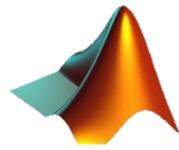


RMXplorer



LAB1 : Sensor

สมาชิก

● นางสาวกัณฑิลา	บุญญาศิริวัฒน์	66340500038
● นายศักดิ์ไชย	ธนาภาสวัฒน์	66340500052
● นางสาวศุภารพิชญ์	เทพกร	66340500057

วัตถุประสงค์หลัก

- เพื่อศึกษาความสามารถในการรับรู้ปริมาณทางฟิสิกส์ของ Sensor ทั้งหมดดังแต่ต้นจนจบกระบวนการได้
- เพื่อศึกษาหลักการทำงานของ Sensor ตั้งแต่การตรวจจับจนถึงการแสดงผล
- เพื่อศึกษาทฤษฎีทางฟิสิกส์ หรือข้อมูลจาก Datasheet ที่สามารถอธิบายการทำงานของ Sensor
- เพื่อศึกษาการออกแบบบริการทดลองให้สอดคล้องกับวัตถุประสงค์และผลการเรียนรู้อย่าง โดยใช้อุปกรณ์ต่างๆ รวมถึงเครื่องมืออย่าง Simulink, MATLAB mlx file และบอร์ด Microcontroller ที่จัดไว้เตรียมให้

สมมติฐาน

- สามารถอธิบายความสามารถในการรับรู้ปริมาณทางฟิสิกส์ของ Sensor ทั้งหมดตั้งแต่ต้นจนจบกระบวนการได้
- สามารถเข้าใจหลักการทำงานของ Sensor โดยต้องเข้าใจทุกขั้นตอนของการทำงานของเซ็นเซอร์ ตั้งแต่การตรวจจับจนถึงการแสดงผล
- สามารถกำหนดตัวแปรในการทดลองได้อย่างเหมาะสมโดย มีทฤษฎีรองรับจากฟิสิกส์ หรือข้อมูลจาก Datasheet ที่นำเชื่อถือ
- สามารถออกแบบวิธีการทดลองให้สอดคล้องกับวัตถุประสงค์และผลการเรียนรู้ย่อย โดยใช้อุปกรณ์ต่างๆ รวมถึงเครื่องมืออย่าง Simulink, MATLAB mlx file และบอร์ด Microcontroller ที่จัดเตรียมให้

ขั้นตอนการดำเนินงาน

1. แบ่งหน้าที่และความรับผิดชอบของสมาชิกในทีม
2. กำหนดสมมติฐานสำหรับแต่ละการทดลอง
3. เตรียมอุปกรณ์ที่จำเป็นสำหรับการทดลอง
4. ดำเนินการทดลองตามขั้นตอนที่วางไว้
5. บันทึกผลการทดลองอย่างละเอียด
6. วิเคราะห์และสรุปผลจากข้อมูลที่บันทึกไว้
7. จัดทำรายงานสรุปผลการทดลอง

1. Potentiometer

1.1 จุดประสงค์

- 1.1.1 สามารถอธิบายการทำงานของ Potentiometer ประเภท Rotary Potentiometer และ Linear Potentiometer ได้
- 1.1.2 สามารถอธิบายความสัมพันธ์ของการตอบสนองของแรงดันไฟฟ้าจากที่มีต่อองค์การหมุนของ Rotary Potentiometer และระยะเวลาการขับเลื่อนของ Linear Potentiometer ได้
- 1.1.3 อธิบายการแปลงสัญญาณ Analog เป็นสัญญาณ Digital ตามรูปแบบของ Schmitt-trigger และสามารถทำการกำหนดจุด Threshold Voltage ที่จะจрог Schmitt-trigger จะเปลี่ยนสัญญาณจากต่ำไปสูงหรือสูงไปต่ำได้

1.2 สมมติฐาน

- 1.2.1 สามารถทำเข้าใจการทำงานของ Potentiometer ประเภท Rotary Potentiometer และ Linear Potentiometer ได้โดย Potentiometer เป็น Sensor สามารถวัดค่าความต้านทานที่เปลี่ยนแปลงได้ตามตำแหน่ง เมื่อหมุนหรือเลื่อน potentiometer จะทำให้เกิดการเปลี่ยนแปลงค่าแรงดันไฟฟ้าในวงจรไปตามข้อมูลที่มีอยู่ใน Datasheet เนื่องจากความต้านทานที่มีการเปลี่ยนแปลงไป
- 1.2.2 ความสัมพันธ์ระหว่างแรงดันไฟฟ้ากับตำแหน่งของ potentiometer มีความสัมพันธ์เป็น function แบบ 1 to 1 คือการเปลี่ยนแปลงของศักยภาพของ Rotary Potentiometer หรือระยะเวลาการเลื่อนของ Linear Potentiometer ที่ค่านี้ มีผลกับการเปลี่ยนแปลงแรงดันไฟฟ้าจาก ค่าหนึ่ง โดยแบ่งออกเป็น 3 ชนิดคือ Linear Taper, Logarithmic Taper, และ Reverse Logarithmic Taper
- 1.2.3 วงจร Schmitt-trigger สามารถเปลี่ยนสัญญาณแรงดันไฟฟ้าจากระดับ Analog เป็นระดับ Digital ได้ เมื่อแรงดันไฟฟ้าถึงค่าความต่างศักย์ที่ตั้งไว้ (Threshold Voltage) โดยสัญญาณจะเปลี่ยนจากต่ำไปสูง หรือจากสูงไปต่ำ

1.3 นิยามศัพท์เฉพาะ

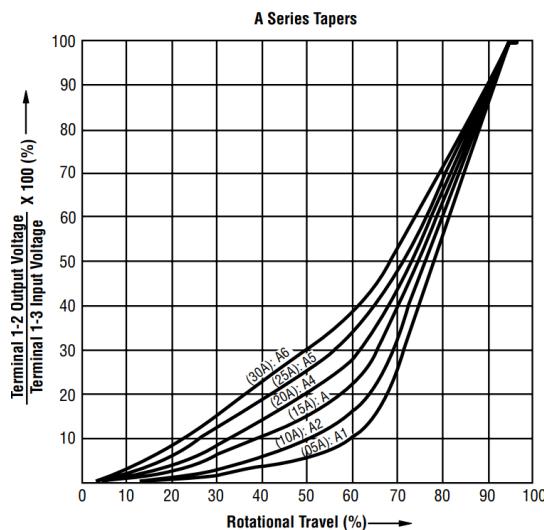
- Analog เป็นสัญญาณที่มีความต่อเนื่องแบบลูกคลื่น ค่าที่ออกมากีจำนวนมาก
- Digital เป็นสัญญาณที่มีความต่อเนื่องแบบสี่เหลี่ยมต่อ กัน ค่าที่ออกมากีแต่ 2 ค่า คือ 0 ,1
- Threshold Voltage เป็นจุดที่เป็นแรงดันขีดเริ่ม เมื่อแรงดันมีค่ามากกว่าจุด Threshold Voltage ค่าแรงดันที่ออกมาก็มีค่าแค่ 0 ,1 เท่านั้น
- Knop คือ ชื่อที่ไว้ใช้เรียกตัวหมุนปรับขีดบน Rotary Potentiometer
- Wiper คือ ชื่อที่ไว้ใช้เรียกตัวเลื่อนบน Linear Potentiometer
- Potentiometer เป็นเซ็นเซอร์วัดตำแหน่งและระยะเวลาการหมุนของชนิดเปลี่ยนแปลงค่าความต้านทาน

1.4 นิยามเชิงปฏิบัติการ

- การหมุนของ Rotary Potentiometer คือ การที่หมุนไปตามขีดเลขบน knob เก็บค่าที่ลํะ 10 ตั้งแต่ 0 ไปจนถึง 100
- การขับเลื่อนของ Linear Potentiometer คือ การเลื่อนตัว slide บน Linear Potentiometer ไปที่ลํะ 6 มิลลิเมตร เนื่องจากจาก Datasheet พบว่าเป็นการวาดกราฟโดยเพิ่มระยะขึ้นที่ลํะ 10 % เป็นจำนวน 11 ค่า โดยเริ่มตั้งแต่ 0 ไปจนถึง 100 ดังนั้นมีระยะทางทั้งหมดที่วัดได้เป็น 60 มิลลิเมตร หารด้วย 10 ซึ่งเป็นจำนวนการเลื่อนที่ต้องการ ก็จะได้ว่าระยะห่างแต่ละจุดมีค่าอยู่ที่ 6 มิลลิเมตร

1.5 เอกสารและงานวิจัยที่เกี่ยวข้อง

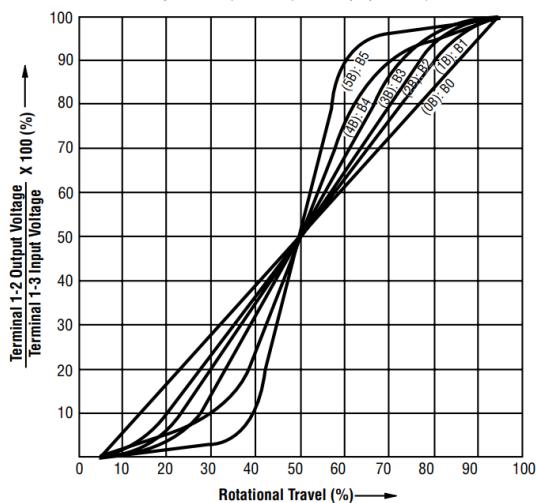
- 1.5.1 Datasheet ของ Rotary Potentiometer และ Linear Potentiometer เพื่อใช้ในการเปรียบเทียบกับกราฟที่ได้ออกมาหลังจากการทดลอง และกราฟจาก Datasheet ว่ากราฟที่ได้เป็น Rotary Potentiometer หรือ Linear Potentiometer ชนิดใด



รูปที่ 1 Datasheet ของ Rotary Potentiometer ชนิด A

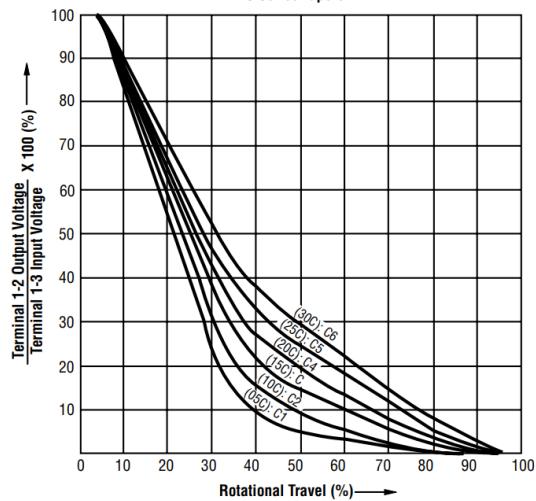
B Series Tapers

PDB181 Standard: Linear taper "B" represents (0B):B0 in taper chart.
 PDB182 Dual Gang: Linear taper "B" represents (3B):B3 in taper chart.

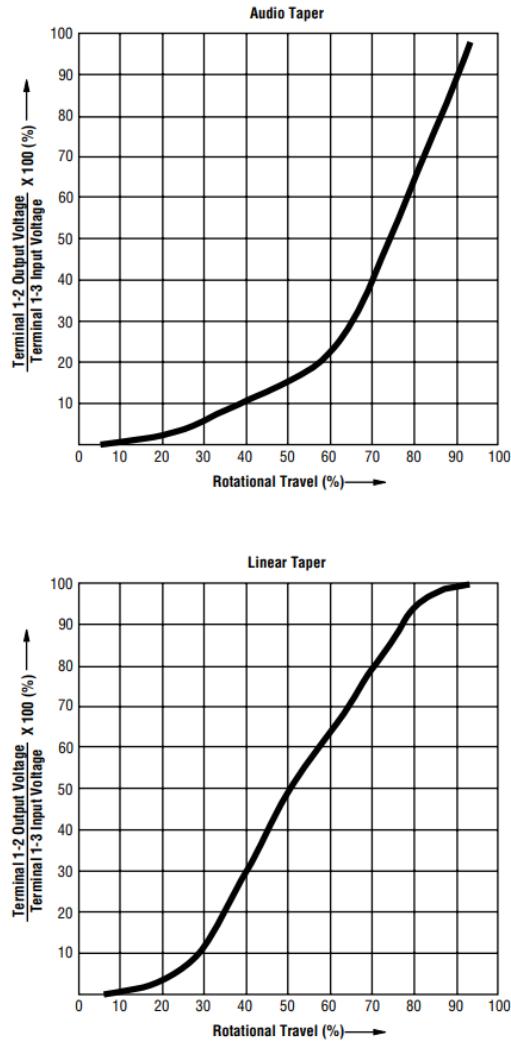


រូប៖ 2 Datasheet ទិន្នន័យ Rotary Potentiometer មុំទី B

C Series Tapers



រូប៖ 3 Datasheet ទិន្នន័យ Rotary Potentiometer មុំទី C



รูปที่ 4 Datasheet ของ Linear Potentiometer

1.6 วิธีดำเนินการทดลอง

Potentiometer ทั้ง 2 ชนิดมีวิธีการดำเนินการทดลองที่คล้ายกัน

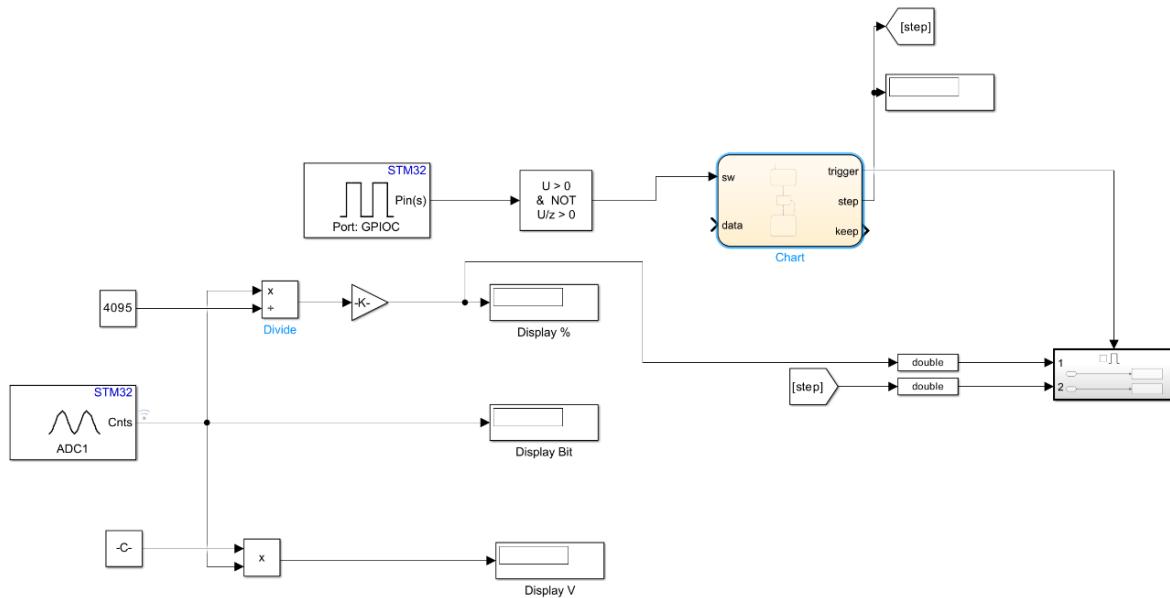
1.6.1 Rotary Potentiometer

- 1.) ออกแบบวิธีการทดลอง โดยจากการที่ได้ศึกษาข้อมูลของ Rotary Potentiometer จาก Datasheet พบว่า ค่า Rotational Travel มีค่าเป็นร้อยละที่มีความละเอียดอยู่ที่ 10 % จากรูปภาพข้างต้นในรูปที่ 1 2 และ 3 ทางคณะผู้จัดทำจึงตัดสินใจที่จะเก็บค่า Potentiometer ทั้ง 3 ชนิด ได้แก่ชนิด A B และ C เป็นจำนวน 11 ค่า เพื่อให้ค่าที่ได้ออกมาไม่ค่าใกล้เคียงกับ Datasheet มากที่สุด โดยเป็นจำนวนค่าละ 500 ครั้ง เพื่อป้องกันการเกิดค่าที่คลาดเคลื่อน ซึ่ง Rotary Potentiometer ที่ติดมากับบอร์ดมีสเกล 100 สเกลพอดี จึงทำการเก็บค่าเริ่มจากค่าที่ 0.00 หน่วย และเพิ่มทีละ 10 หน่วยจนไปถึง 100 หน่วย



