CMPE242 Spring2023 PART II

April 3rd (Monday)

Road map for the 2nd half of the Semester:

IIOT (Industrial IOT):

ADC Data Validation → (1) F.F.T. (Fast Fourier Transform)

Analog Sensor interface Design.

(2) Power Spectrum Technique.

(3) Hardware Architecture Aspects

Ion Selective Electrode Sensors { Many Applications in different Industry Sector.

Homework Extension Next Monday with Demo.

Project: Due the End of the Semester.

Implementation (1/3) 33%
- PID.
- I2C Sensor
- PWM Motor Control
  - Pre-processing CKT.
  - ADC

Research Part (1/3):
(1) P.P.T Presentation on State-of-the-Art Technology in the Embedded world.
(2) Report (GuideLine)
(3) Proposal (No more than One page), Submit to the CANVAS for Approval.
— By Wednesday / Monday Next

(1/3) Demo & Presentation: By the end of the Semester.

Example: for Analog ISE Sensor interface Design.

Ref: → 2018S-20-NH3- 4Design2018-1-16.pdf

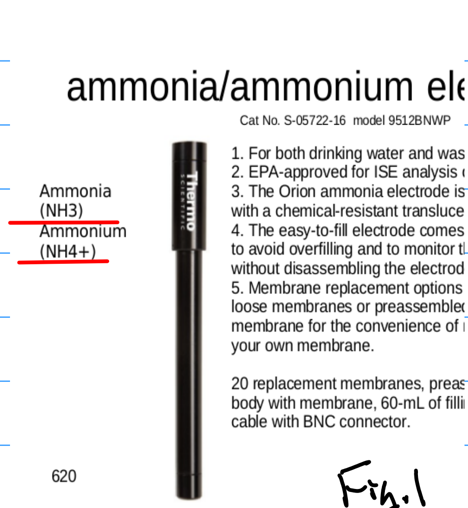


Fig.1

google Definition: (1)

Principle of ion-selective electrode (I.S.E.) An ideal I.S.E. consists of a thin membrane across which only the intended ion can be transported. The transport of ions from a high conc. to a low one through a selective binding with some sites within the membrane creates a potential difference.

→ Example of A Battery

ISE Sensor

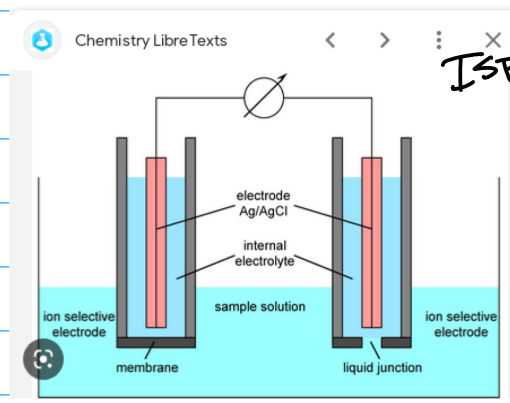


Fig.2

CmpE242

Spring 2023

Working Principle of Battery - Electrical E... Visit

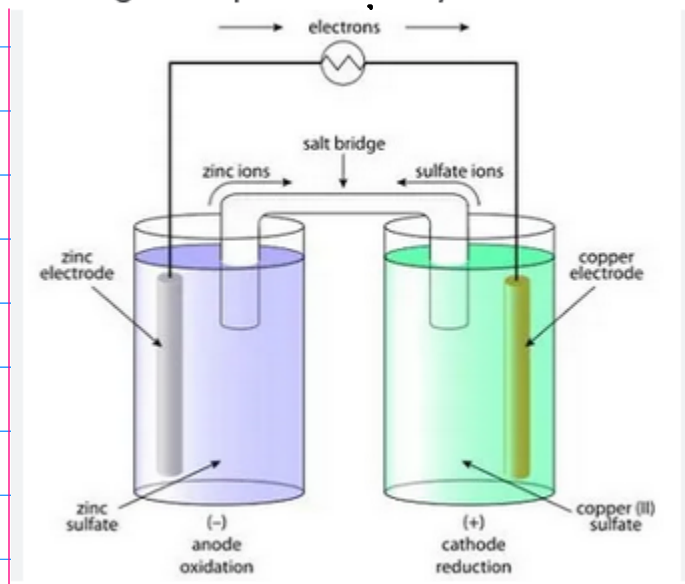


Fig. 3

Observation: Use Battery As An Example to Demonstrate Ion Selective Electrode Sensor. See NH_3/NH_4^+ sensor in Fig. 1.

Characteristic

Typical NH3 Calibration Curve

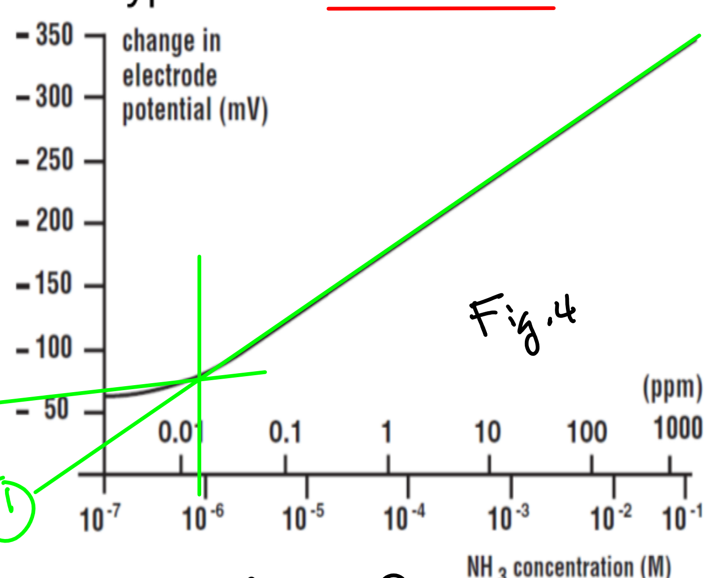


Fig. 4

Note 1. We like to have the Linear Characteristics from the Calibration Curve, Such as [1, 10], [10, 100], [100, 1000], etc.

Note 2: For the Non Linear Part, Let's Perform Linearization — By using Piece-wise Linear Lines.

$\begin{cases} \text{Piece-wise Line 1.} \\ \text{Piece-wise Line 2.} \end{cases}$

Next Step is to formulate Each Line by using Linear Equation.

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1} \quad \ldots (1)$$

Solve for $y = bx + c$ (See the Previous Notes).

With Simplification By Removing Very Low Concentration Part, we have

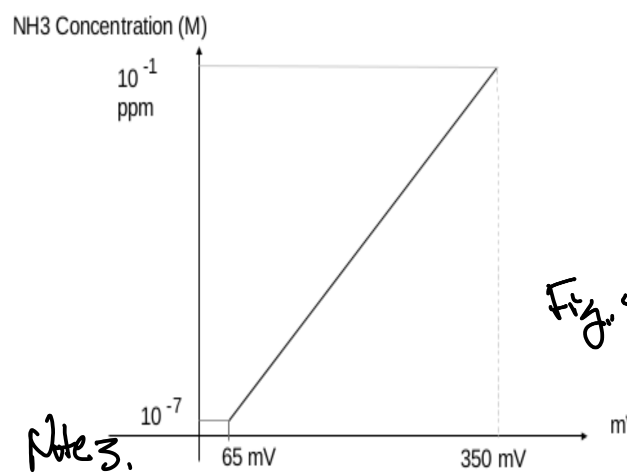


Fig. 5

Note 3. Then, Change the Cal-Curve to the Characteristic Curve e.g. Horizontal Axis is voltage for the Design of interface.

Industrial Analog Sensor Interface Protocol:

1° Output has to be defined by Current. 2° The Range 4~20mA.

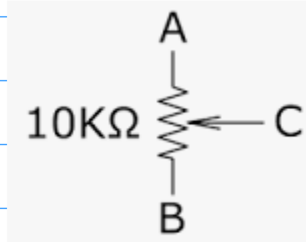


Fig.1

Control.com
https://control.com › Technical Articles

Why is 4-20 mA Current Used for Industrial Analog Sensors?

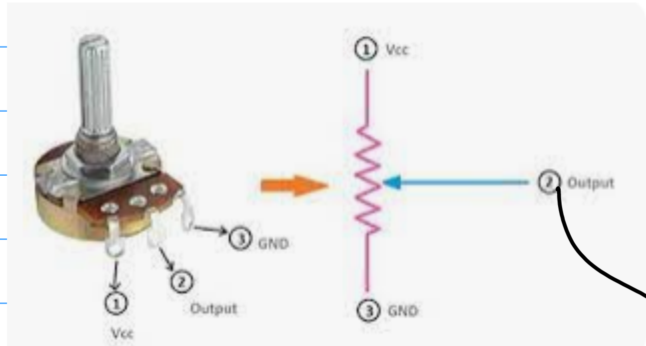
Note 4. Fit to the Dynamic Range of your target platform.

