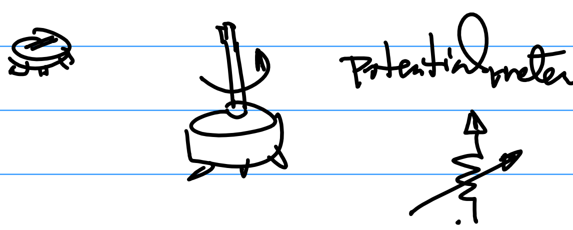
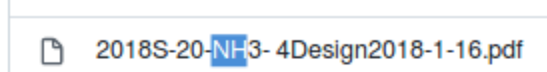


Fig. 2.

Example: Background of Analog to Digital Conversion.

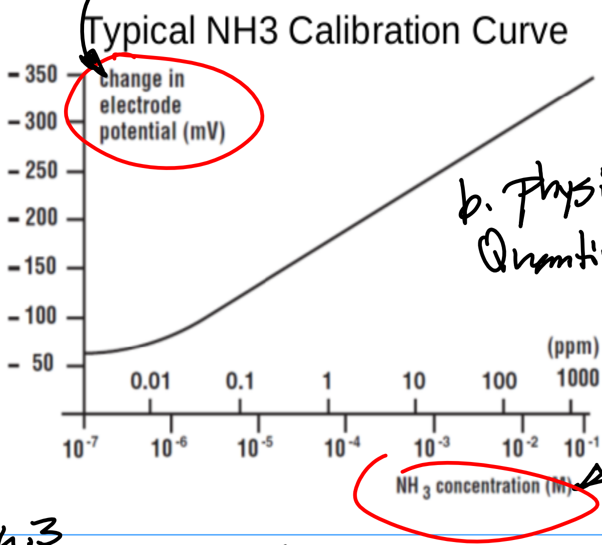
Sensor (Analog) Input,
Analog Current





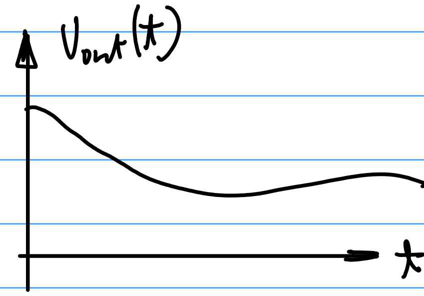
CmpE242 April 11, Monday, 22

Note: a. Analog Change Output



b. Physical Quantity

Fig.3



1° Sampling ADC.

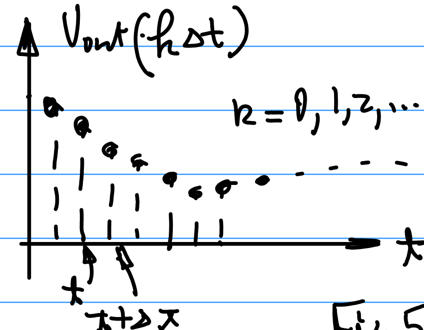
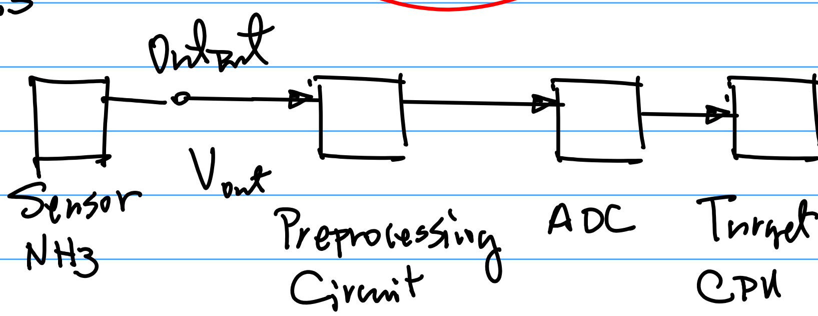