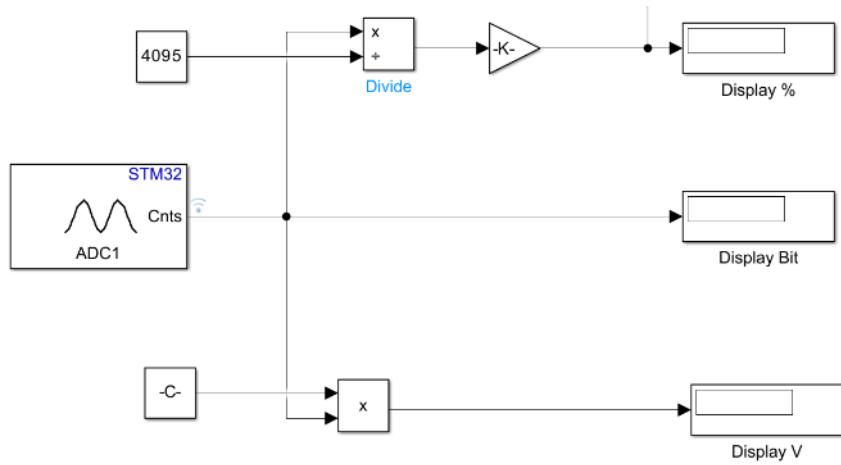
รูปที่ 5 knop ของ Rotary Potentiometer ชนิด A ,B และ C

2.) ทำการเขียนโปรแกรมและสร้าง Simulink เพื่อเก็บค่าข้อมูล 500 ค่า ตามที่ได้ออกแบบไว้ ซึ่งจะได้ diagram ออกแบบตามภาพด้านล่าง



รูปที่ 6 การเขียนโปรแกรมและสร้าง Simulink

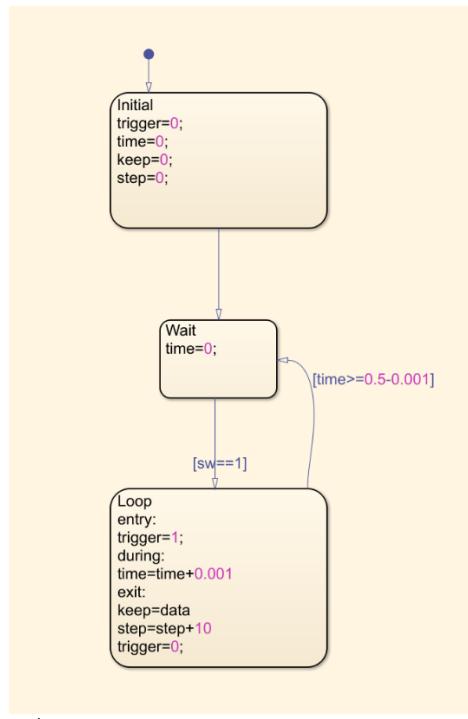
รับค่าสัญญาณจาก Potentiometer ด้วย Analog to Digital Converter และทำการคำนวณเพื่อหาเปอร์เซ็น Output ของ Potentiometer เพื่อนำค่าที่ได้ส่งเข้า chart



รูปที่ 7 การออกแบบการทดลองใน Simulink

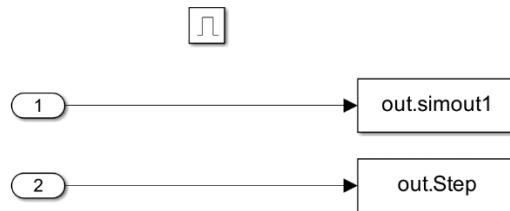
จากภาพด้านล่างใน Chart นั้นจะมี state อยู่ 3 state คือ Initial, Wait และ Loop โดยการทำงานของแต่ละstate มีดังนี้

1. Initial เป็น state เริ่มต้น ที่กำหนดค่าของตัวแปร ซึ่งในที่นี้ได้แก่ trigger = 0, time = 0, keep = 0, และ step = 0 หลังจากตั้งค่าเริ่มต้นเสร็จแล้ว ระบบจะทำการเปลี่ยนไปที่ state Wait
2. Wait เป็น state ที่ time ถูกตั้งค่าให้เป็น 0 และจะทำงานวนอยู่ใน state นี้ไปเรื่อยๆ จนกว่าเมื่อไหร่ที่ sw==1 (กดสวิตช์สีฟ้าบนบอร์ด Nucleo STM32G474RE) จะเป็นจริง แล้วระบบก็จะทำการเปลี่ยนไปที่ state Loop
3. Loop เป็น state ที่จะทำงาน เมื่อเงื่อนไข sw == 1 เป็นจริง และจะมีการกำหนด state ย่อลงไป
 - Entry : กำหนดให้ trigger = 1
 - During : จะทำการอัปเดตค่า time โดยที่เพิ่มขึ้นทีละ 0.001 s
 - Exit : กำหนดให้ keep = data, กำหนดให้ step = step + 10 และ trigger = 0
 และจะออก state Loop กลับไป state Wait เมื่อ loop time จนมีค่ามากกว่า 0.5



รูปที่ 8 การออกแบบของ chart ใน Simulink

จากภาพ block in หรือ input ด้านบนจะทำการรับข้อมูลที่เก็บค่าจาก Potentiometer ส่วน block in หรือ input ด้านล่างจะทำการรับข้อมูลที่เก็บค่าของ step และทำการส่งข้อมูลทั้ง 2 ส่วน ไปที่ workspace ในโปรแกรม MATLAB ผ่าน block To Workspace ใน Simulink



รูปที่ 9 การออกแบบของ Subsystem ใน Simulink

```

Data = out.simout1;
chunkSize = 500; % ขนาดของแต่ละก้อน
nChunks = floor(length(Data) / chunkSize); % จำนวนก้อนที่หนึ่ง
averages = zeros(1, nChunks); % สร้าง array เพื่อเก็บค่าเฉลี่ยแต่ละ step
schmitt = zeros(1, nChunks);
step = linspace(0,100,11);

for i = 1:nChunks
    % หา index เป็นไปและสิ้นสุดของแต่ละก้อน
    startIndex = (i - 1) * chunkSize + 1;
    endIndex = i * chunkSize;

    % คำนวณเฉลี่ยแต่ละช่วงและเก็บไว้ใน averages
    averages(i) = mean(Data(startIndex:endIndex));
    %schmitt trigger คือ
    if averages(i) >= 50.0
        schmitt(i) = 1;
    else
        schmitt(i) = 0;
    end
end

%แสดงค่า
disp('Averages for each chunk:');
disp(averages);
disp(schmitt);

%Plot Graph
ax_1 = subplot(1,2,1);
ax_2 = subplot(1,2,2);

plot(ax_1,step,averages);
title(ax_1,"Voltage:Rotation");
grid(ax_1,"on");
xlabel(ax_1,"Rotational Travel");
ylabel(ax_1,"Vout/Vin*100");
hold on;

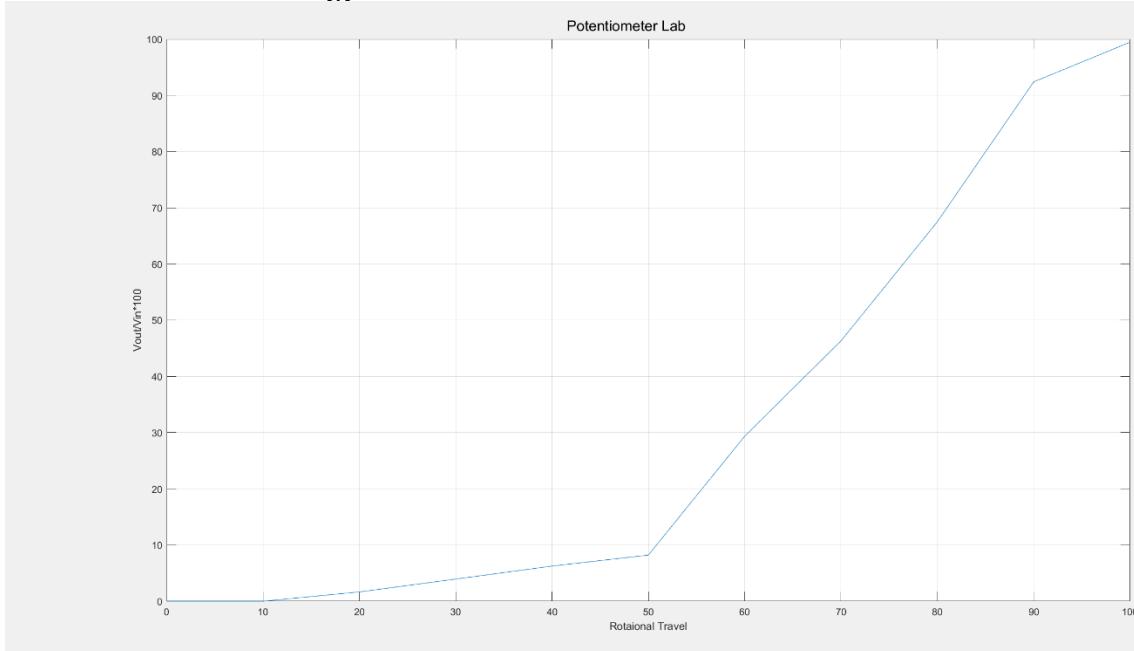
plot(ax_2,step,schmitt);
title(ax_2,"Schmitt Trigger");
grid(ax_2,"on");
xlabel(ax_2,"Rotational Travel");
ylabel(ax_2,"Trigger");
hold off;
sgtitle("Potentiometer (B) and Schmitt trigger Lab");

```

รูปที่ 10 Code การวิเคราะห์ค่า และ Plot Graph

เมื่อได้ค่าอุปกรณ์แล้ว นำข้อมูลที่ได้มาวิเคราะห์เป็นกราฟโดยให้แกน x คือ Rotational Travel ส่วน

$$\text{แกน } y \text{ คือ } \frac{V_{out}}{V_{in}} \times 100$$



รูปที่ 11 ตัวอย่างการวาดกราฟของ Rotary Potentiometer ชนิด A

- 3.) วิเคราะห์กราฟที่ได้จากการทดลองเปรียบเทียบกับกราฟใน Datasheet เพื่อรับว่าเป็นกราฟของ Rotary Potentiometer ชนิดใด โดยอ้างอิงจากลักษณะเส้นกราฟที่สอดคล้องกัน เช่น ความชันของ กราฟ หรือรูปแบบแรงดันที่เปลี่ยนแปลงไป เป็นต้น
- 4.) ทำการวิเคราะห์และสรุปผลการทดลอง

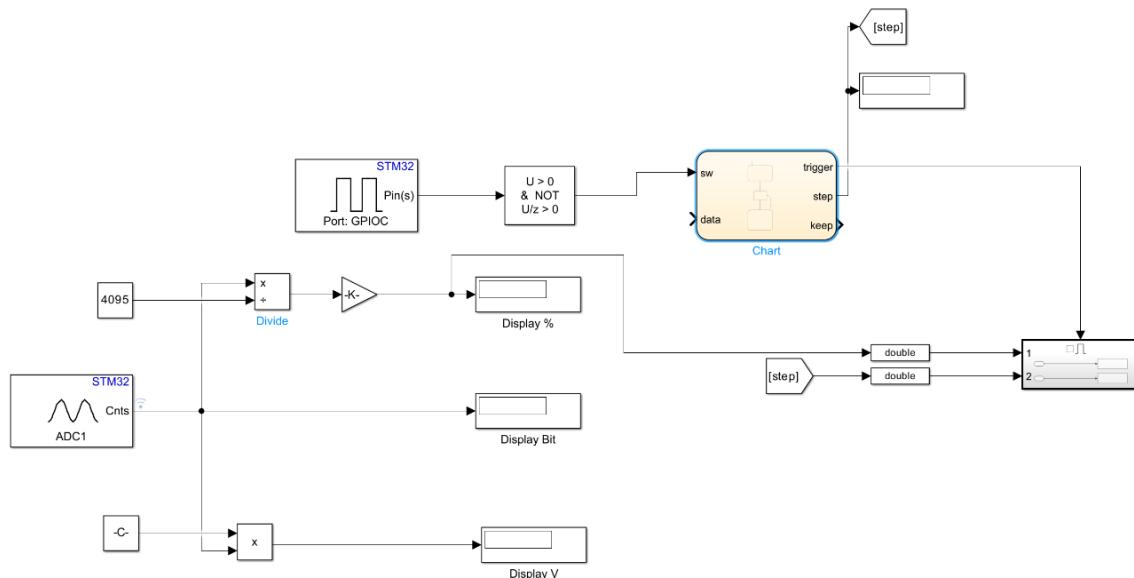
1.6.2 Linear Potentiometer

- 1.) ออกแบบวิธีการทดลอง โดยจากการที่ได้ศึกษาข้อมูลของ Linear Potentiometer จาก Datasheet พบว่า ค่า Rotational Travel มีค่าเป็นร้อยละที่มีความละเอียดอยู่ที่ 10 % จากรูปภาพข้างต้นในรูป ที่ 4 ทางคณผู้จัดทำจึงตัดสินใจที่จะเก็บค่า Potentiometer ทั้ง 2 ชนิด ได้แก่ ชนิด Audio Taper และ Linear Taper เป็นจำนวน 11 ค่า เพื่อให้ค่าที่ได้ออกมา มีค่าใกล้เคียงกับ Datasheet มากที่สุด โดยเป็นจำนวนค่าละ 500 ครั้ง เพื่อป้องกันการเกิดค่าที่คลาดเคลื่อน ซึ่ง Linear Potentiometer ที่ ติดมากับบอร์ดมีระยะทางทั้งหมดเป็น 60 มิลลิเมตร เมื่อหารด้วย 10 ซึ่งเป็นจำนวนการเลื่อนที่ ต้องการ ก็จะได้ว่าระยะห่างแต่ละจุดมีค่าอยู่ที่ 6 มิลลิเมตร