↓

External ADC Needed

CMOS Range ADC [0, 3.3]

TTL " " [0, 5.0]



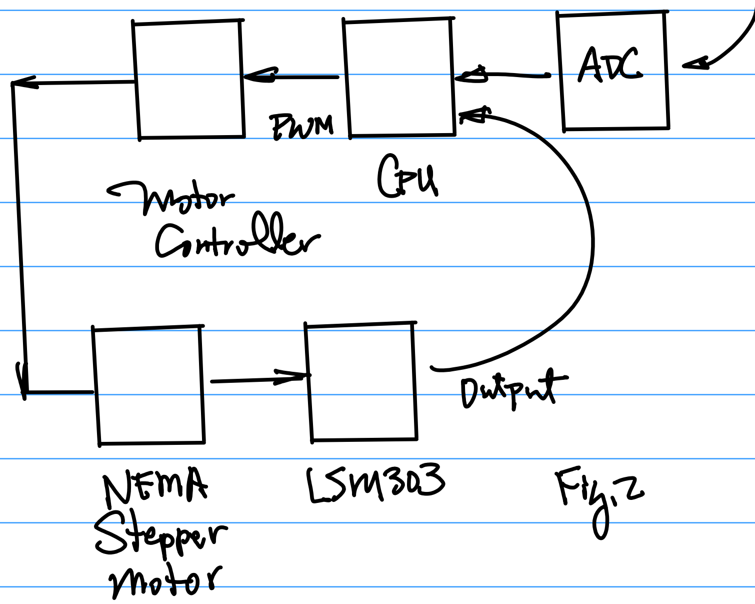


Fig.2

https://www.adafruit.com/product/1083

April 5th (Wed)

Note 1° New updated Due Date for the Motor Control, please use PWM from your target platform as the input to the motor controller.

2° ADC Unit for the project Selection Guide.

1° Interface protocol I2C; ~4 mbps

2° 0.5~2 MSPS (Million Samples per Second)

3° 10~12 bits per sample.

Example: P.O.T.

ADS1015 12-Bit ADC - 4 Channel with Programmable Gain Amplifier - STEMMA QT / Qwiic

Product ID: 1083

$9.95

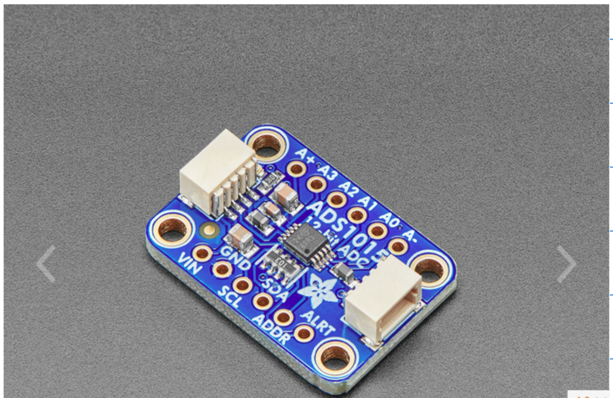


Fig.3

Note: Input Voltage Range

TEXAS INSTRUMENTS

ADS1013
ADS1014
ADS1015

www.ti.com

SBAS473C – MAY 2009 – REVISED OCTOBER 2009

ELECTRICAL CHARACTERISTICS

All specifications at –40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted. Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1013, ADS1014, ADS1015 MIN	TYP	MAX	UNIT
ANALOG INPUT					
Full-scale input voltage(1)	$V_{IN} = (AIN_P) - (AIN_N)$		±4.096/PGA		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = ±6.144V(1)		10		MΩ
	FS = ±4.096V(1), ±2.048V		6		MΩ
	FS = ±1.024V		3		MΩ
	FS = ±0.512V, ±0.256V		100		MΩ

Example: Design Objective.

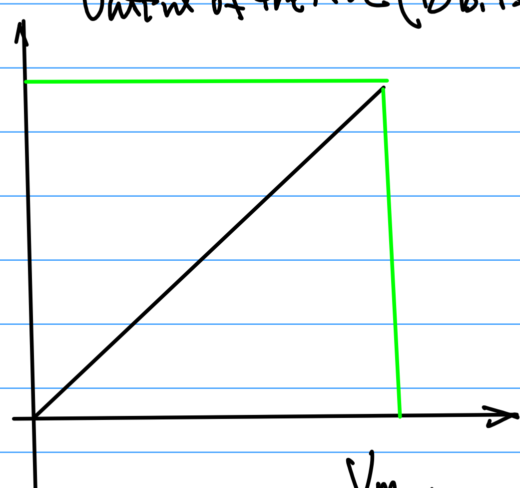
1° Selection of An ADC.

Construct the Characteristic Curve of the ADC

$V_{max} = 3.3V$ for CMOS

Output of the ADC (10bits)

1024



Vin

Vmax

Input to ADC

Fig.4

2° Design Objective: To Design A pre-process unit to make the Analog Sensor Output match to the ADC input dynamic Range.

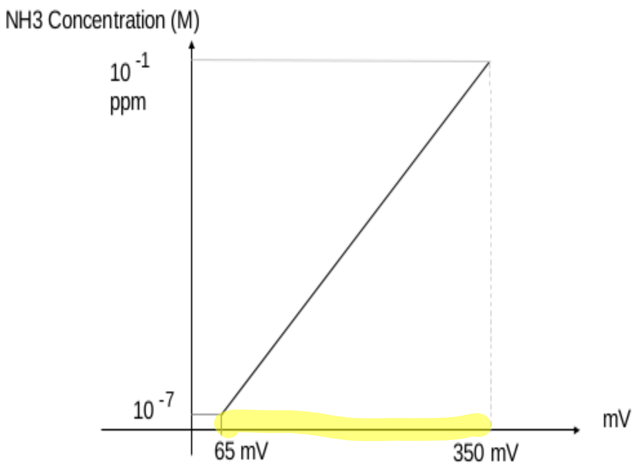


Fig.5

Step1. 65mV → 0

350mV → Vmax

In general,

$V_{s,min} \to V_{ADC,min}$ ...(1)

$V_{s,max} \to V_{ADC,max}$ ...(2)

April 10 (Mon).

Example: Continuation of the Preprocessing Design.

Note1. Check 1015 ADC Dynamic Range for the Input. [0, 3.3V]. Verification is Needed

Theoretical Analysis:

Step1. Provide "offset" to shift The Sensor dynamic Range, Subtraction Can be utilized for this purpose. e.g.

$V_{sen} - V_{offset}$ ... (1)

which will Lead to the Result Below.

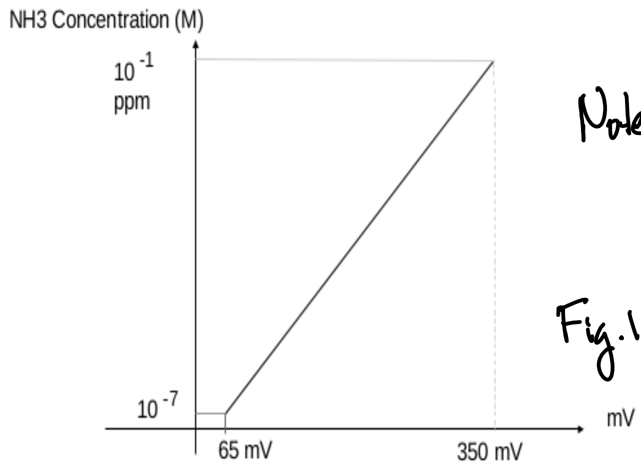


Fig.1a

Note1. For more generalized Case, Let
$V_{min} = 65mV$, $V_{max} = 350mV$.
So, the offset $= -V_{min}$.
then, the Upper Bound after offset is

$V_{max} - V_{min}$

↓ Subtraction/offset

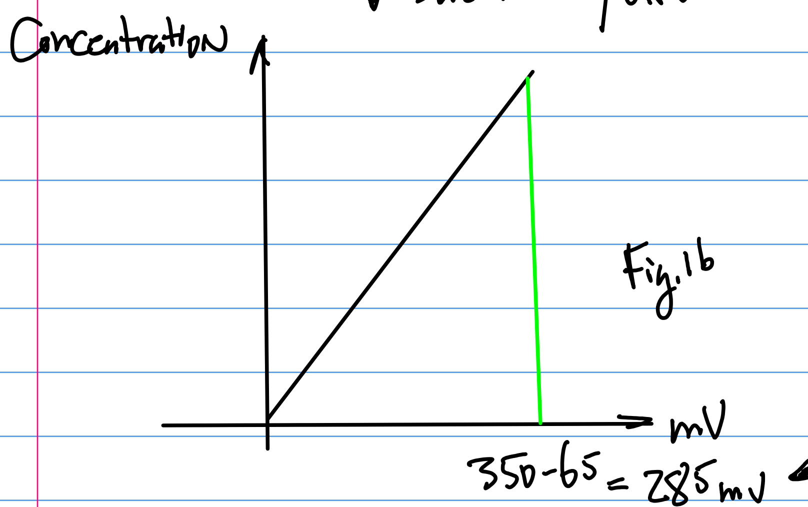


Fig.1b

Step2. To magnify the Sensor Output Range to Match the entire Dynamic Range of the ADC.