Fig.5



2° Quantization

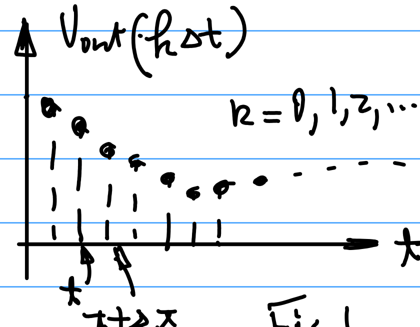
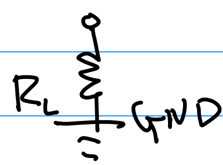
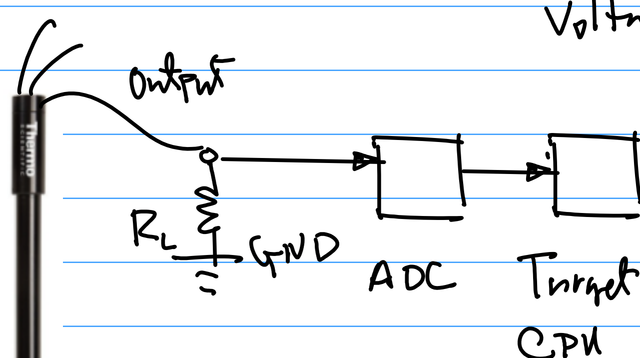


Fig.6



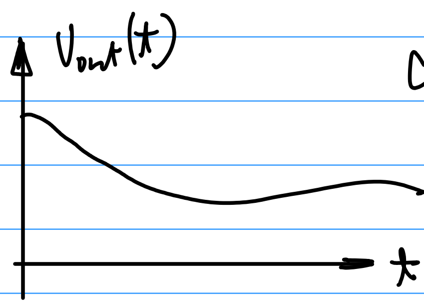
Conversion: Current i to Voltage v.



Note: the magnitude after the quantization becomes one of 2^x possible levels.

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

Fig4.



Continuous from the Analog Sensor.

Performance of ADC:

1. Sampling Rate:
(Common) 500KSPS
(Samples per Second)

Sampling Theorem (Nyquest ~)

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

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)

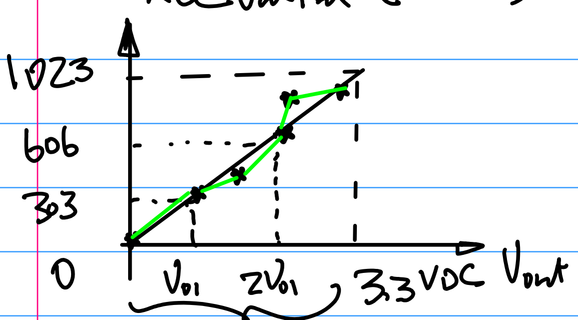


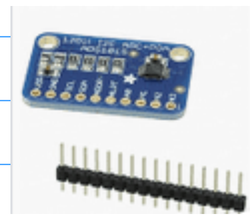
Fig.7

Fig.7 ADC Input Range (Dynamic ~)

$$\frac{\Delta Output}{\Delta V_{out}} = k \text{ (Slop of the Line in Fig.7)}.$$

We have to measure "Linearity" of a given ADC.
Homework (After the project of LSM303)
Formal Due Date to Be Announced in the next lecture.

1° Identify & Purchase Your ADC module.



ADC with I2C Interface, ADS1115.

2° Generate Schematics.

Start with Symbols

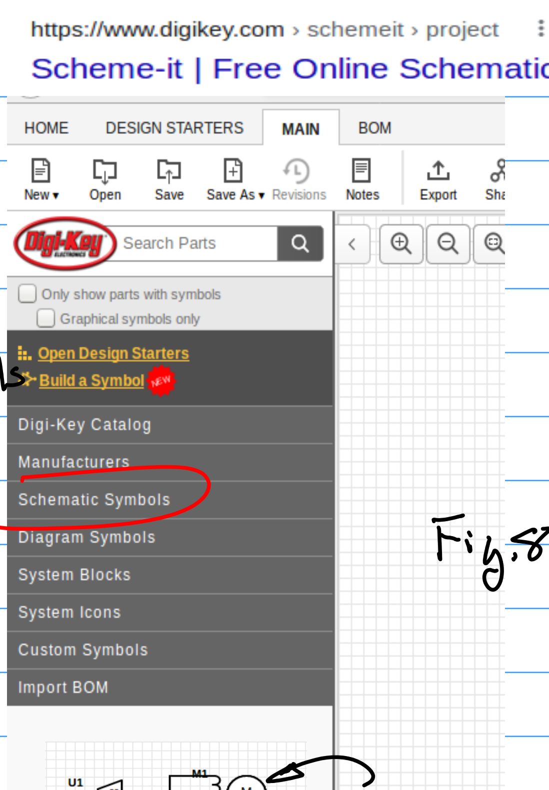
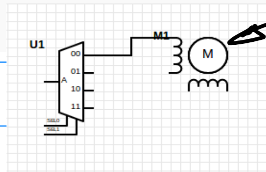