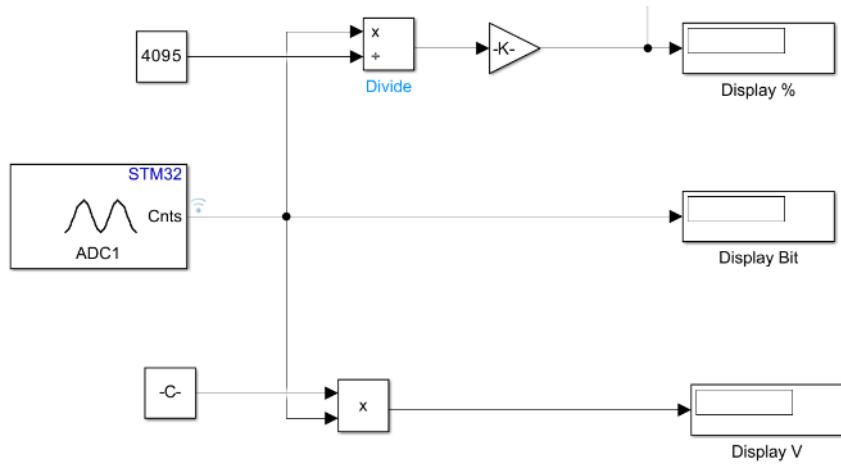
รูปที่ 12 wiper ของ Linear Potentiometer A และ B

2.) ทำการเขียนโปรแกรมและสร้าง Simulink เพื่อเก็บค่าข้อมูล 500 ค่า ตามที่ได้ออกแบบไว้ ซึ่งจะได้ diagram ออกแบบด้านล่างนี้



รูปที่ 13 การเขียนโปรแกรมและสร้าง Simulink

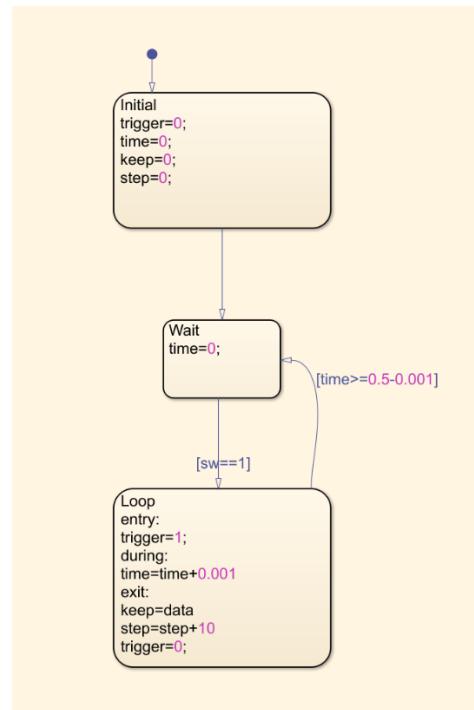
รับค่าสัญญาณจาก Potentiometer ด้วย Analog to Digital Converter และทำการคำนวณเพื่อหาเปอร์เซ็น Output ของ Potentiometer เพื่อนำค่าที่ได้ส่งเข้า chart



รูปที่ 14 การออกแบบการทดลองใน Simulink

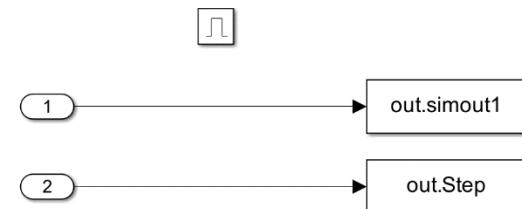
จากภาพด้านล่างใน Chart นั้นจะมี state อยู่ 3 state คือ Initial, Wait และ Loop โดยการทำงานของแต่ละstate มีดังนี้

1. Initial เป็น state เริ่มต้น ที่กำหนดค่าของตัวแปร ซึ่งในที่นี้ได้แก่ trigger = 0, time = 0, keep = 0, และ step = 0 หลังจากตั้งค่าเริ่มต้นเสร็จแล้ว ระบบจะทำการเปลี่ยนไปที่ state Wait
2. Wait เป็น state ที่ time ถูกตั้งค่าให้เป็น 0 และจะทำงานวนอยู่ใน state นี้ไปเรื่อยๆ จนกว่าเมื่อไหร่ที่ sw==1 (กดสวิตช์สีฟ้าบนบอร์ด Nucleo STM32G474RE) จะเป็นจริง แล้วระบบก็จะทำการเปลี่ยนไปที่ state Loop
3. Loop เป็น state ที่จะทำงาน เมื่อเงื่อนไข sw == 1 เป็นจริง และจะมีการกำหนด state ย่อลงไป
 - Entry : กำหนดให้ trigger = 1
 - During : จะทำการอัปเดตค่า time โดยที่เพิ่มขึ้นทีละ 0.001 s
 - Exit : กำหนดให้ keep = data, กำหนดให้ step = step + 10 และ trigger = 0
 และจะออก state Loop กลับไป state Wait เมื่อ loop time จนมีค่ามากกว่า 0.5



รูปที่ 15 การออกแบบของ chart ใน Simulink

จากภาพ block in หรือ input ด้านบนจะทำการรับข้อมูลที่เก็บค่าจาก Potentiometer ส่วน block in หรือ input ด้านล่างจะทำการรับข้อมูลที่เก็บค่าของ step และทำการส่งข้อมูลทั้ง 2 ส่วน ไปที่ workspace ในโปรแกรม MATLAB ผ่าน block To Workspace ใน Simulink



รูปที่ 16 การออกแบบของ Subsystem ใน Simulink

```

Data = out.simout1;
chunkSize = 500; % ขนาดของแต่ละกลุ่ม
nChunks = floor(length(Data) / chunkSize); % จำนวนกลุ่มที่เหลือ
averages = zeros(1, nChunks); % สร้าง array เพื่อเก็บค่าเฉลี่ยแต่ละ step
schmitt = zeros(1,nChunks);
step = linspace(0,100,11);

for i = 1:nChunks
    % หา index เริ่มต้นและสิ้นสุดของแต่ละช่วง
    startIndex = (i - 1) * chunkSize + 1;
    endIndex = i * chunkSize;

    % คำนวณเฉลี่ยของแต่ละช่วงและเก็บไว้ใน averages
    averages(i) = mean(Data(startIndex:endIndex));
    %schmitt trigger คือ
    if averages(i) >= 50.0
        schmitt(i) = 1;
    else
        schmitt(i) = 0;
    end
end

%แสดงค่า
disp('Averages for each chunk:');
disp(averages);
disp(schmitt);

```

%Plot Graph

```

ax_1 = subplot(1,2,1);
ax_2 = subplot(1,2,2);

plot(ax_1,step,averages);
title(ax_1,"Voltage:Rotation");
grid(ax_1,"on");
xlabel(ax_1,"Rotational Travel");
ylabel(ax_1,"Vout/Vin*100");
hold on;

plot(ax_2,step,schmitt);
title(ax_2,"Schmitt Trigger");
grid(ax_2,"on");
xlabel(ax_2,"Rotational Travel");
ylabel(ax_2,"Trigger");
hold off;

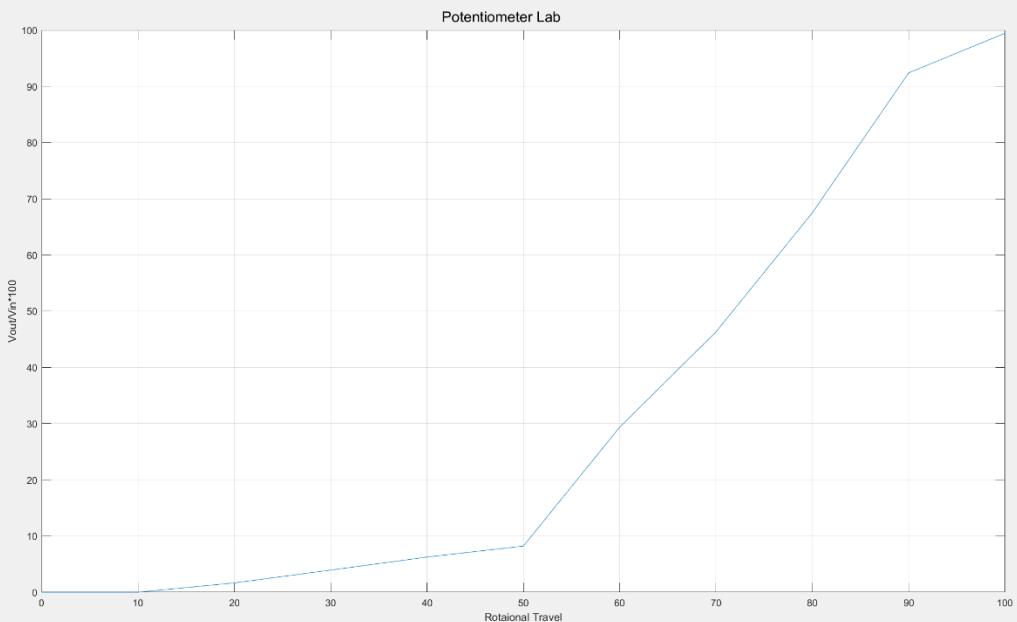
```

sgtitle("Potentiometer (B) and Schmitt trigger Lab");

รูปที่ 17 Code การวิเคราะห์ค่า และ Plot Graph

เมื่อได้ค่าอุกมาแล้ว นำข้อมูลที่ได้มามาวิเคราะห์เป็นกราฟโดยให้แกน x คือ Rotational Travel ส่วน

$$\text{แกน } y \text{ คือ } \frac{V_{out}}{V_{in}}$$



รูปที่ 18 ตัวอย่างการวัดกราฟของ Rotary Potentiometer ชนิด A

- 3.) วิเคราะห์กราฟที่ได้จากการทดลองเปรียบเทียบกับกราฟใน Datasheet เพื่อรับรู้ว่าเป็นกราฟของ Linear Potentiometer ชนิดใด โดยอ้างอิงจากลักษณะเส้นกราฟที่สอดคล้องกัน เช่น ความชันของ กราฟ หรือรูปแบบแรงดันที่เปลี่ยนแปลงไป เป็นต้น
- 4.) ทำการวิเคราะห์และสรุปผลการทดลอง

1.7 วัสดุอุปกรณ์

1.7.1	PTA6043-2015DPA103	จำนวน 1 ตัว
1.7.2	PTA6043-2015DPB103	จำนวน 1 ตัว
1.7.3	PDB181-K420K-103A2	จำนวน 1 ตัว
1.7.4	PDB181-K420K-103B	จำนวน 1 ตัว
1.7.5	PDB181-K420K-103C	จำนวน 1 ตัว
1.7.6	Nucleo STM32G474RE	จำนวน 1 ตัว
1.7.7	สายอับโหลด	จำนวน 1 ชุด
1.7.8	PotenXplorer	จำนวน 1 ชุด
1.7.9	ไม้บรรทัด/เวอร์เนีย	จำนวน 1 ชิ้น
1.7.10	Jumper Wire Cable	

1.8 ตัวแปร

1.8.1 Rotary Potentiometer

ตัวแปรต้น	: ระยะที่หมุน Potentiometer แต่ละวนิด
ตัวแปรตาม	: แรงดันไฟฟ้าจากที่วัดได้ของ Potentiometer แต่ละวนิด
ตัวแปรควบคุม	: แรงดันไฟฟ้าที่จ่ายเข้าบอร์ด ยึดห้องและรุ่นของ Potentiometer แต่ละวนิด

1.8.2 Linear Potentiometer

ตัวแปรต้น	: ระยะที่เลื่อน wiper ปรับ Potentiometer แต่ละวนิด
ตัวแปรตาม	: แรงดันไฟฟ้าจากที่วัดได้ของ Potentiometer แต่ละวนิด
ตัวแปรควบคุม	: แรงดันไฟฟ้าที่จ่ายเข้าบอร์ด ยึดห้องและรุ่นของ Potentiometer แต่ละวนิด

1.9 ผลการทดลอง

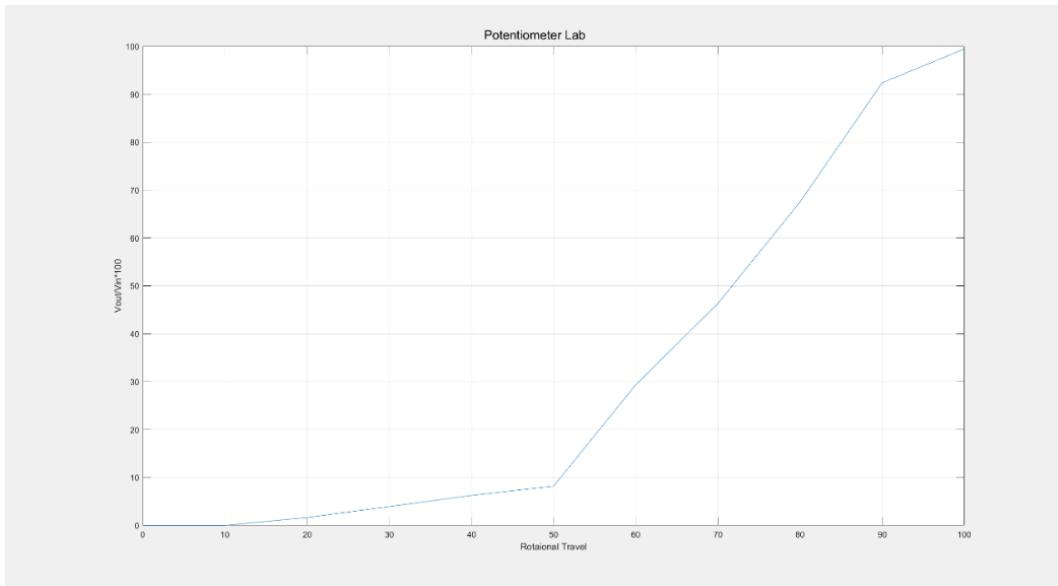
1.9.1 ผลการทดลองที่ได้จาก Potentiometer

หลังจากที่คณะผู้จัดทำได้เขียนโปรแกรมวัดกราฟในโปรแกรม MATLAB และโดยกำหนดให้แกน x คือ Rotational Travel และแกน y คือค่าเบอร์เซ็นต์ของ $\frac{V_{out}}{V_{in}}$