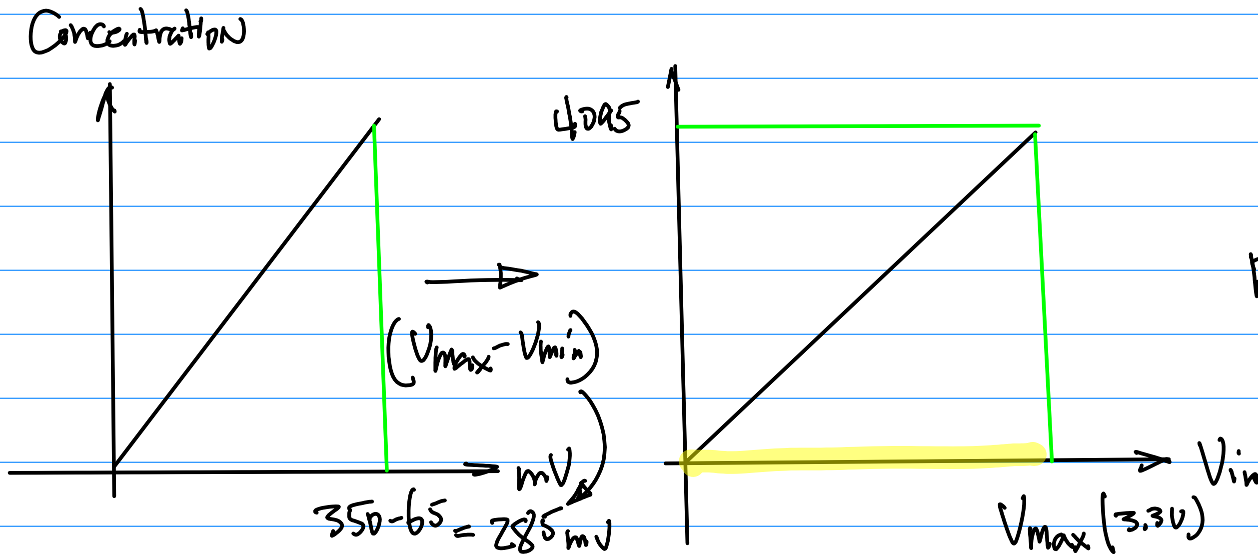


Fig.2

Find the Gain for the Magnification

$$A = \frac{V_{\text{Output Range}}}{V_{\text{Input Range}}} = \frac{3.3}{285 \times 10^{-3}} \cong 11.58$$

where 3.3VDC is from 1015 ADC for Example.

Example: Hardware Design for the pre-processing.

Ref.

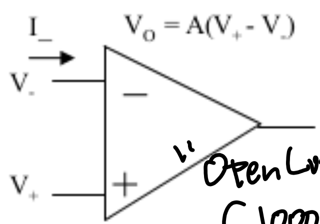
1-0lecture10_OpAmp Circuits.pdf –

Note1: Using OpAmp for Pre-processing Not for the Buffering.

OpAmp Device As a Buffering Stage

Both Analog and Digital Circuit

Note 2: Background



(1) To protect the previous stage's output signal, which is the input to the next stage, while sampling/connecting the signal to its next stage logic circuit. (2) Unit gain non-inverting OpAmp configuration is an excellent choice.

1000 or Bigger

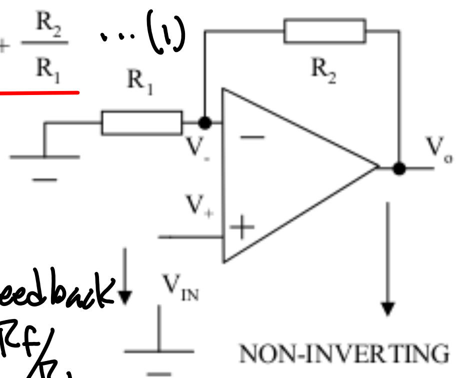
1×10^{-9} A or Smaller

Ideal OpAmp Properties: (1) very large gain, A>>M; (2) draws very little current, $I_- \sim 0$, e.g., very high impedance; (3) $V_O = A(V_+ - V_-)$ is finite range, which leads to $V_+ = V_-$.

V_1 $I \ll \delta$ V_2

for Example 100MΩ or higher.

$$\frac{V_O}{V_{IN}} = 1 + \frac{R_2}{R_1} \quad \ldots(1)$$

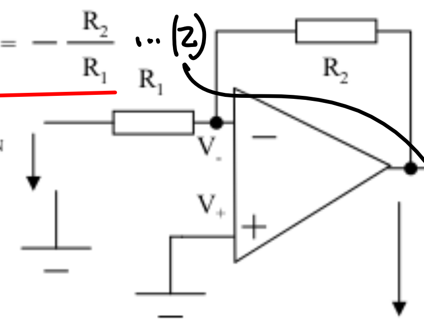


NON-INVERTING

R_2 for feedback

so $A = 1 + R_f/R_1$

$$\frac{V_O}{V_{IN}} = -\frac{R_2}{R_1} \quad \ldots(2)$$



INVERTING

$A = -\frac{R_f}{R_1}$

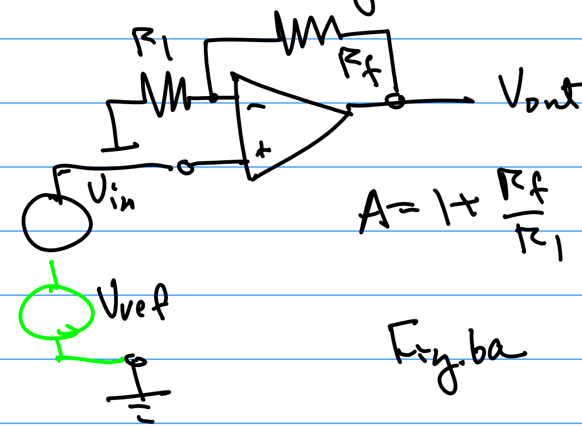
Fig.3

Harry Li, PH.D. SJSU

Now, Consider the design Implementation for 2-Step Process.

Tools { Non-Inverting | Inverting
└ Linear System: Super Impose One Signal (offset) on to the other Signal (Input).

For Non-Inverting



$A = 1 + \frac{R_f}{R_1}$

Fig. 6a

"Black Box"

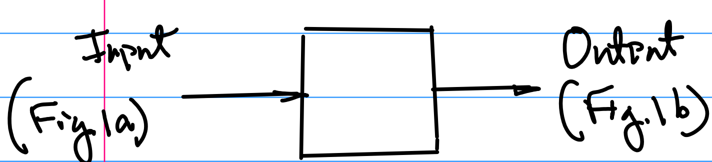
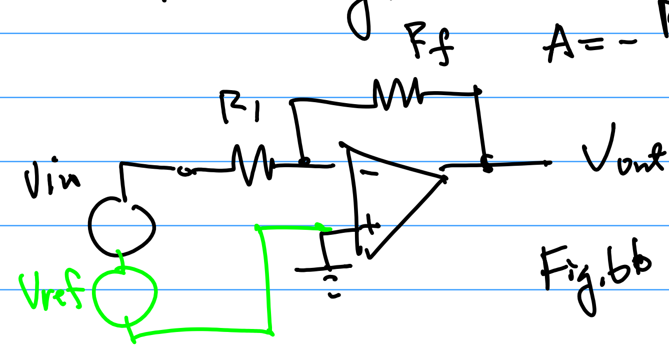


Fig. 4.

Linear System. "Super Impose Signal for Offset"

For Inverting.



$A = -R_f/R_1$

Fig. 6b

Vin.
Input (Fig 1a) $\longrightarrow \Sigma \longrightarrow V_{in}' = V_{in} + V_{ref}$

$V_{ref} = -V_{min}$
$(= -65\,mV.)$

Fig. 5

In the Circuit Design. Just Connect 2 inputs together.

Note: Selection of Non-Inverting v.s. Inverting Configuration Depends on your design Need. This Design is an illustration of Super Imposing an "offset", e.g. V_{ref}.

Now, to Magnify the Signal to match ADC's Dynamic Range. Choose Non-Inverting Configuration (see Fig. 2)

Hence

$A = 1 + \frac{R_f}{R_1} = 11.58$

CmpE242

Spring 2023

Choose $R_1 = 1\ k\Omega$ (10.58 kΩ)

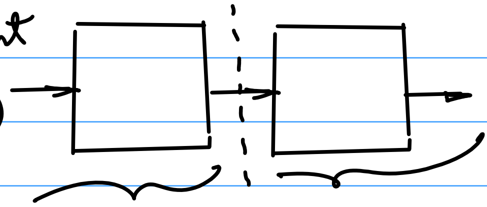
solve for $R_f \cong$ ~~77~~ kΩ ?

please verify it ?