Fig.8



Example here with a stepper motor

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

$(V_{01}, D_1), (V_{02}, D_2), \ldots (V_{010}, D_{10}),$

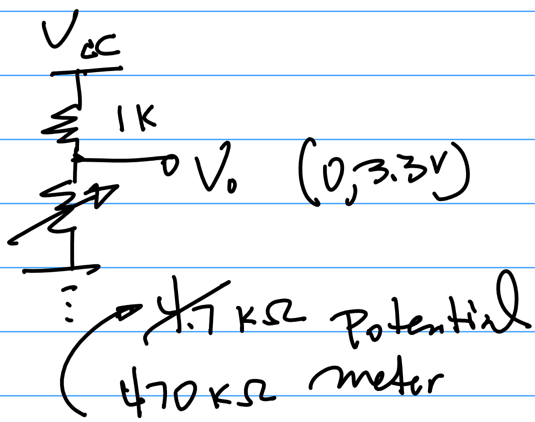


Fig.9

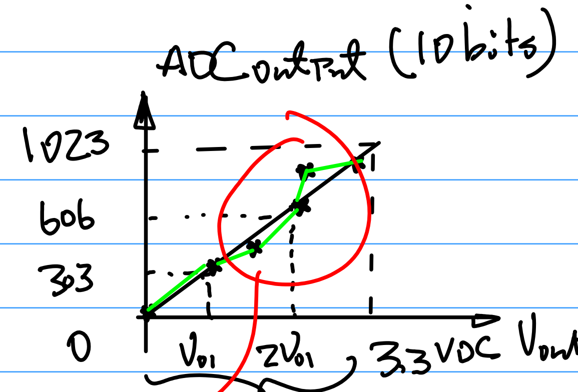
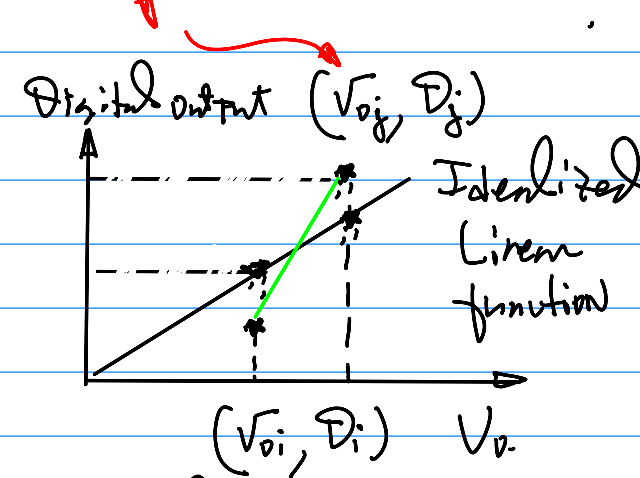


Fig. 11 ADC Input Range (Dynamic ~)

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

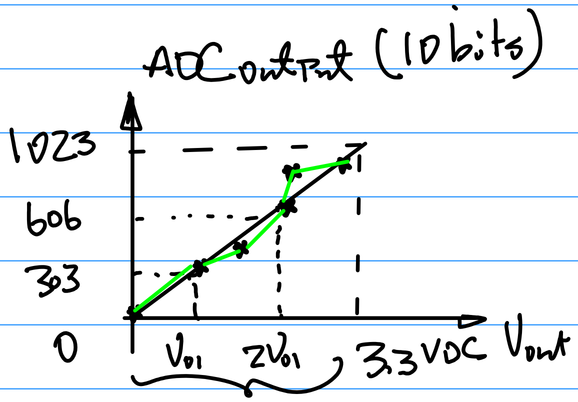


Fig.10 ADC Input Range (Dynamic ~)

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

6. Correction of Non-Linear Behavior