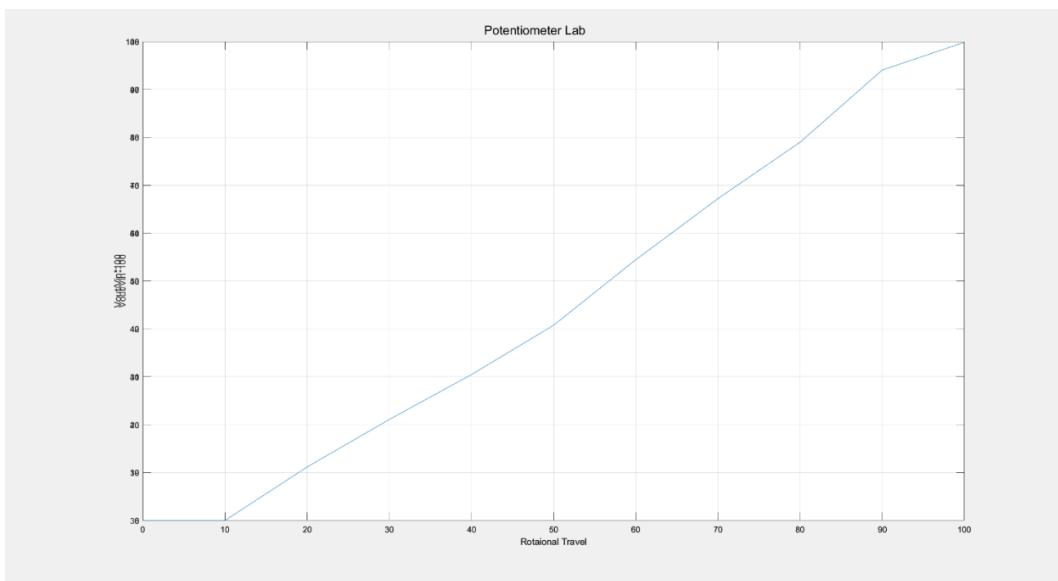
1.) Rotary Potentiometer

ผลปรากฏว่า Rotary Potentiometer ชนิด A ให้กราฟออกมาเป็น Exponential growth หรือกราฟของพิงก์ชันเอกซ์โพเนนเชียลที่มีค่าค่ออยู่ ๆ เพิ่มขึ้น ในขณะที่ Rotary Potentiometer ชนิด B ให้กราฟ

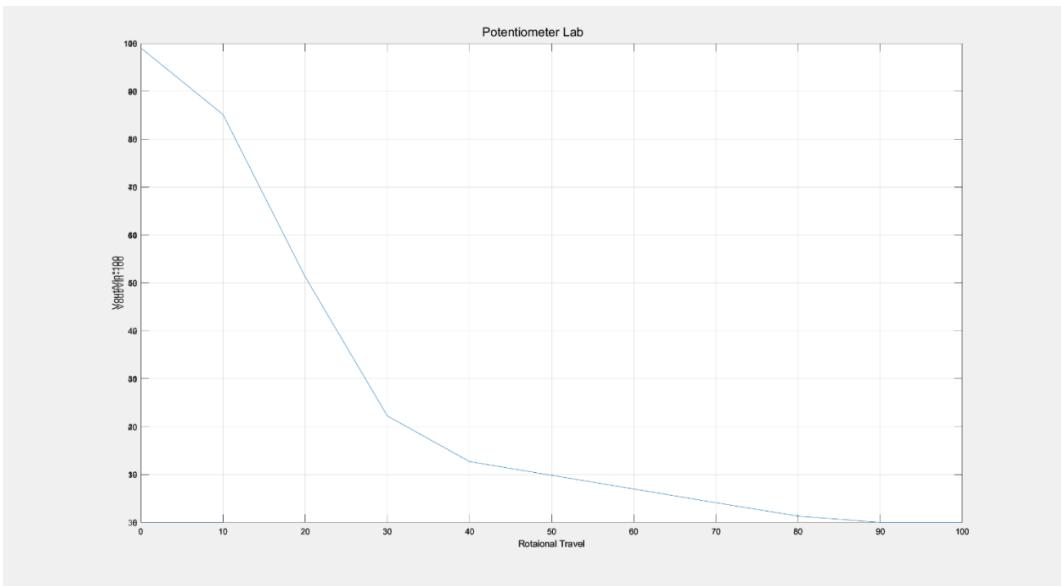
เป็นแบบ Linear graph หรือกราฟเชิงเส้น และ Rotary Potentiometer ชนิด C ที่ให้กราฟออกมานี้เป็น Exponential decay หรือกราฟของฟังก์ชันเอกซ์โพเนนเชียลที่มีค่าลดลง



รูปที่ 19 กราฟของ Rotary Potentiometer ชนิด A



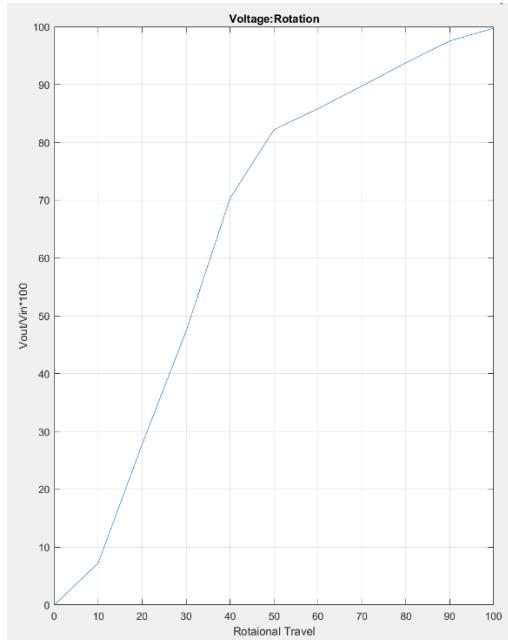
รูปที่ 20 กราฟของ Rotary Potentiometer ชนิด B



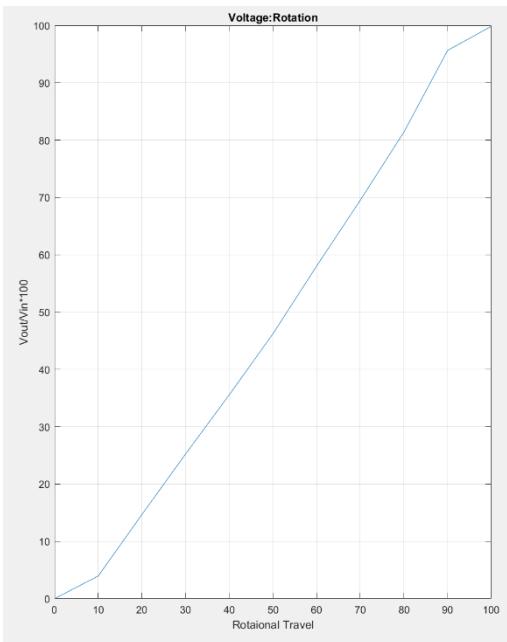
รูปที่ 21 กราฟของ Rotary Potentiometer ชนิด C

2.) Linear Potentiometer

ผลปรากฏว่า Linear Potentiometer A ให้กราฟออกแบบเป็น Audio Taper (Exponential graph) และ B ให้กราฟออกแบบเป็น Linear Taper (Linear graph) หรือกราฟเชิงเส้น



รูปที่ 22 กราฟของ Linear Potentiometer A



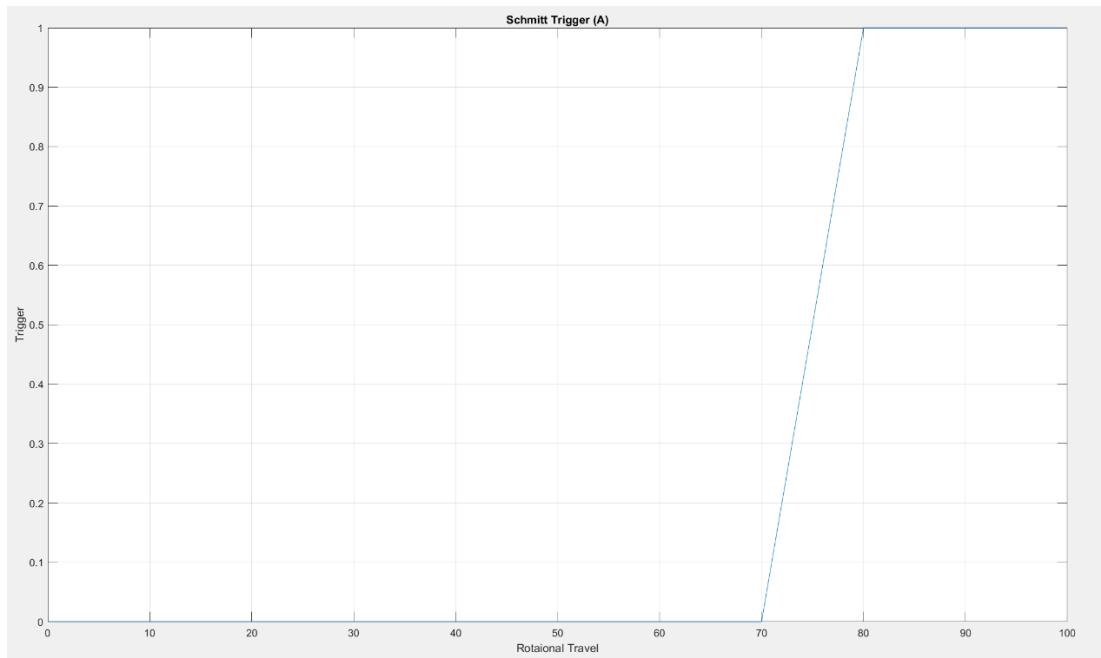
รูปที่ 23 กราฟของ Linear Potentiometer B

3.) Schmitt trigger

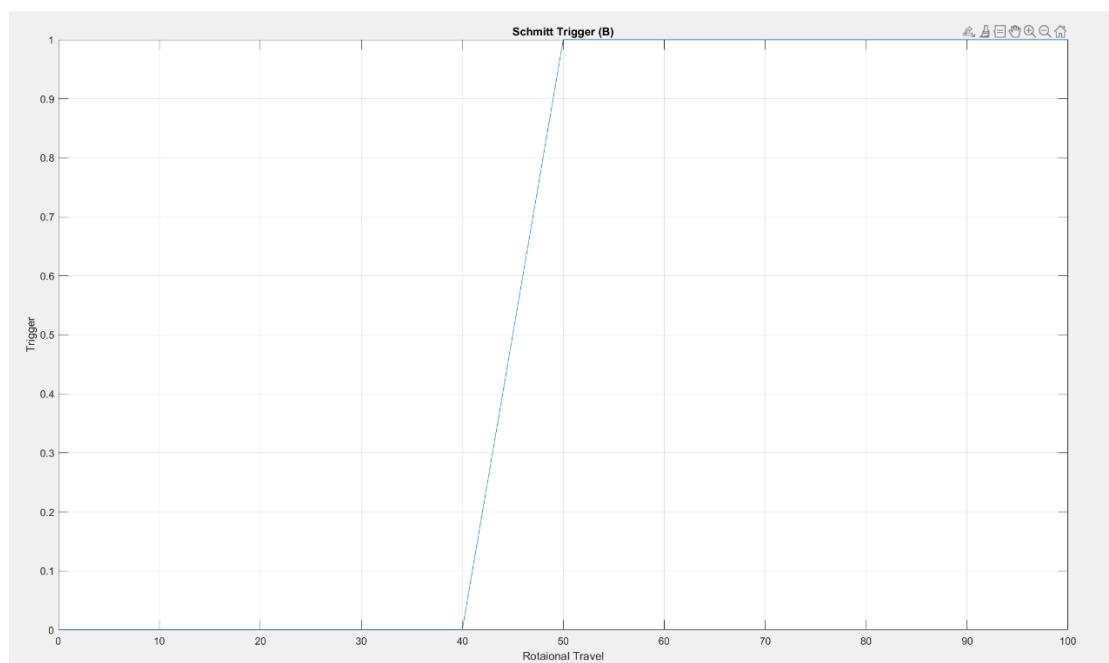
วงจร Schmitt-trigger เป็นวงจรที่เปลี่ยนสัญญาณแรงดันไฟฟ้า Analog เป็น Digital ได้เมื่อแรงดันไฟฟ้าถึงค่าความต่างศักย์ที่ตั้งไว้ (Threshold Voltage) โดยสัญญาณจะเปลี่ยนจากต่ำไปสูงหรือจากสูงไปต่ำ

คณะกรรมการจัดทำยังได้ทำการวาดกราฟ Schmitt trigger ของ Rotary Potentiometer ทั้ง 3 ชนิด และ Linear Potentiometer ทั้ง 2 ชนิด โดยกำหนด Threshold Voltage อยู่ที่ 1.65V โดยกำหนดให้แกน x คือ Rotational Travel และแกน y คือค่า Trigger

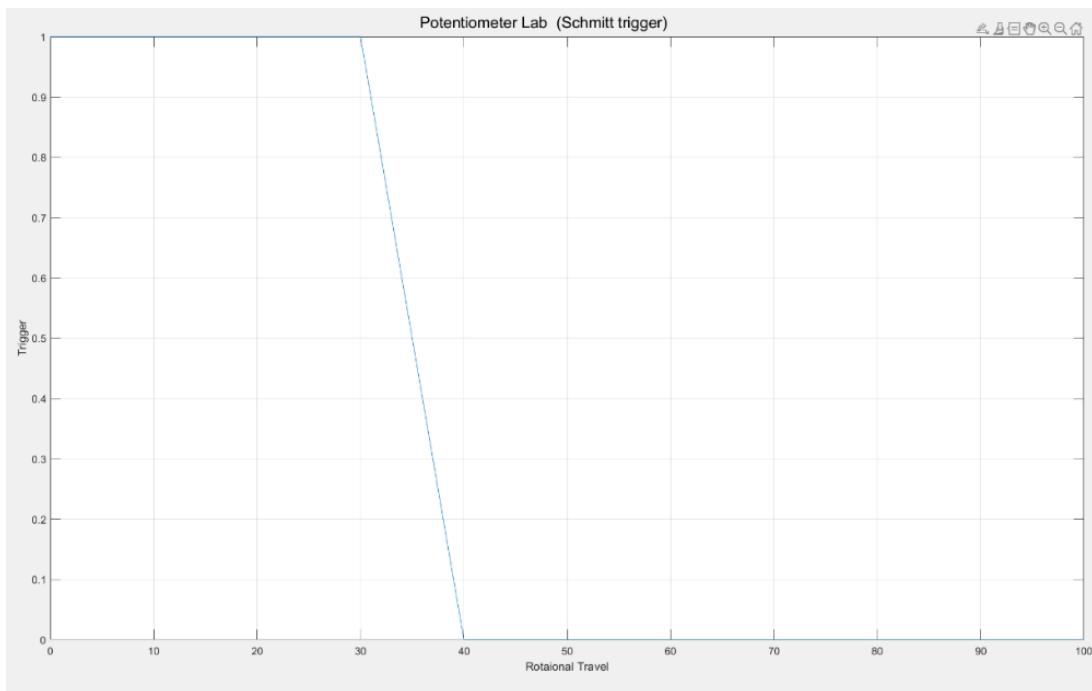
พบว่าในส่วนของ Rotary Potentiometer กราฟ Schmitt trigger ของชนิด A เกิดการเปลี่ยนแปลงค่า trigger จาก 0 ไปที่ 1 เมื่อ Rotational Travel มีค่าอยู่ที่ 70 หน่วย ในขณะที่กราฟ Schmitt trigger ของชนิด B เกิดการเปลี่ยนแปลงค่า trigger จาก 0 ไปที่ 1 เมื่อ Rotational Travel มีค่าอยู่ที่ 40 หน่วย แต่กราฟ Schmitt trigger ของชนิด C กลับเกิดการเปลี่ยนแปลงค่า trigger จาก 1 ไปที่ 0 เมื่อ Rotational Travel มีค่าอยู่ที่ 30 หน่วย