Stage1 Stage2

Box1 Box2

Input (Fig 1a)



offset Gain Amplification

Analog Devices
https://www.analog.com › ltspice-simulator

LTspice Information Center

LTspice® is a powerful, fast, and **free SPICE simulator** software, schematic capture and waveform viewer with enhancements and models for improving the ...

Free for DownLoad, Originated from "Linear" A Silicon Valley Company.

EasyEDA
https://easyeda.com

EasyEDA - Online PCB design & circuit simulator

EasyEDA is a **free** and easy to use circuit design, **circuit simulator** and in your web browser.

Requirements: To Be Able to Run SPICE Simulator.

April 12 (Wed).

Note 1. The Last Project Preparation. (Requires the Semester End Presentation).

Note 2. Implementation of ADC Unit.

P.O.T. 47K or 470K or Similar.

↓

ADC → Target → PWM → Controller

LSM303 ← Motor ←

Note 3. ADC Data Validation

FFT. Power Spectrum.

Note 4. Road Map.

IIOT (Analog Sensors. 4~20mA)

↓

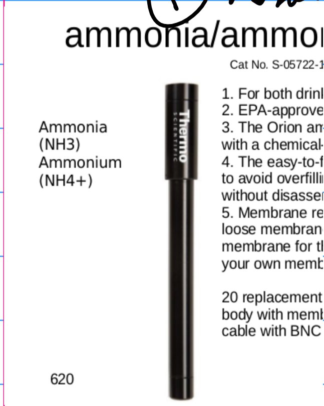
Preprocessing

↓

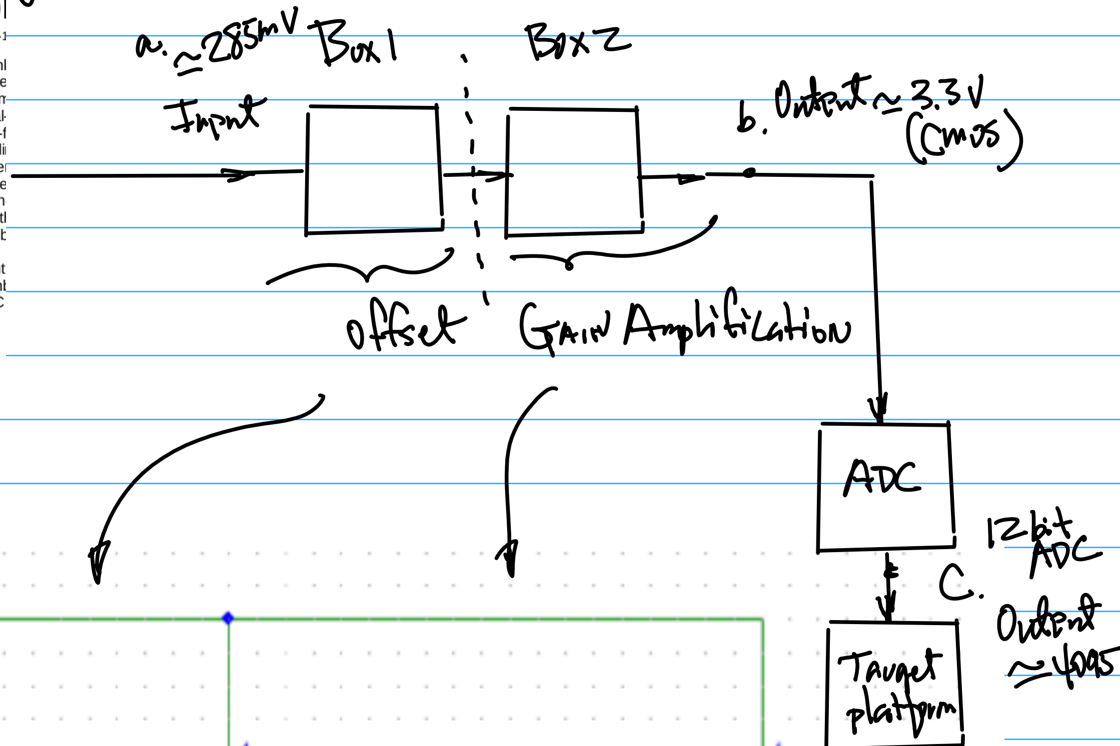
OpAmp. (On-Line Simulation Tool)

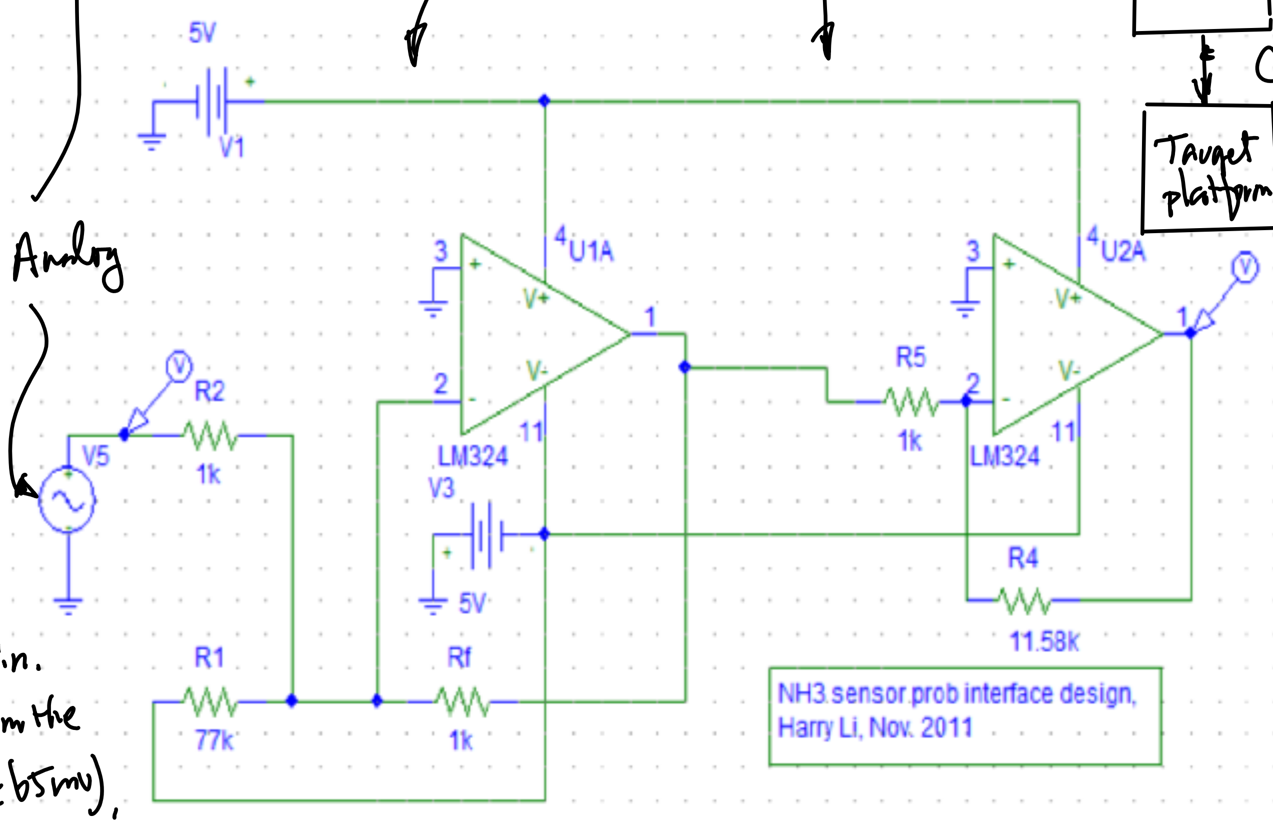
Example:

ISE
① Analog Sensor



Stage1 Stage2



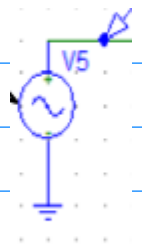


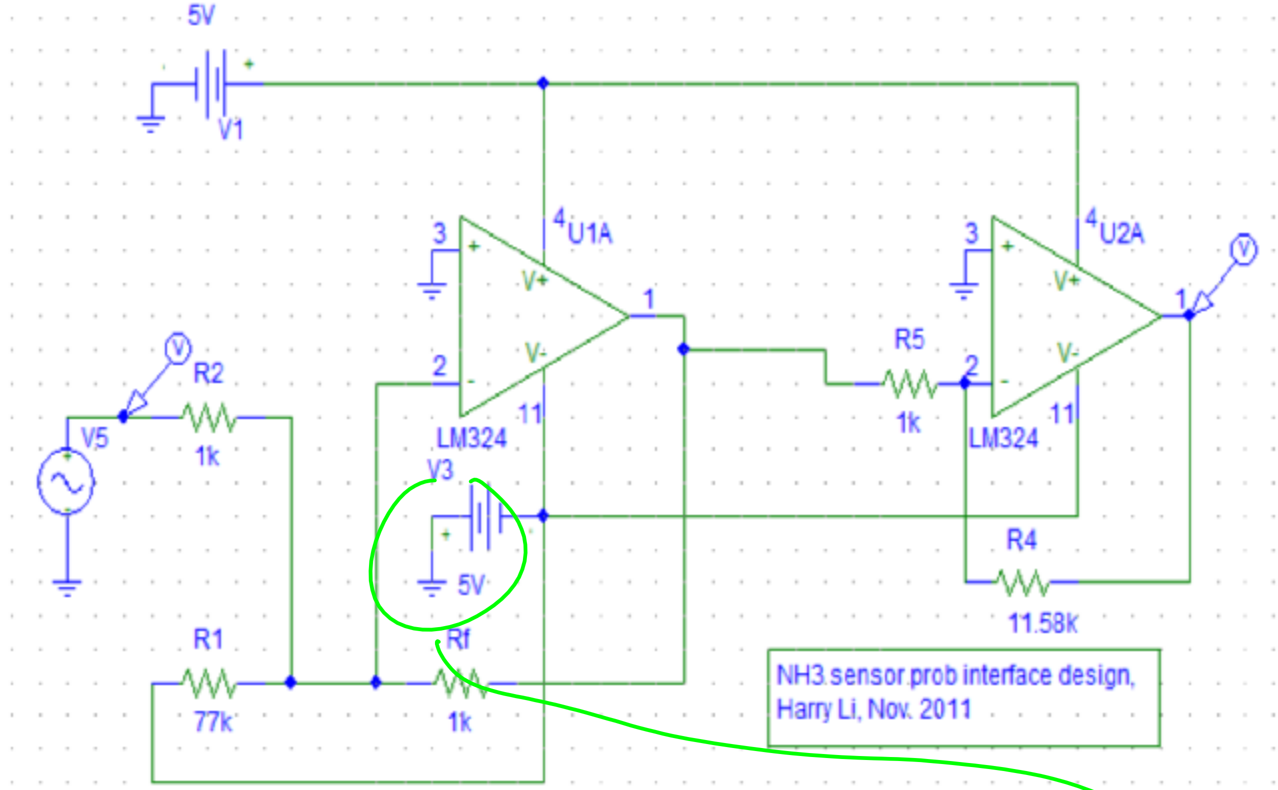
Note: 1° Analog Sensor — Signal Source
Sine Wave Signal

2° for min. Input from the Sensor. (~65mV), we want to get ADC Output = 0.

for max Input from the Sensor, we want to get Output$_{ADC}$: (3.3VDC) 4095

Note: 3. Simulation of the Sensor Output as the input to the pre-processing Circuit.





April 17 (Monday).

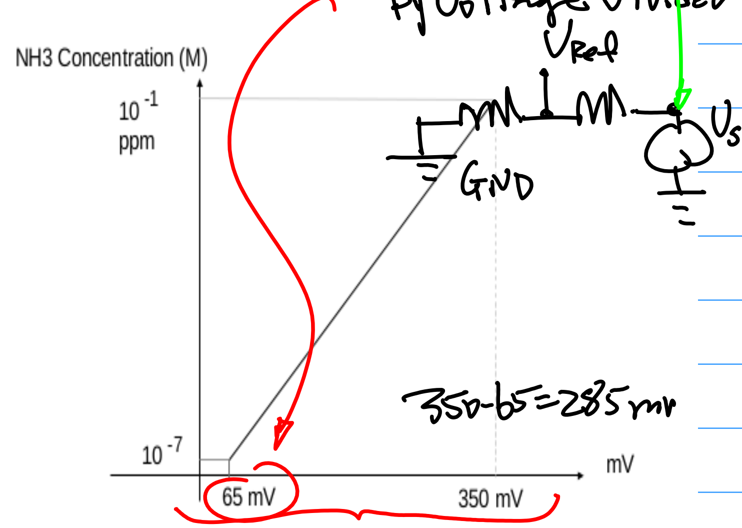
1° Project (Integration of Homework + ADC). Due May 7 (Sunday) plus Research Part / Presentation.

2° Bonus Points (5%) for BLDC Motor Control; 3-phase (u, v, w) motor Control.

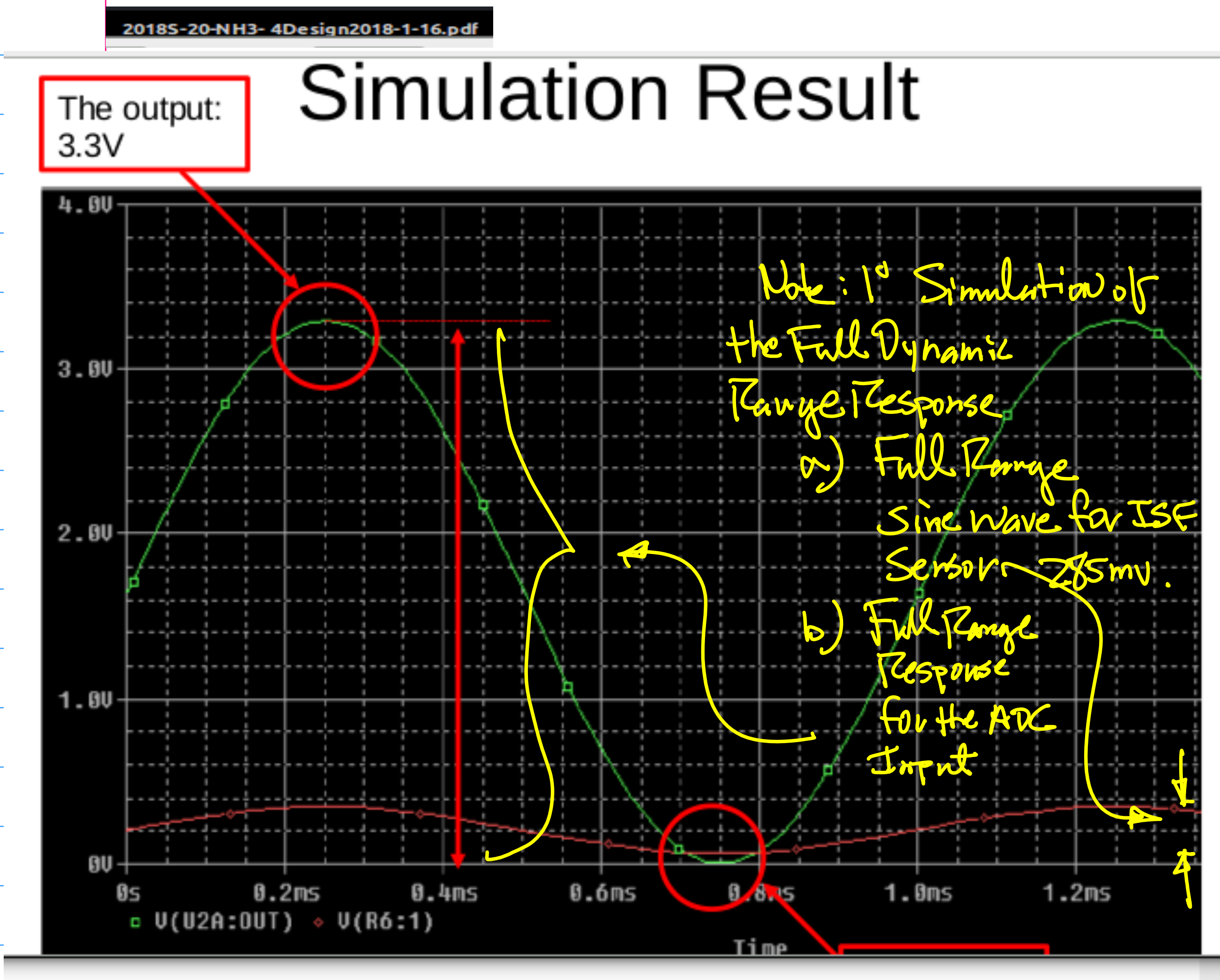
3° SPICE Simulation for

▷ Pre-processing Circuit Simulation.

Example: Continuation. Provide "offset" By Voltage Divider



$350 - 65 = 285\ mv$

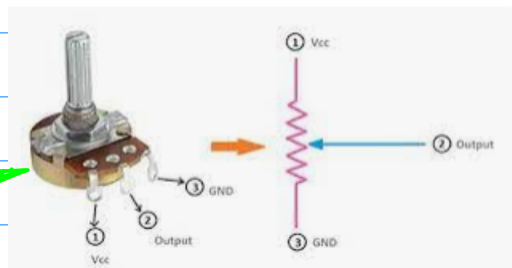


ADC Data Validation

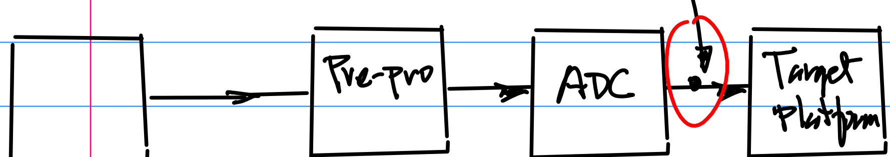
Tool: Fourier Spectrum Analysis.

Data Validation

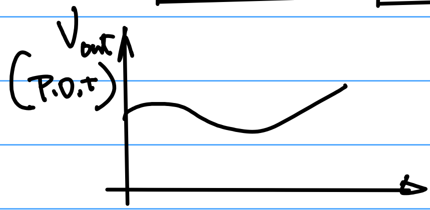
P.O.T. Human Operator

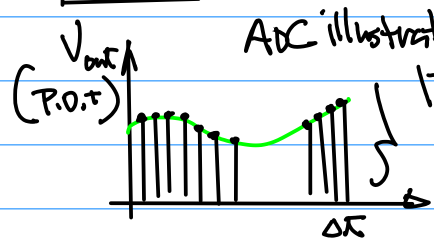


See PP.48 For Details.



ADC Output





ADC illustration → $\{x(n) | n = 0, 1, 2, \ldots\}$

12bit ADC.

$[0, 4095]$

Background / Formulation.

To Validate $\{x(n)\}$, or $\{x(n) \mid n=0,1,2,\dots,\}$

$x(n)$.

D.F.T (Discrete Fourier Transform) is defined as follows.

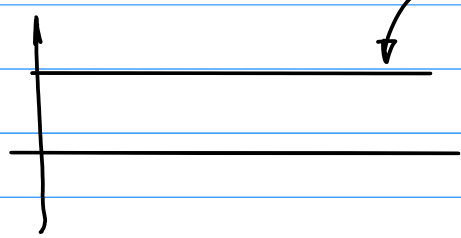
Time Index

$$X(m)=\frac{1}{N}\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi\frac{mn}{N}} \quad \dots (1)$$

physical meaning: $X(m)$, Discrete Fourier Transform.

m: Frequency Index

$m=0$, D.C. index; $X(0)$ D.C. Component.



$m=1$, $X(1)$ Foundamental Frequency Conponment.

N: One Period. Total No. of Points Per a period. Such as $N=1024, 2048, 4096$, etc.

$N=2^x$ for FFT Only. (Fast Fourier Transform)

$$e^{-j2\pi\frac{mn}{N}} = \cos 2\pi\frac{mn}{N} - j\sin 2\pi\frac{mn}{N} \quad \dots (2)$$

Euler Formula

$$e^{j\phi} = \cos\phi + j\sin\phi \quad \dots (3)$$

Power Spectrum of $X(m)$.

$$x(n) \rightarrow \boxed{\text{D.F.T.}} \xrightarrow{X(m)} \boxed{\text{PWR Spectrum}} \rightarrow P(m)$$

Let's Define the Power Spectrum as:

$$P(m) = \sqrt{Re^2[X(m)] + Im^2[X(m)]} \quad \dots (4)$$

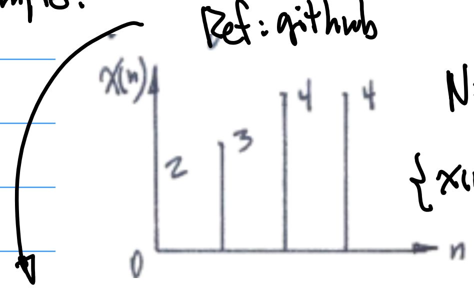
Where

$$Re[X(m)] = Re\left[\frac{1}{N}\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi\frac{mn}{N}}\right]$$

$$Im[X(m)] = Im\left[\frac{1}{N}\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi\frac{mn}{N}}\right]$$

Example:

Ref: github



$N=4$

$\{x(n)\}$

CMPE242-Embedded-Systems- / 2018S-26-1D-DFTv2.pdf

$x(0)=2$, $x(1)=3$, $x(2)=4$, $x(3)=4$.

Find $X(m)$ D.F.T.

$$X(m) = \frac{1}{4}\sum_{n=0}^{3} x(n)\, e^{-j2\pi\frac{mn}{4}} \quad \dots (5)$$

Next. From Eqn (5)

For m=0,

$$X(0) = \frac{1}{4}\sum_{n=0}^{3} x(n)\, e^{-j2\pi\frac{0n}{N}} = \frac{1}{4}\sum_{n=0}^{3} x(n)$$

$$= \frac{1}{4}\left(x(0)+x(1)+x(2)+x(3)\right) \quad \ldots (6a)$$

For m=1

$$X(1) = \frac{1}{4}\sum_{n=0}^{3} x(n)\, e^{-j2\pi\cdot\frac{1\cdot n}{4}}$$

$$= \frac{1}{4}\left(X(0)\cdot 1 + x(1)e^{-j2\pi\frac{1}{4}} + x(2)e^{-j2\pi\frac{2}{4}} + x(3)e^{-j2\pi\frac{3}{4}}\right) \quad \ldots (6b)$$

April 19 (Wed)

Final Exam Schedule: 18th (Thur)
12:15-2:30 pm

Group I Classes

Group I classes are those classes which meet M, W, F, MTW, MWR, MTWF, MWRF, MTWRF, MW, WF, MWF, MF, TW, WR, MT, WS.

Regular Class Start Times	Final Examination Days	Final Examination Time
7:00 through 8:25 AM	Friday, May 19	7:15-9:30 AM
8:30 through 9:25 AM	Tuesday, May 23	7:15-9:30 AM
9:30 through 10:25 AM	Thursday, May 18	7:15-9:30 AM
10:30 through 11:25 AM	Monday, May 22	9:45 AM-12:00 PM
11:30 AM through 12:25 PM	Wednesday, May 17	9:45 AM-12:00 PM
12:30 through 1:25 PM	Friday, May 19	12:15-2:30 PM
1:30 through 2:25 PM	Tuesday, May 23	12:15-2:30 PM
2:30 through 3:25 PM	Thursday, May 18	12:15-2:30 PM
3:30 through 4:25 PM*	Monday, May 22	2:45-5:00 PM
4:30* through 5:25 PM*	Wednesday, May 17	2:45-5:00 PM

Note: Final Exam is in the Same format as the midterm. Be Sure to Bring Your Prototype System.

Note 2: Project ON CANVAS. Part I & Part II. 25 pts.

Part I: Integration of PID Control with ADC

Part II: Research. PPT.

Note 3: Presentation Date

8th (Monday) ~ 10th (Wed) Final (18th, Th)

(1) Presentation.
(2) In-Class Demo of the Project.

12:15-2:30 pm.

Last Day of Class 15th. (Monday) Review.

Example: Continuation of D.F.T. Example.

For m=2, from Eqn (1) pp58

$$X(m)\Big|_{m=2} = \frac{1}{4}\sum_{n=0}^{3} x(n)\, e^{-j2\pi\frac{2\cdot n}{4}}$$

$$= \frac{1}{4}\left[x(0)e^{-j0} + x(1)e^{-j2\pi\frac{1\cdot 2}{4}} + x(2)e^{-j2\pi\frac{2\cdot 2}{4}} + x(3)e^{-j2\pi\frac{3\cdot 2}{4}}\right] \quad \ldots (6c)$$

for m=3,

$$X(m)\Big|_{m=3} = \frac{1}{4}\sum_{n=0}^{3} x(n)\, e^{-j2\pi\frac{3\cdot n}{4}}$$

$$= \frac{1}{4}\left[x(0)e^{-j\cdot 0} + x(1)e^{-j2\pi\frac{3\cdot 1}{4}} + x(2)e^{-j2\pi\frac{3\cdot 2}{4}} + x(0)e^{-j2\pi\frac{3\cdot 3}{4}}\right] \quad \ldots (6d)$$

Hence, we can form a Col. Vector By Arranging $X(0)$, $X(1), \dots, X(3)$ as follows.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \frac{1}{4} E_{4\times4} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad \dots(1)$$

where

$$E_{4\times4} = \begin{bmatrix} W_{00} & W_{01} & W_{02} & W_{03} \\ W_{10} & W_{11} & W_{12} & W_{13} \\ W_{20} & W_{21} & W_{22} & W_{23} \\ W_{30} & W_{31} & W_{32} & W_{33} \end{bmatrix}_{4\times4} \quad \dots(2)$$

Note: W_{ij} $\quad\dots(3)$

Index for Row. Index for Col.

From Eqn (1) & (2), we have

$$\begin{array}{c} \\ m \downarrow \end{array} \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \frac{1}{4} \begin{array}{c} \xrightarrow{\quad n \quad} \\ m \downarrow \begin{bmatrix} W_{00} & W_{01} & W_{02} & W_{03} \\ W_{10} & W_{11} & W_{12} & W_{13} \\ W_{20} & W_{21} & W_{22} & W_{23} \\ W_{30} & W_{31} & W_{32} & W_{33} \end{bmatrix}_{4\times4} \end{array} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \downarrow n \quad \dots(3)$$

$W_{ij}\Big|_{i=1, j=2}$

$$\frac{1}{4}\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

W_{ij}, i for Row. j for Col.

where $W_{ij} = e^{-j2\pi\frac{i\cdot j}{N}}$ $\quad\dots(4)$

From Euler Equation

$$e^{-j2\pi\frac{i\cdot j}{N}} = \cos 2\pi\frac{i\cdot j}{N} - j\sin 2\pi\frac{i\cdot j}{N} \quad \dots(5)$$

Find the Entry of E matrix at 1st Row, 1st Col. Location.