รูปที่ 24 Schmitt trigger ของ Rotary Potentiometer ชนิด A

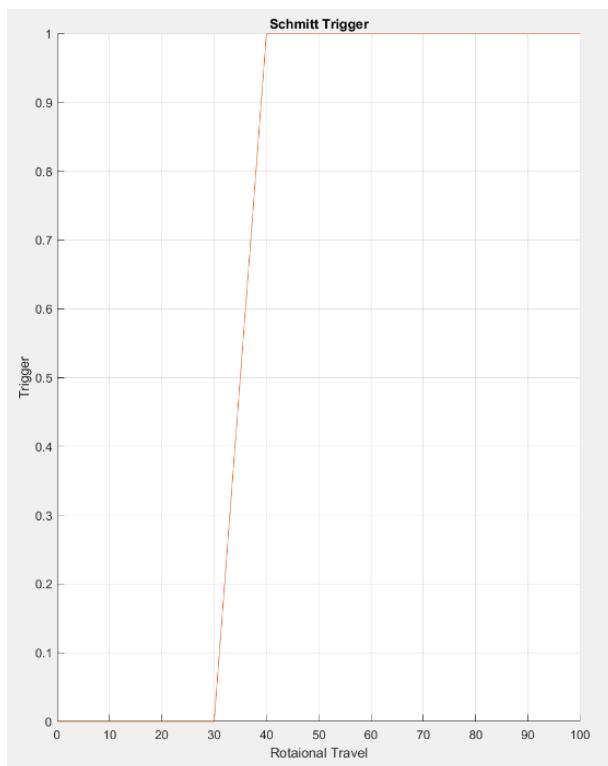


รูปที่ 25 Schmitt trigger ของ Rotary Potentiometer ชนิด B

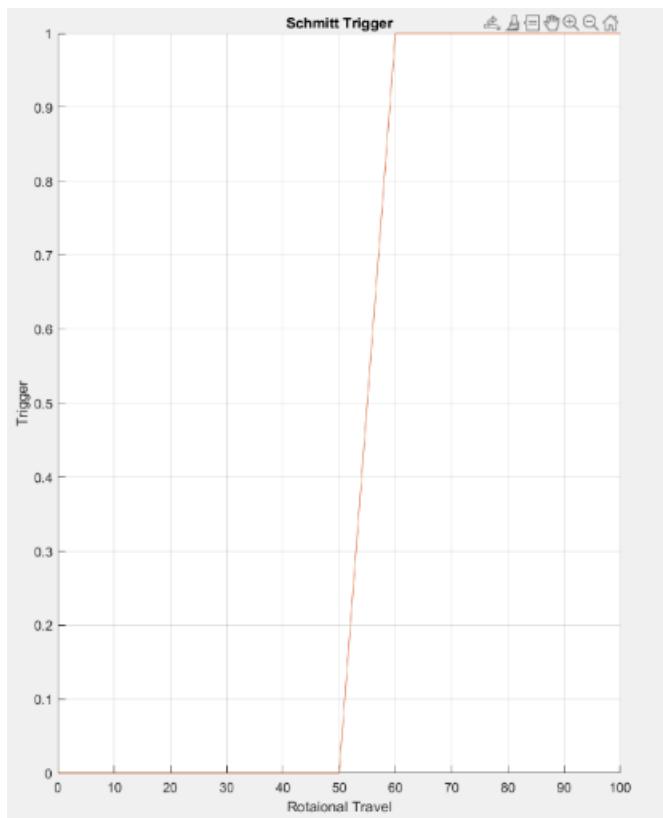


รูปที่ 26 Schmitt trigger ของ Rotary Potentiometer ชนิด C

นอกจากรูปที่ 25 ที่เป็นส่วนของ Linear Potentiometer กราฟ Schmitt trigger ของชนิด A เกิดการเปลี่ยนแปลงค่า trigger จาก 0 ไปที่ 1 เมื่อ Rotational Travel มีค่าอยู่ที่ 30 หน่วย และกราฟ Schmitt trigger ของชนิด B เกิดการเปลี่ยนแปลงค่า trigger จาก 0 ไปที่ 1 เมื่อ Rotational Travel มีค่าอยู่ที่ 50 หน่วย



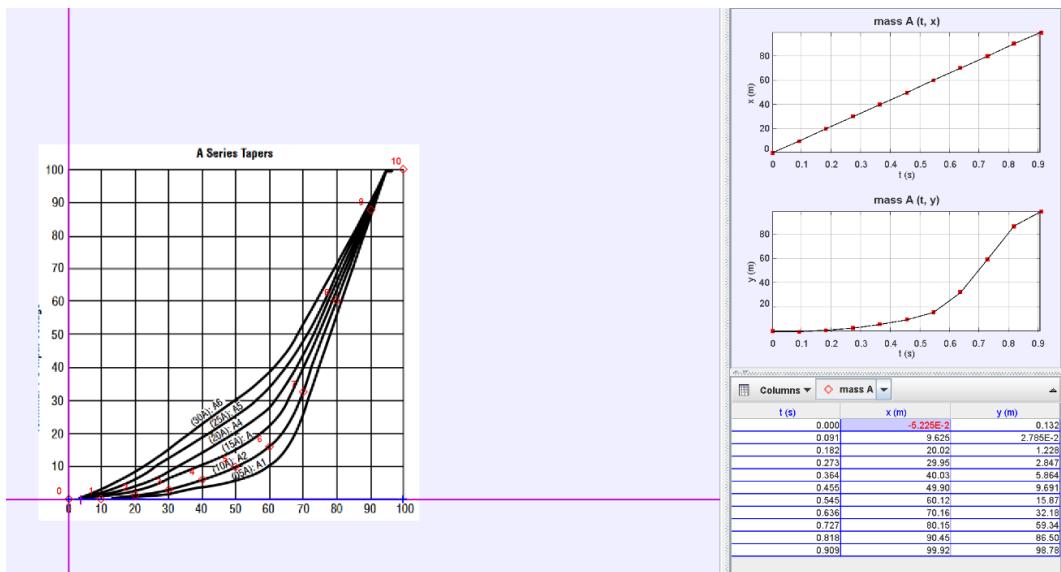
รูปที่ 27 กราฟ Schmitt trigger ของ Linear Potentiometer A



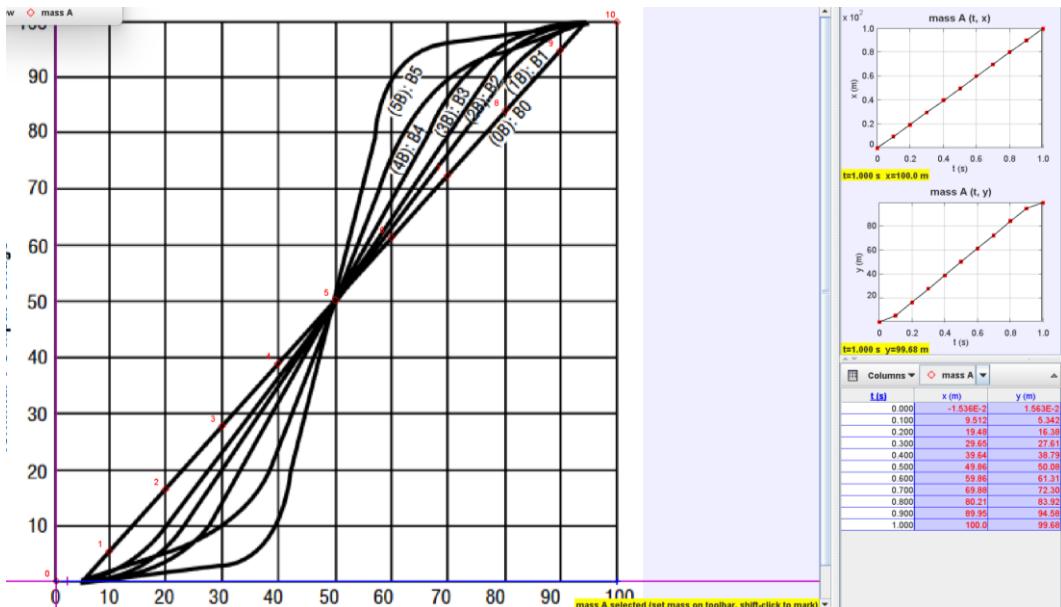
รูปที่ 28 กราฟ Schmitt trigger ของ Linear Potentiometer B

1.9.2 ผลการทดลองที่ได้จากการวิเคราะห์ Datasheet ของ Potentiometer ต่าง ๆ

ทางคณะผู้จัดทำได้นำกราฟจาก Datasheet ของ Rotary Potentiometer และ Linear Potentiometer ชนิดต่าง ๆ เข้าโปรแกรม Tracker เพื่อเก็บค่า Ideal ของ Potentiometer ที่อยู่บนบอร์ด และนำค่าที่ได้ไปวาดกราฟค่าความคลาดเคลื่อน



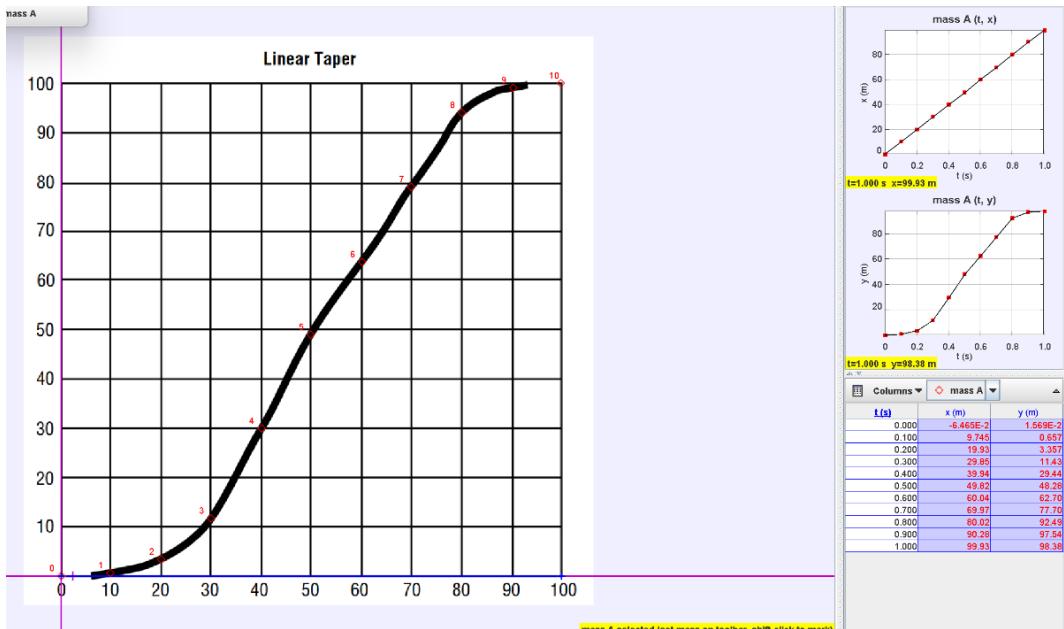
รูปที่ 29 ใช้โปรแกรม Tracker เก็บค่า Ideal จาก Datasheet ของ Rotary Potentiometer ชนิด A



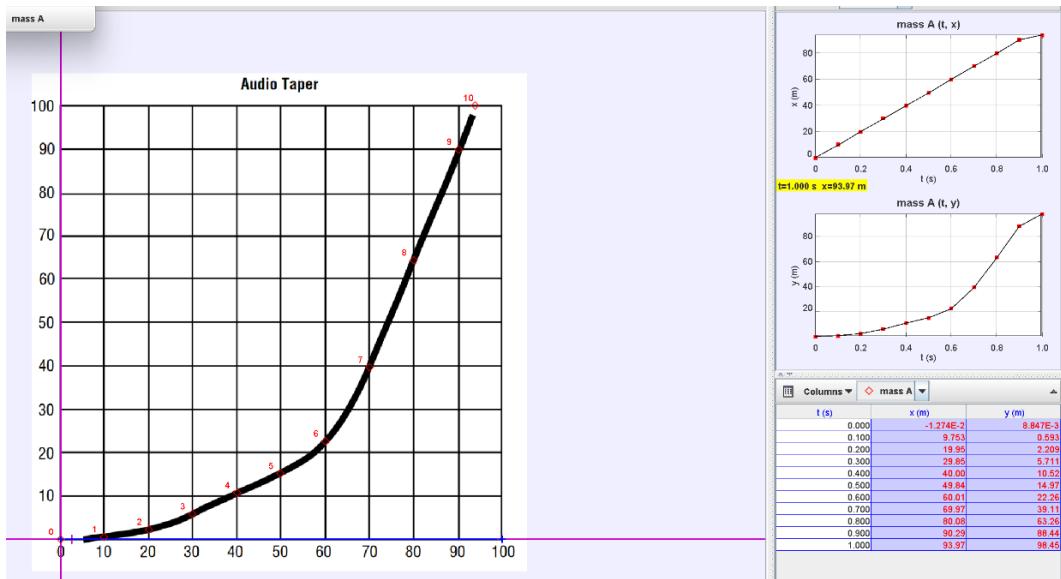
รูปที่ 30 ใช้โปรแกรม Tracker เก็บค่า Ideal จาก Datasheet ของ Rotary Potentiometer ชนิด B



รูปที่ 31 ใช้โปรแกรม Tracker เก็บค่า Ideal จาก Datasheet ของ Rotary Potentiometer ชนิด C



รูปที่ 32 ใช้โปรแกรม Tracker เก็บค่า Ideal จาก Datasheet ของ Linear Potentiometer Linear Taper



รูปที่ 33 ใช้โปรแกรม Tracker เก็บค่า Ideal จาก Datasheet ของ Linear Potentiometer Audio Taper

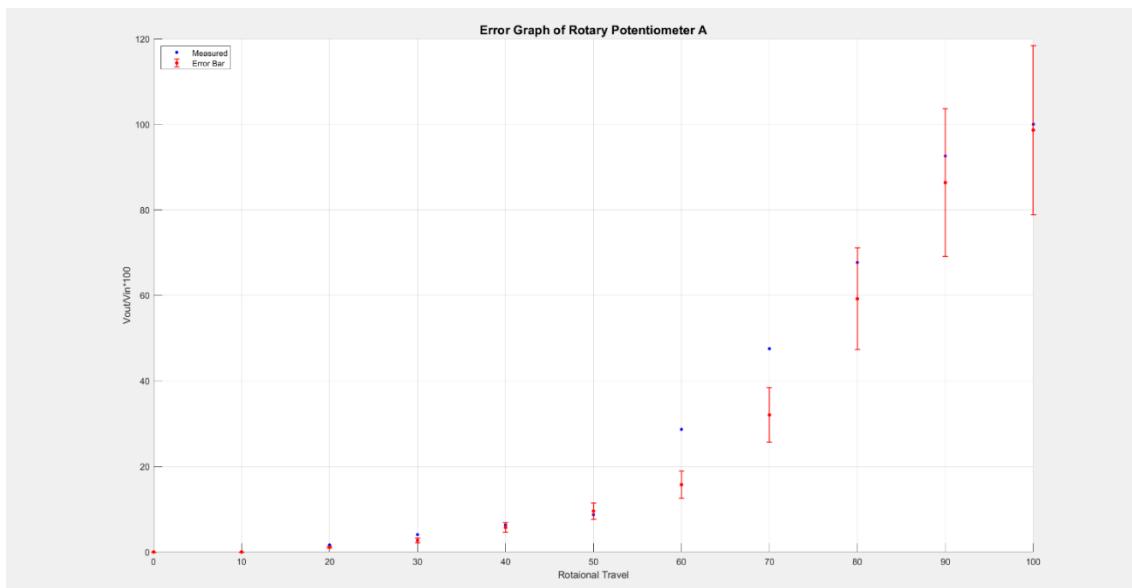
1.10 สรุปผล

จากการทดลองสรุปได้ว่า ค่าที่ได้จากการทดลองหมุนของ Rotary Potentiometer และการขับเลื่อนของ Linear Potentiometer ที่มีชนิดที่ต่างกันส่งผลต่อแรงดันไฟฟ้าในขาออกที่ต่างกันออกไป และเมื่อนำค่าที่ได้มาวัดเป็นกราฟ ผลปรากฏว่าได้กราฟออกตามข้อมูลที่มีอยู่ใน Datasheet โดย Rotary Potentiometer ที่อยู่บนบอร์ดจะมีอยู่ 3 ประเภทได้แก่ Type A, Type B, และ Type C ส่วน Linear Potentiometer จะมีอยู่ 2 ประเภทได้แก่ Audio Taper และ Linear Taper

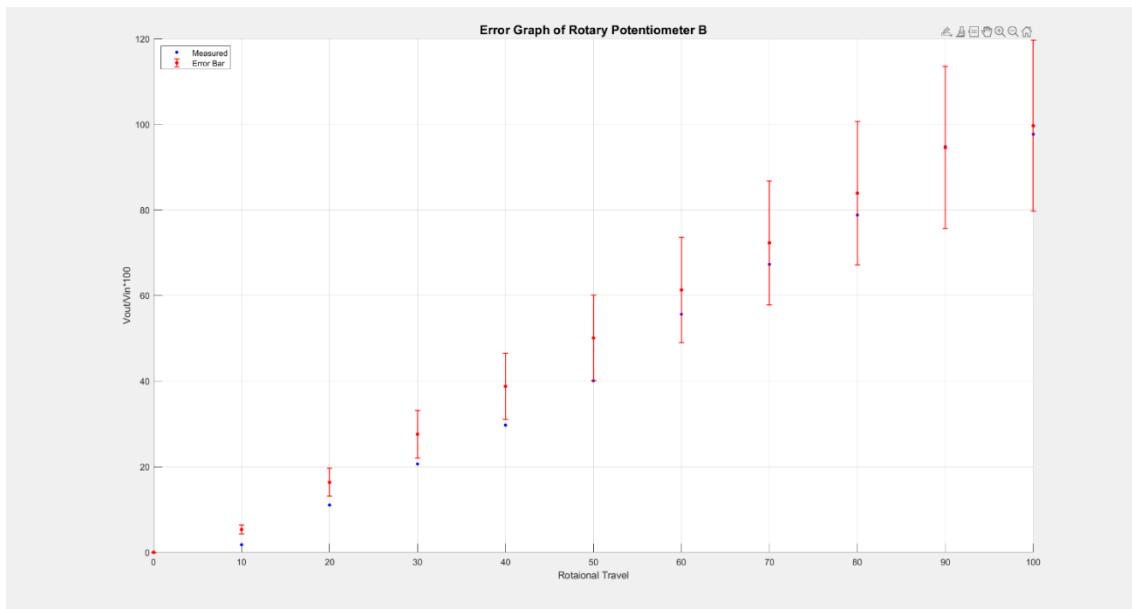
โดย Rotary Potentiometer A , Rotary Potentiometer C และ Linear Potentiometer A จะมีการตอบสนองในรูปแบบตามสมการ Exponential ส่วน Rotary Potentiometer B และ Linear Potentiometer B จะมีการตอบสนองในรูปแบบตามสมการเชิงเส้น (Linear)

1.11 อภิปรายผล

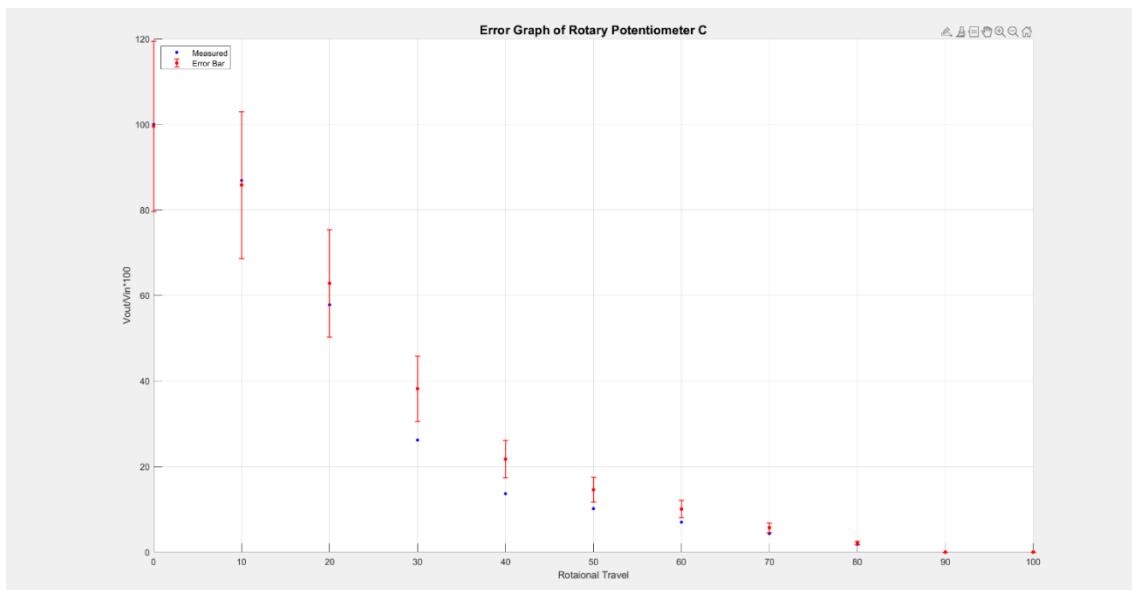
จากการทดลองจะพบได้ว่า ค่าที่ได้จากการทดลองนั้น มีทั้งค่าที่อยู่ในเกณฑ์ และบางค่าที่ไม่อยู่ในเกณฑ์ของ Datasheet แต่ละชนิด ซึ่งสาเหตุที่อาจทำให้เกิดค่าคลาดเคลื่อนได้ก็มีอยู่หลายอย่าง ทั้งจากองศาในการหมุน หรือ ระยะทางการขับของ Potentiometer ที่ไม่ตรงตามที่ออกแบบการทดลองไว้ เนื่องจากสเกลที่อยู่บนบอร์ดอาจไม่ตรงกับตำแหน่งของ Potentiometer พอดี หรือแม้แต่การใช้สายจ้มที่มีปัญหาทำให้วงจรมีค่าความต้านทานที่คลาดเคลื่อนได้เช่นกัน



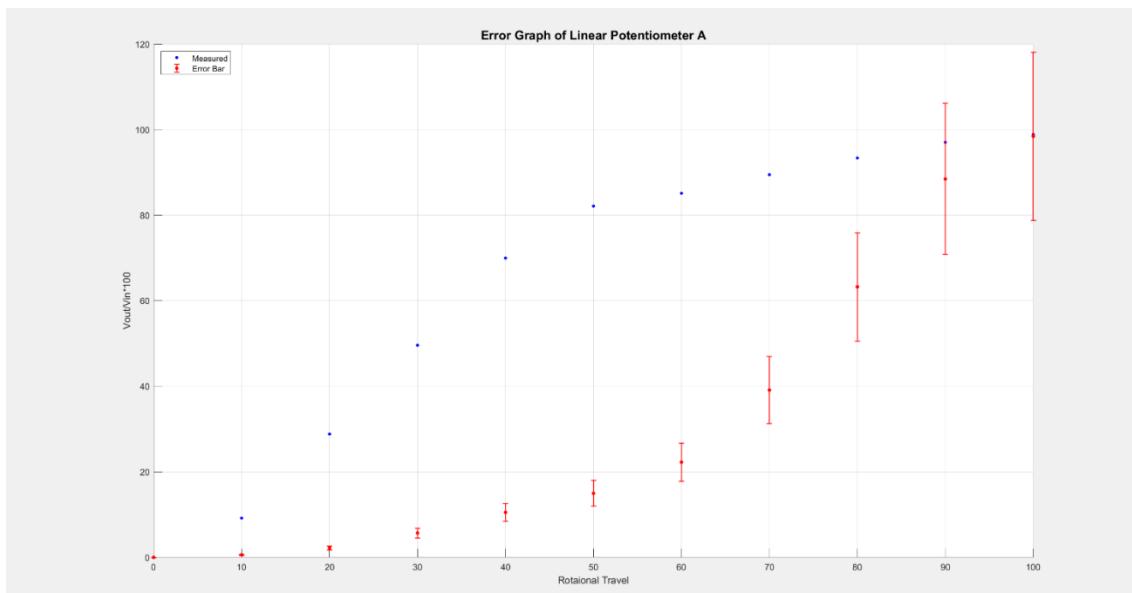
รูปที่ 34 กราฟแสดงค่าความคลาดเคลื่อนของ Rotary Potentiometer Type A



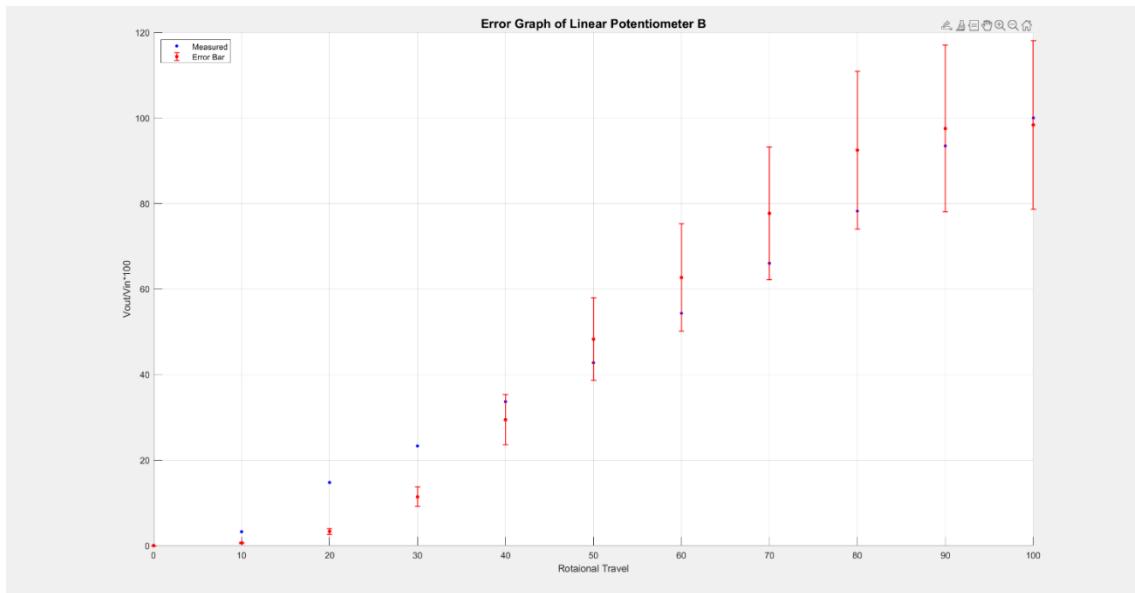
รูปที่ 35 กราฟแสดงค่าความคลาดเคลื่อนของ Rotary Potentiometer Type B



รูปที่ 36 กราฟแสดงค่าความคลาดเคลื่อนของ Rotary Potentiometer Type C



รูปที่ 37 กราฟแสดงค่าความคลาดเคลื่อนของ Linear Potentiometer Audio Taper



รูปที่ 38 กราฟแสดงค่าความคลาดเคลื่อนของ Linear Potentiometer Linear Taper

จาก Datasheet ของ Potentiometer ค่ามาตรฐานความคลาดเคลื่อนของตัวต้านทาน (Standard Resistance Tolerance) ใน Potentiometer มีค่าอยู่ที่ $\pm 20\%$ นอกจากนี้ยังมีเรื่องของสัญญาณรบกวนที่เกิดขึ้นขณะที่ทำการทดลองมาเกี่ยวข้องด้วยเช่นกัน ดังนั้นจากปัจจัยที่ได้กล่าวในข้างต้นจึงทำให้ผลการทดลองเกิดความคลาดเคลื่อนในการจัดเก็บข้อมูลบ้าง

1.12 ข้อเสนอแนะ

- 1.12.1 ทำการเก็บค่าให้ละเอียดมากขึ้น โดยแบ่งช่วงให้ละเอียดขึ้น
- 1.12.2 ทำงาน Low pass Filter เพิ่มใน Simulink เพื่อลดสัญญาณรบกวนที่เกิดขึ้น

1.13 เอกสารอ้างอิง

- 1.13.1 Datasheet Rotary Potentiometer
- 1.13.2 ความแตกต่างของสัญญาณ Analog และ digital
- 1.13.3 Datasheet Linear Potentiometer
- 1.13.4 All You Need to Know About Potentiometers
- 1.13.5 Schmitt trigger
- 1.13.6 Error bar Matlab