Since, we have Eqn (5).

$$e^{-j2\pi\frac{i\cdot j}{N}} = e^{-j2\pi\frac{i\cdot j}{4}}$$

$$= e^{-j2\pi\frac{i\cdot j}{4}}\Big|_{\substack{i=0\\ j=0}} = 1$$

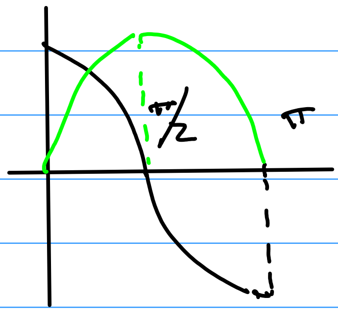
for the 3rd Row, 2nd col.

$i=2$, $\quad j=1$

From

$$e^{-j2\pi\frac{i\cdot j}{N}}\Big|_{\substack{i=2, j=1\\ N=4}} = e^{-j2\pi\frac{2}{4}}$$

$$e^{-j2\pi\frac{2}{4}} = e^{-j\pi} = \cos\pi - j\sin\pi = -1$$



$$P(m) = \sqrt{Re^2[X(m)] + Im^2[X(m)]}$$

$$P(0) = \sqrt{Re^2[X(0)] + Im^2[X(0)]}$$
$$= \sqrt{3.25^2 + 0^2} = 3.25$$

$X(0) = \frac{1}{4}(2+3+4+4) = 3.25$

$X(1) = \frac{1}{4}(2-3j-4+4j) = \frac{1}{4}(-2+j)$

$X(2) = \frac{1}{4}(2-3+4-4) = \frac{1}{4}(-1)$

$X(3) = \frac{1}{4}(2+3j-4-4j) = \frac{1}{4}(-2-j)$

Now, Consider the Last Row, Last Col. $i = N-1 = 3$, $j = N-1 = 3$.

From

$$e^{-j2\pi\frac{3\cdot3}{4}} = e^{-j\,9\pi/2} = e^{-j\frac{8\pi}{2} - j\frac{\pi}{2}}$$
$$= e^{-j4\pi - j\frac{\pi}{2}} = 1\cdot e^{-j\frac{\pi}{2}}$$
$$= \cos\frac{\pi}{2} - j\sin\frac{\pi}{2} = 0 - j\cdot 1 = -j$$

$\cos 4\pi - j\sin 4\pi = 1$ ✓

for $m=1$,

$$P(1) = \sqrt{Re^2[X(1)] + Im^2[X(1)]}$$
$$= \sqrt{(-2/4)^2 + (1/4)^2}$$ finish the evaluation

April 24 (Monday).

Example: FFT.C

CMPE242-Embedded-Systems- / 2018S-26-fft.c

From the Handout, Ref from the github.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \end{bmatrix} \quad \cdots(3^*)$$

So,

step.

Step 1. Find $E_{N\times N}$. 2. Find D.F.T. using $E_{N\times N}$ And $x(n)$.

Step 3. Find $P(m)$ for $m = 0, 1, \cdots, N-1$

CMPE242

Spring 2023

1°

Note: You can Compile/Build for Your Laptop platform. No Need for Arm-Linux-gcc Cross Compiler. 6z

```
/*************************************************************************
 * Program is for CMPE class use, see Dr. Harry Li's lecture notes for details *
 * Reference: Digital Signal Processing, by A.V. Oppenhaim;
 * fft.c for calculting 4 points input, but you can easily expand this to 2^x inputs;
 * : x0.1;                    Date: Sept. 2009;
 * Note: cross compiled for arm-linux-gcc, be sure to modify make file
to link math lib when compiling, by adding -lm
This code then was tested on ARM11 board. Feb 2015.
*************************************************************************/
```

2° link Math Library

~/Desktop/SJSU/befor2018/EE264/EE264Ubuntu/OpenGL/CT/lec22FFTIFFT$ ./fft

```
harry@harry-laptop: ~/Desktop/SJSU/befor2018/EE264
harry@harry-laptop:~/Desktop/SJSU/befor2018/EE264/
IFFT$ ls
fft  fft_cpp.cpp  fft_ifft_cpp.cpp  ifft_cpp.cpp
harry@harry-laptop:~/Desktop/SJSU/befor2018/EE264/
IFFT$ ./fft
*********Before*********
X[1]:real == 2.000000 imaginary == 0.000000
X[2]:real == 3.000000 imaginary == 0.000000
X[3]:real == 4.000000 imaginary == 0.000000
X[4]:real == 4.000000 imaginary == 0.000000


**********After*********
X[1]:real == 13.000000 imaginary == 0.000000
X[2]:real == -2.000000 imaginary == 1.000000
X[3]:real == -1.000000 imaginary == 0.000000
X[4]:real == -2.000000 imaginary == -1.000000
It took me 145 clicks (0.000000 seconds).
harry@harry-laptop:~/Desktop/SJSU/befor2018/EE264/
IFFT$
```

```
/****************************************************
*
//This program is converted from a FORTRAN program
g *
//book by A.V. Oppenheim; *
//Status: Tested; *
//gcc fft.cpp -o fft *
//Additional information: See Dr. Hua Harry Li's ha
//fft.cpp for calculting 4 pts input, but can easil
*
//****************************************************
```

Note: To Compile -lm ON x86 platform.

April 26 (Wed)

Example: Data Validation using Power Spectrum of the FFT.

Note 1: BaseLine Code 128 pts.

Power Spectrum.

Note: $P(m) = P(m+kN)$ ... (1)

Periodical Function.

Period = N

$P(m) = P(-m)$ ... (2)

even function

Nyquest Sampling Theorem:

$f_{sampling} \geq 2 f_{max}$ ... (1)

Create 128 pts Data As a BaseLine Test.

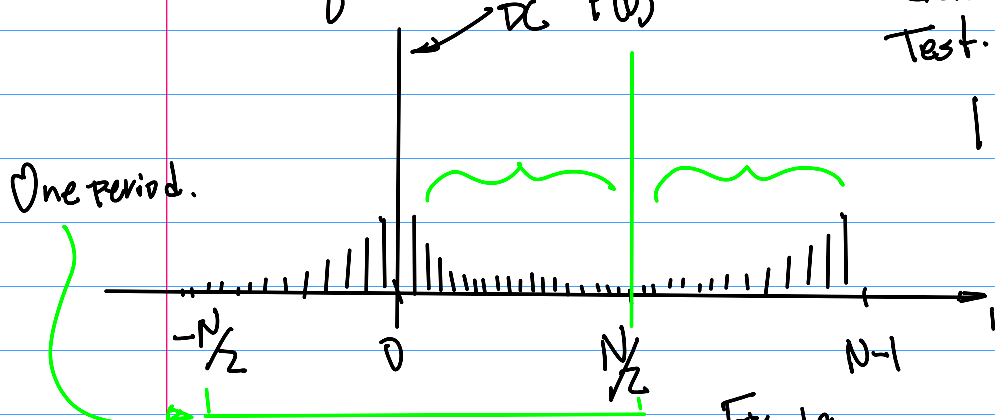


Fig.1a

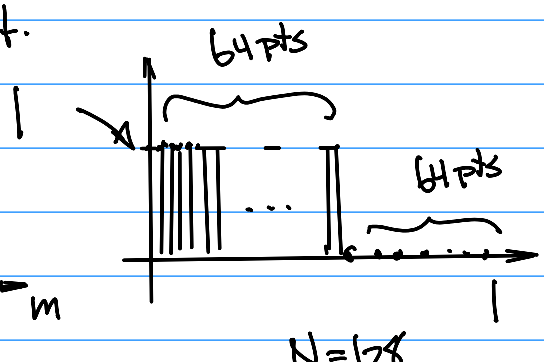


Fig.1b

CmpE242

Spring 2023

Execution of the F.F.T and Computation of the Power Spectrum, Lead to the Similar plot as in Fig. 1a. plot 1.

Discussion:

$P(m)\Big|_{m=N/2}$ Freq. Componment at the highest Frequency Index m.

Now, Modify the Input Data to 64 pts "1"s + 32 pts "1". Leave 32 pts "0".

e.g. 96 pts "1"s plus 32 pts "0".

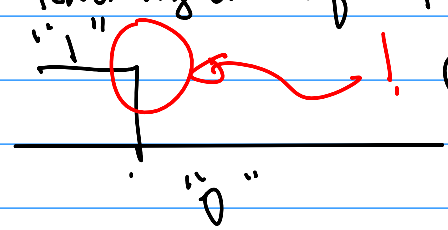
plot the $P(m)$. plot 2

Observation: plot 1 (64 pts "1"s) has more higher Frequency Components. OR, more precisely, more energy in the higher Frequency Range.