2. Incremental Encoder

2.1 จุดประสงค์

- 2.1.1 สามารถอธิบายหลักการทำงานของ Incremental Encoder และลักษณะสัญญาณ Output ของ Incremental Encoder เมื่อทิศทางและความเร็วการหมุนเปลี่ยนแปลงไปได้ และอธิบายรูปแบบเฟสของ สัญญาณในแต่ละช่อง Output
- 2.1.2 อธิบายการตั้งค่า Config Parameter ในระบบ IOC และการตั้งค่าโหมด Encoder บน STM32 เพื่อรับ การอ่านค่าตำแหน่งและความเร็วของมอเตอร์
- 2.1.3 สามารถอ่านค่า Raw Signal ของ QEI Method และ แปลงเป็น Relative Position (pulses) ด้วยการ Integrate Discrete Velocity (diff_counts) และแสดงผลพร้อมกันได้
- 2.1.4 นักศึกษาต้องสามารถเขียนโปรแกรม โดยประยุกต์ใช้ MATLAB และ Simulink ในการสั่งการหรือรับค่า ร่วมกับบอร์ด Nucleo STM32G474RE โดยใช้สัญญาณจากการหมุน Incremental Encoder ด้วยมีอเป็น Input และ แสดงสัญญาณ Output จากการ Log สัญญาณ แสดงผลเป็นกราฟจาก Data Inspector ใน MATLAB Simulink และดึงให้เห็นว่าสัญญาณ Output ประผันตามสัญญาณ Input แบบ Real Time โดยมี Output เป็น Raw Signal, Angular Position, Angular Velocity ในหน่วย SI
- 2.1.5 สามารถหาค่า Pulses Per Revolution และ Resolution ของ Incremental Encoder จากการทดลอง
- 2.1.6 อธิบายการอ่านค่า Quadrature Encoder ทุกรูปแบบด้วยวิธี QEI และ Polling Method พร้อมพิจารณา ข้อจำกัดและพฤติกรรมของการอ่านค่าด้วย Polling และ QEI
- 2.1.7 นักศึกษาต้องสามารถเขียนโปรแกรมเพื่อ Homing Incremental Encoder ได้

2.2 สมมติฐาน

- 2.2.1 Incremental Encoder จะส่งสัญญาณ Output ที่สัมพันธ์กับการเปลี่ยนแปลงทิศทางและความเร็วการ หมุน โดยรูปแบบเฟสของสัญญาณในแต่ละช่อง Output จะเปลี่ยนแปลงตามทิศทางที่หมุนไป
- 2.2.2 การตั้งค่า Config Parameter ในระบบ IOC และการตั้งค่าโหมด Encoder บน STM32 สามารถรองรับ การอ่านค่าตำแหน่งและความเร็วของมอเตอร์ได้อย่างแม่นยำ
- 2.2.3 สามารถอ่านค่า Raw Signal ของ Incremental Encoder ด้วยวิธี QEI และแปลงเป็น Relative Position ได้ผ่านการ Integrate Discrete Velocity (diff_counts) ซึ่งจะแสดงผลสอดคล้องกับตำแหน่งจริง
- 2.2.4 สามารถใช้ MATLAB และ Simulink ร่วมกับบอร์ด Nucleo STM32G474RE เพื่อรับค่าการหมุนจาก Incremental Encoder และแสดงผลเป็นกราฟแบบ Real Time ซึ่งจะแสดงผลในรูปแบบของ Raw Signal, Angular Position และ Angular Velocity ในหน่วย SI ได้อย่างต่อเนื่องตามสัญญาณ Input ที่ เปลี่ยนแปลง
- 2.2.5 สามารถคำนวณค่า Pulses Per Revolution และ Resolution ของ Incremental Encoder ได้อย่าง ถูกต้อง และค่าเหล่านี้จะสอดคล้องกับความละเอียดของสัญญาณ Output ที่วัดได้

2.2.6 วิธี QEI และ Polling Method สามารถอ่านสัญญาณ Output ของ Quadrature Encoder ได้ทุกรูปแบบ โดยแต่ละวิธีมีข้อจำกัดและพุติกรรมที่แตกต่างกัน ซึ่ง QEI คาดว่าจะให้ผลลัพธ์ที่ต่อเนื่องกว่า Polling Method

2.2.7 สามารถเขียนโปรแกรมเพื่อทำ Homing ให้กับ Incremental Encoder เพื่อกำหนดจุดเริ่มต้นของตำแหน่งได้อย่างแม่นยำ

2.3 นิยามศัพท์เฉพาะ

- Incremental Encoder คือ อุปกรณ์ที่ใช้ในการตรวจจับการหมุนหรือตำแหน่งของวัตถุ โดยให้สัญญาณแบบต่อเนื่องเป็นพลส์ (pulses) ซึ่งสามารถใช้ระบุตำแหน่งเชิงมุมและทิศทางการหมุนได้ โดยไม่ต้องบอกร่องรอยเดินทาง เนื่องจากต้องมีการหาจุดเริ่มต้น (Homing) ในการเริ่มต้นการใช้งาน
- QEI (Quadrature Encoder Interface) คือ วิธีการหนึ่งในการอ่านสัญญาณของ Encoder โดยสามารถอ่านได้ทั้งขอบขาขึ้นและขาลงของสัญญาณ A และ B เพื่อเพิ่มความละเอียดของการนับพลส์ และยังช่วยระบุทิศทางการหมุนได้
- Polling Method คือ วิธีการอ่านสัญญาณ Encoder โดยการตรวจสอบ (polling) สถานะของพอร์ต 1/0 เป็นช่วง ๆ ตาม timer ที่ตั้งค่าไว้ ซึ่งต่างจาก QEI ที่อ่านสัญญาณได้แบบต่อเนื่อง Polling Method มักมีประสิทธิภาพต่ำกว่าและอาจทำให้เกิดการสูญเสียพลส์หากความถี่ของสัญญาณสูง
- Config Parameter คือ ค่าพารามิเตอร์การตั้งค่าต่าง ๆ ที่สามารถปรับแต่งได้ในระบบควบคุม เช่น STM32 เพื่อให้เหมาะสมกับการใช้งาน Incremental Encoder ในการกำหนดตำแหน่ง ความเร็ว และทิศทาง เช่น ค่า PPR, การกำหนดโหมดการอ่านค่าของ Encoder เป็นต้น
- Wrap คือ กระบวนการที่ค่าเชิงมุมของ Encoder ถูกจำกัดให้อยู่ในช่วง 0 ถึง 360 องศา (หรือในขอบเขตอื่น) และเมื่อตำแหน่งเกินขอบเขตที่กำหนดไว้จะถูกลบไปยังจุดเริ่มต้นใหม่
- Unwrap คือ กระบวนการทางคณิตศาสตร์ที่ใช้ในการแปลงค่ามุมที่วนรอบ (wrap-around) ให้เป็นมุมต่อเนื่อง ซึ่งช่วยให้การคำนวณตำแหน่งเชิงมุมใน Encoder ที่ทำการหมุนหลายรอบมีความถูกต้องและต่อเนื่อง
- PPR (Pulses Per Revolution) คือ จำนวนพลส์ที่เกิดขึ้นเมื่อ Encoder หมุนครบหนึ่งรอบ มักเป็นตัวกำหนดความละเอียดของ Encoder โดยยิ่ง PPR สูง จะยิ่งมีความละเอียดในการอ่านตำแหน่งที่มากขึ้น
- Quadrature Encoder คือ Encoder ที่ให้สัญญาณสองช่อง A และ B ซึ่งมีเฟสต่างกัน 90 องศา และสามารถใช้กำหนดทิศทางและตำแหน่งได้
- Raw Signal คือ สัญญาณดิบที่ออกจาก Encoder โดยตรง ซึ่งยังไม่ได้ผ่านการประมวลผลหรือการแปลงใด ๆ มีลักษณะเป็นสัญญาณสี่เหลี่ยมจากช่องสัญญาณ A และ B ที่บ่งบอกถึงการเปลี่ยนแปลงของตำแหน่งและทิศทางของการหมุน

- Angular Position คือ ตำแหน่งเชิงมุมของ Encoder โดยปกติระบุเป็นมุมในหน่วยองศาหรือเรเดียนที่เปลี่ยนแปลงตามการหมุน
- Angular Velocity คือ ความเร็วเชิงมุม หรืออัตราการเปลี่ยนแปลงของตำแหน่งเชิงมุมตามเวลา เป็นหน่วยที่ใช้บอกอัตราการหมุนของ Encoder
- Relative Position คือ ตำแหน่งสัมพันธ์ที่นับจากจุดเริ่มต้นการอ่านค่าสุดท้ายของ Encoder โดยจะไม่มีจุดอ้างอิงแบบแน่นอน
- Resolution คือ ความละเอียดของ Encoder ในการวัดตำแหน่ง ซึ่งขึ้นอยู่กับค่า Pulses Per Revolution (PPR) ยิ่งค่ามากจะยิ่งวัดตำแหน่งได้ละเอียดมากขึ้น

2.4 นิยามเชิงปฏิบัติการ

- Input ของ QEI คือ Count สำหรับ QEI, Input ที่อ่านได้จาก Encoder จะถูกนับเป็น count ซึ่งหมายถึงจำนวนการเปลี่ยนแปลงของขอบสัญญาณ (อาจเป็นขอบขาขึ้นและขาลง) ทำให้ QEI สามารถอ่านตำแหน่งและทิศทางได้โดยอิงจากจำนวน count ที่สะสม
- Input ของ Polling Method คือ Pulses สำหรับ Polling Method, Input ที่อ่านได้จาก Encoder จะเป็น pulses ซึ่งเป็นสัญญาณที่เกิดการเปลี่ยนแปลงของสัญญาณ A หรือ B ซึ่ง
- Output คือ ค่า Output จาก Encoder จะประกอบด้วยตำแหน่งเชิงมุม (Angular Position) หรือความเร็วเชิงมุม (Angular Velocity) ซึ่งขึ้นอยู่กับการประมวลผลข้อมูลจาก Raw Signal หรือ count ที่ได้
- Discrete Velocity (diff_counts) คือ ค่าแตกต่างระหว่าง count แต่ละครั้งที่อ่านได้ ใช้สำหรับคำนวณความเร็วหรือการเปลี่ยนแปลงตำแหน่งแบบต่อเนื่อง
- Homing คือ กระบวนการตั้งจุดเริ่มต้นให้กับ Encoder เพื่อกำหนดตำแหน่งที่แน่นอนที่ตำแหน่งศูนย์ หรือจุดอ้างอิงเริ่มต้น
- counter period คือ ช่วงเวลาการนับหรือรอบของตัวนับ (counter) ในระบบหรือวงจรที่ใช้วัดเวลาหรือความถี่ของสัญญาณ โดย "Counter Period" นี้จะเป็นระยะเวลาที่ตัวนับใช้เพื่อตรวจสอบการเปลี่ยนแปลงของสัญญาณที่เข้ามา ในที่นี้เราทำการกำหนดโดย การนำค่ามากสุดที่ TIM เราอ่านได้ (16 bit) ลบกับ เศษจากการหาร 16-bit ด้วย PPR

2.5 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.5.1 What do X1, X2, and X4 position encoding mean for incremental encoders?

1) X1 Encoding

ในการอ่านค่าแบบ X1 ในการอ่านค่าแบบ X1 Encoding ของ Quadrature Encoder จะอ่านสัญญาณเพียง ขอบหนึ่งของแต่ละช่องสัญญาณ (เช่น ขอบขาขึ้นของสัญญาณ A หรือ B) ซึ่งเป็นรูปแบบพื้นฐาน

ที่สุดในการอ่าน สัญญาณจาก Encoder เพื่อหาตำแหน่งการหมุน โดยสามารถสังเกตได้จากการเปลี่ยนแปลงของสัญญาณดังนี้

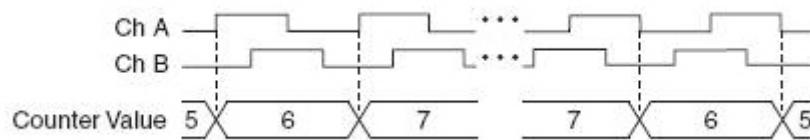
- ขอบขาขึ้นของสัญญาณ A: เมื่อสัญญาณ A เปลี่ยนจาก 0 เป็น 1 (ขอบขาขึ้น) เราจะดูค่าสัญญาณ B เพื่อกำหนดทิศทาง:

- ถ้าสัญญาณ B มีค่าเป็น 0 หมายความว่า Encoder หมุนในทิศทาง ไปข้างหน้า
- ถ้าสัญญาณ B มีค่าเป็น 1 หมายความว่า Encoder หมุนในทิศทาง ย้อนกลับ

- ขอบขาลงของสัญญาณ A : เมื่อสัญญาณ A เปลี่ยนจาก 1 เป็น 0 (ขอบขาลง) เราจะใช้สัญญาณ B เพื่อกำหนดทิศทางเช่นกัน:

- ถ้าสัญญาณ B มีค่าเป็น 1 แสดงว่า Encoder หมุนไปข้างหน้า
- ถ้าสัญญาณ B มีค่าเป็น 0 แสดงว่า Encoder หมุนย้อนกลับ

การอ่านค่าแบบ X1 จึงเน้นการอ่านเฉพาะขอบขาขึ้น หรือขอบขาลง ของสัญญาณ A และตรวจสอบสัญญาณ B เพียงหนึ่งครั้งต่อรอบ เพื่อให้ได้ข้อมูลตำแหน่งและทิศทางการหมุนเบื้องต้น



รูปที่ 39 X1 Encoding ของ Quadrature Encoder

2) X2 Encoding

ในการอ่านแบบ X2 Encoding เราจะอ่าน สองขอบของสัญญาณ A (ขอบขาขึ้นและขอบขาลง) โดยใช้สัญญาณ B เป็นตัวช่วยในการกำหนดทิศทาง ดังนี้

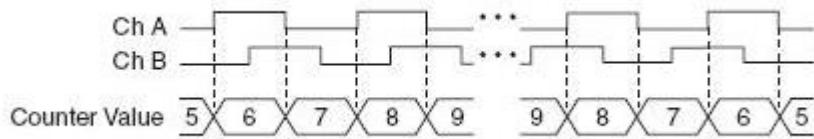
- ขอบขาขึ้นของสัญญาณ A: เมื่อสัญญาณ A เปลี่ยนจาก 0 เป็น 1 (ขอบขาขึ้น) เราจะดูค่าสัญญาณ B เพื่อกำหนดทิศทาง:

- ถ้าสัญญาณ B มีค่าเป็น 0 หมายความว่า Encoder หมุนในทิศทาง ไปข้างหน้า
- ถ้าสัญญาณ B มีค่าเป็น 1 หมายความว่า Encoder หมุนในทิศทาง ย้อนกลับ