Create 3rd Data set, $96+16=112$ "1"s.

↓

fewer higher Frequency Comp.

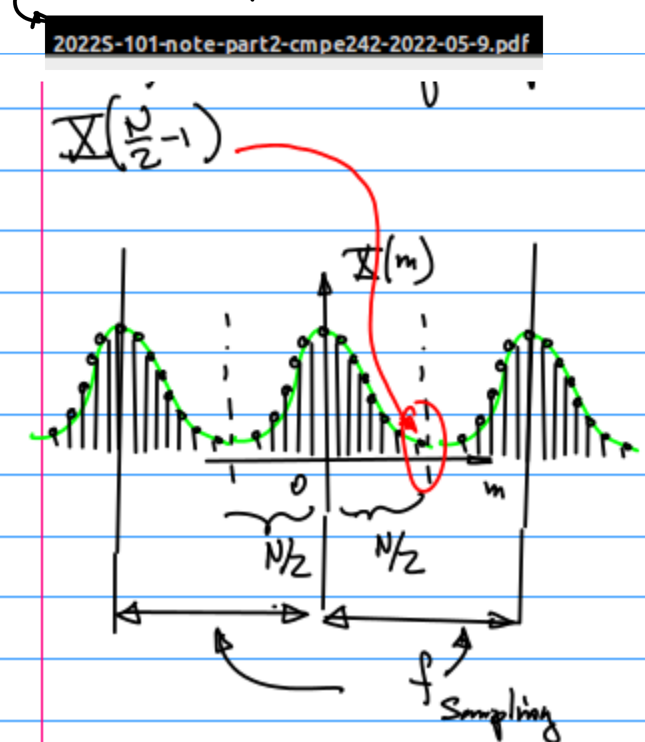


! Gibb's ~

Remark: The Signal Energy Distribution in the higher frequency Range Can be demonstrated with These 3 plots. Therefore the Sampling frequency for each of them satisfies the following condition

$$f_{Sampling_1} \geq f_{Sampling_2} \geq f_{Sampling_3} \quad \ldots (2)$$

Ref. PP25



Note 1: The $f_{Sampling}$ sets Apart of each Period of the $P(m)$;

Note 2: In The Higher Frequency Range, the energy distribution of Non-zero $P(m)$ Contributes to "Alisazing".

Note 3: To Validate ADC Data, we define the following index

$$\eta = \sum_{m \in \Omega_1} P(m) \Big/ \sum_{m=0}^{N-1} P(m) \quad \ldots (5)$$

where Ω_1 : Higher frequency Range. Such as

$$N_{Low} \le m \le N/2 \quad \ldots (6)$$

User defined Lower Bound of the High Freq. Range.

Example, for N=128. N/2 highest Freq. Index, $\frac{N}{2} = 64$, then depending on the Application We can have $N_{Low} = 50$

$$\sum_{m=50}^{64} P(m) \quad \ldots (7)$$

May 1st (Monday)
Example: Compute power spectrum for the data validation using N=4;
Sol:

From the power spectrum
P(0), P(1), P(2), P(3) for the entire period N;

Suppose we want validate the date based on equation (5), assuming the frequency index m = 1, and 2;

eta M <= 20%

Hence,
The total energy = P(0) + P(1) ... + P(3);
(please perform the calculation off line)

The energy in the higher frequncy range:

P(1) + P(2) + P(3)

Then the ration eta:

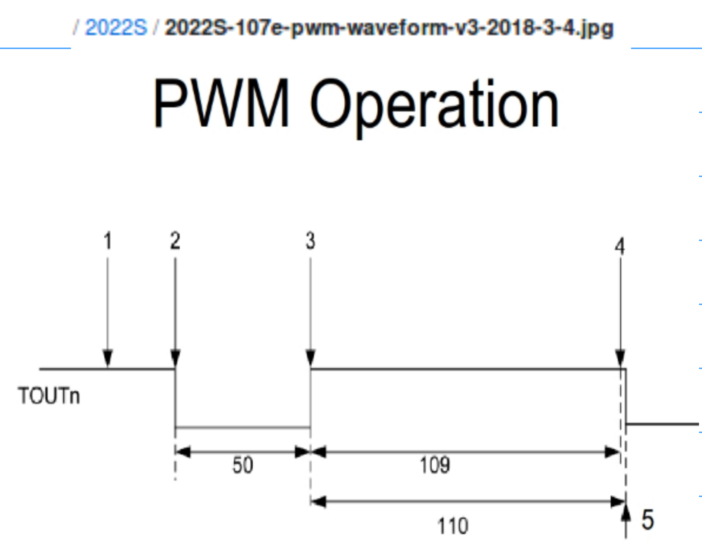
$$\text{eta} = \frac{P(1) + P(2) + P(3)}{P(0) + P(1) + P(2) + P(3)}$$

Comparing this result to the limit set by the application requirement 20%

If not, then increase the sampling frequency per Nyquest Theorem.

Now consider the last part of the class, PWM and ADC Archietecture.

Example:
Ref. Timing/SPRs for PWM



FromSamsung ARM11 data sheet;

Ref: pp. 1

2022S-101-note-part2-cmpe242-2022-05-9.pdf

Ref: Samsung ARM 11 CPU datasheet

2021F-105-#0-cpu-a... Add files

Using the example from the lecture note, pp. 1

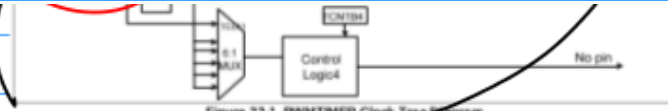


Figure 32-1. PWMTIMER Clock Tree Diagram

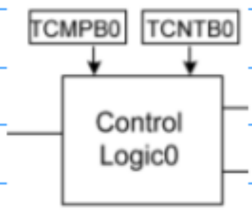
32 PWM TIMER

This chapter describes the functions and usage of PV

a. Input: PCLK peripheral clock.
$\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots,$ of the System Clock

b. Output (2 outputs)

d. Special Purpose Registers

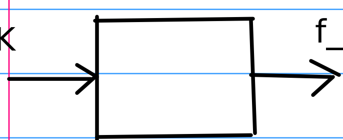


CONF (Configuration Register)

CNT (Control — "Count")

CMP (Comparision)

define f_{pwm} & Duty Cycle.



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

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

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

$= 200 \times 10^6$

Defined By Special Purpose Register.

PP.2

Design Guidelines for PWM:

① CNT (TCNTB∅) Define f_{PWM} ③ CONF Configuration Register. Defines f_{PWM}

Control/Count Timer Buffer∅ for PWM∅

② CMP (TCMPB∅) Defines Duty Cycle

Comparision

Example: Suppose PCLK $= 500 \times 10^6$ (MHz), Find the Counts N for TCNTB∅

Suppose to Drive A Stepper Motor $f_{PWM} = 2KHz = 2 \times 10^3$

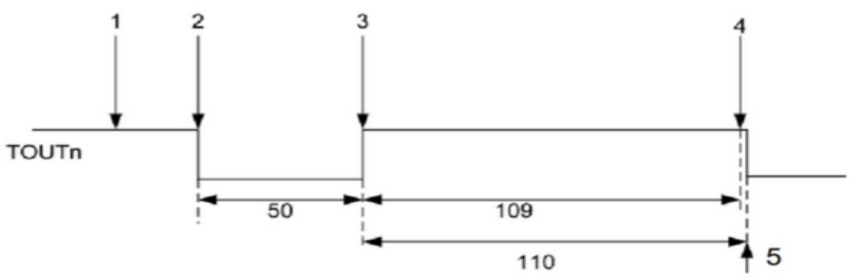
$$\frac{PCLK}{N} = f_{PWM} \quad \ldots (4)$$

2022S / 2022S-107e-pwm-waveform-v3-2018-3-4.jpg

Five steps for defining (1) frequency of the PWM output; (2) the duty cycle;
Note 1: CNT is responsible for defining f_pwm, the calculation of N, e.g., the number of counts is given in Equation (4) on this pape.
Note 2: f_pwm is achieved by counter down counting from N, once it reached to 0, then interrupt is triggered and then repeating the process, so f_pwm is achieved.
Note 3: for duty cycle definition, we compute 2 parts, the first one is the calculation of the counts number from the percentage of the desired duty cycle. Example: 69% duty cycle, 69% * N, if N in the waveform example is 159, then M = 109;

In summary, the 5 steps process of setting up f_pwm and duty_cycle is given below:

PWM Operation



Example:

1. Initialize the TCNTBn with 159(50+109) and the TCMPBn with 109.
2. Start Timer by setting the start bit and manual update bit off.
The TCNTBn value of 159 is loaded into the down-counter, the output is driven low.
3. When down-counter counts down to the value in the TCMPBn register 109, the output is changed from low to high.
4. When the down-counter reaches 0, the interrupt request is generated.
5. The down-counter is automatically reloaded with TCNTBn, which restarts the cycl