- ขอบขาลงของสัญญาณ A : เมื่อสัญญาณ A เปลี่ยนจาก 1 เป็น 0 (ขอบขาลง) เราจะใช้สัญญาณ B เพื่อกำหนดทิศทางเช่นกัน:

- ถ้าสัญญาณ B มีค่าเป็น 1 แสดงว่า Encoder หมุนไปข้างหน้า
- ถ้าสัญญาณ B มีค่าเป็น 0 แสดงว่า Encoder หมุนย้อนกลับ

ในแบบ X2 การอ่านจะเกิดขึ้น สองครั้งต่อหนึ่งรอบของสัญญาณ A ทั้งของขาขึ้นและขอบขาลง ทำให้ความละเอียดของการอ่านเพิ่มขึ้นเป็นสองเท่าเมื่อเทียบกับ X1



รูปที่ 40 X1 Encoding ของ Quadrature Encoder

3) X4 Encoding

X4 Encoding เป็นการอ่านที่มีความละเอียดสูงสุด โดยจะอ่าน ทั้งขอบขาขึ้นและขาลงของสัญญาณ ทั้ง A และ B ทำให้ได้ข้อมูลการหมุนสี่ครั้งต่อหนึ่งรอบของสัญญาณ A ดังนี้:

- ขอบขาขึ้นของสัญญาณ A:

- ถ้า B เป็น 0 หมายถึง Encoder หมุนไปข้างหน้า
- ถ้า B เป็น 1 หมายถึง Encoder หมุนย้อนกลับ

- ขอบขาลงของสัญญาณ A:

- ถ้า B เป็น 1 หมายถึง Encoder หมุนไปข้างหน้า
- ถ้า B เป็น 0 หมายถึง Encoder หมุนย้อนกลับ

- ขอบขาขึ้นของสัญญาณ B:

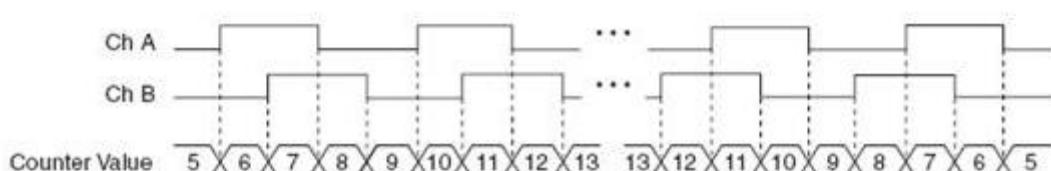
- ถ้า A เป็น 1 หมายถึง Encoder หมุนไปข้างหน้า
- ถ้า A เป็น 0 หมายถึง Encoder หมุนย้อนกลับ

- ขอบขาลงของสัญญาณ B:

- ถ้า A เป็น 0 หมายถึง Encoder หมุนไปข้างหน้า
- ถ้า A เป็น 1 หมายถึง Encoder หมุนย้อนกลับ

ใน X4 Encoding การอ่านเกิดขึ้น สี่ครั้งต่อหนึ่งรอบของสัญญาณ A ทำให้ความละเอียดของการอ่าน

สูงสุด



รูปที่ 41 X1 Encoding ของ Quadrature Encoder

2.5.2 การอ่านค่า STM32G4 timer instance features

STM32G4 timer instance features

52

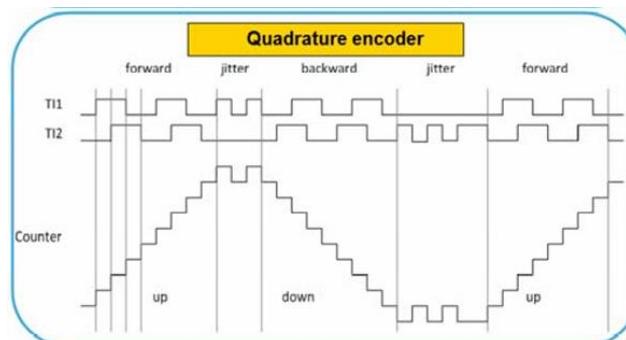
Feature	TIM1	TIM8	TIM20	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM15	TIM16	TIM17
	(Advanced Control)			(General-Purpose)			(Basic)		(General-Purpose)			
Clock source	CK_INT External input pin External trigger input ETR			CK_INT External input pin External trigger input ETR Internal trigger inputs			CK_INT		CK_INT External input pin Internal trigger inputs		CK_INT External input pin	
Resolution	16-bit			32-bit	16-bit	16-bit	32-bit	16-bit			16-bit	
Prescaler				16-bit								
Counter direction	Up, Down, Up&Down			Up, Down, Up&Down			Up		Up			
Repetition counter	✓			-			-		✓			
Synchronization	Master	✓		✓			✓		✓			
	Slave	✓		✓			-		-			
Number of channels	6: ➢ CH1/CH1N ➢ CH2/CH2N ➢ CH3/CH3N ➢ CH4/CH4N ➢ CH5 and CH6 output only, not available externally			4: ➢ CH1 ➢ CH2 ➢ CH3 ➢ CH4			0		2: ➢ CH1/CH1N ➢ CH2		1: ➢ CH1/CH1N	
Trigger input	✓			✓			-		✓		✓	

รูปที่ 42 STM32G4 timer instance features

2.5.3 Quadrature encoder

บางตัวจับเวลาใน STM32 รองรับ荷มดการนับขึ้น/ลง เช่น ตัวจับเวลาควบคุมขั้นสูง (timers) 1, 8, 20 และตัวจับเวลาทั่วไป 2, 3, 4, 5 โดยทิศทางการนับสามารถตั้งค่าได้ด้วยซอฟต์แวร์หรือจัดการโดยอัตโนมัติเมื่ออยู่ในโหมด PWM แบบจัดศูนย์ (center-aligned PWM mode)

ในโหมดนี้ ทิศทางการนับจะเปลี่ยนโดยอัตโนมัติเมื่อเกิดการล้นหรือขาดของเคาน์เตอร์ (counter overflow/underflow) ซึ่งช่วยลดเสียงรบกวนทางอะคูสติก โดยเพิ่มความถี่การรับกวนของกระแสน้ำให้เป็นสองเท่าจากความถี่ PWM เดิม ทำให้ได้สมดุลที่เหมาะสมสมรรถห่วงการสัญญาณจากการสวิตช์และการลดเสียงรบกวน ทิศทางการนับยังสามารถจัดการอัตโนมัติได้เมื่อจับเวลาอยู่ในโหมด Encoder



รูปที่ 43 Quadrature encoder

2.6 วิธีดำเนินการทดลอง

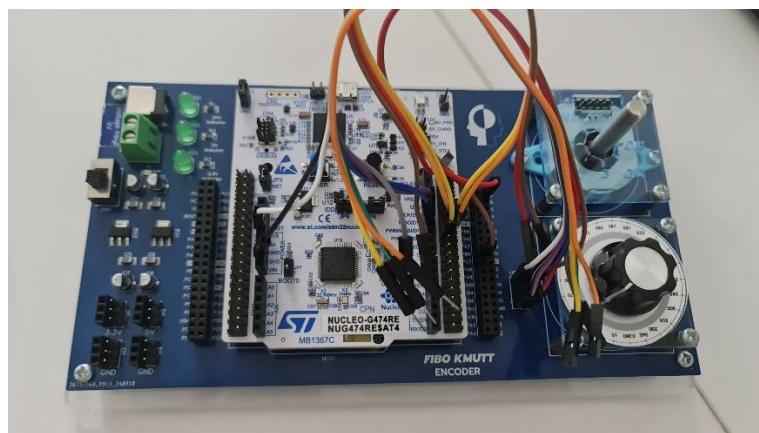
2.6.1 เริ่มจากทำการหาค่า PPR ของ Encoder ของทั้ง 2 ตัว โดยทำการอ่านค่า Count ที่ได้จากการหมุน Encoder ไปครอบ 1 รอบ โดยทำการเก็บค่าทั้งหมด 3 ครั้ง

2.6.2 นำค่า PPR ที่ได้มาใช้ในการทดลองต่อโดยใช้ในการกำหนดค่า counter period โดย เอาค่าความละเอียดที่ TIM เราอ่านได้ (16 bit) ลบกับ เศษจากการการ 16 bit ด้วย PPR

2.6.3 ทำการเก็บค่าและคำนวณข้อ 2.6.1 และ 2.6.2 ของ Encoder ทั้ง 2 ตัว

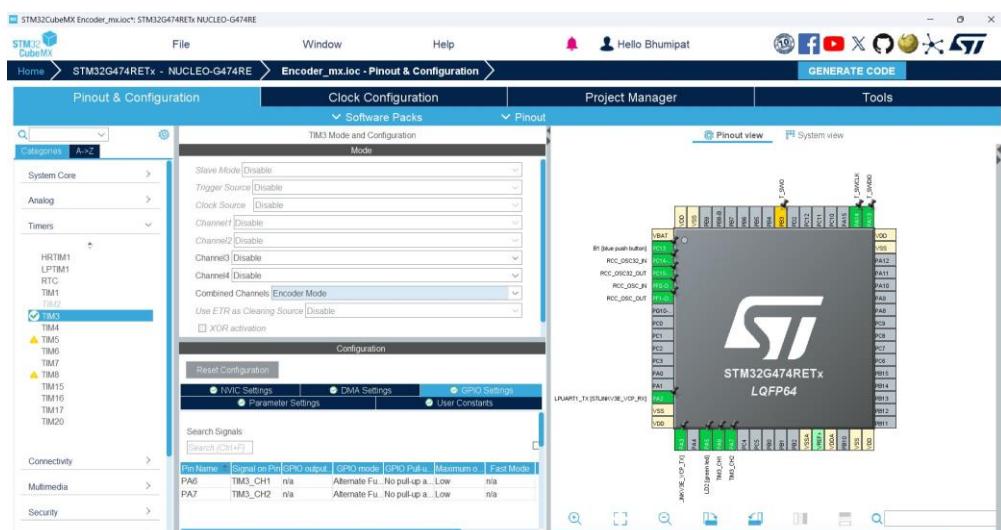
2.6.4 ทำการต่อสายและตั้งค่า MATLAB Simulink โดยเริ่มจากการตั้งค่า pin ต่าง ๆ ว่าทำงานอย่างไร ซึ่งจากข้อมูล STM32G4 timer instance features โดยกำหนด TIM ที่ใช้เป็น TIM3, 4, 8 จาก Resolution เป็น 16-bit และ Counter direction เป็น Up, Down, Up&Down และต่อสายดังนี้

- TIM 3 เป็นการอ่านค่า X1 โดย CH1 เป็น PIN PA6 และ CH2 เป็น PIN PA7
- TIM 3 เป็นการอ่านค่า X2 โดย CH1 เป็น PIN PB11 และ CH2 เป็น PIN PB12
- TIM 3 เป็นการอ่านค่า X4 โดย CH1 เป็น PIN PC6 และ CH2 เป็น PIN PC7

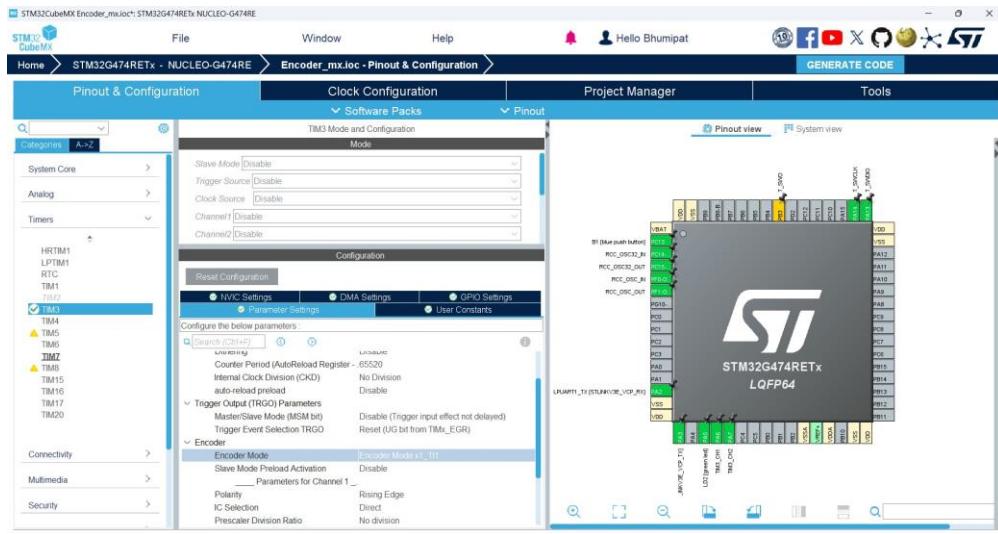


รูปที่ 44 รูปการต่อสายกับบอร์ด

2.6.5 จานนั้นตั้งค่า PIN ใน CubeMX ดังภาพ



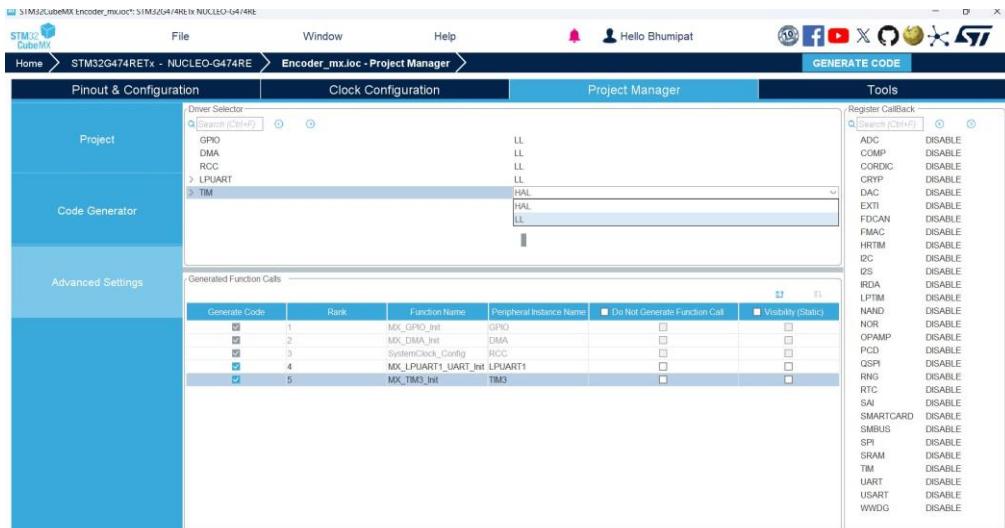
รูปที่ 45 การตั้งค่า Tim ใน CubeMX



รูปที่ 46 การตั้งค่า Tim ใน CubeMX (ต่อ)

ทำการตั้งค่าพิมพ์ต่างๆโดยที่กล่าวไว้ได้เลือกใช้ TIM 3 TIM 5 และกี TIM 8 โดย ตั้งค่า Combined Mode ให้เป็น Encode Mode

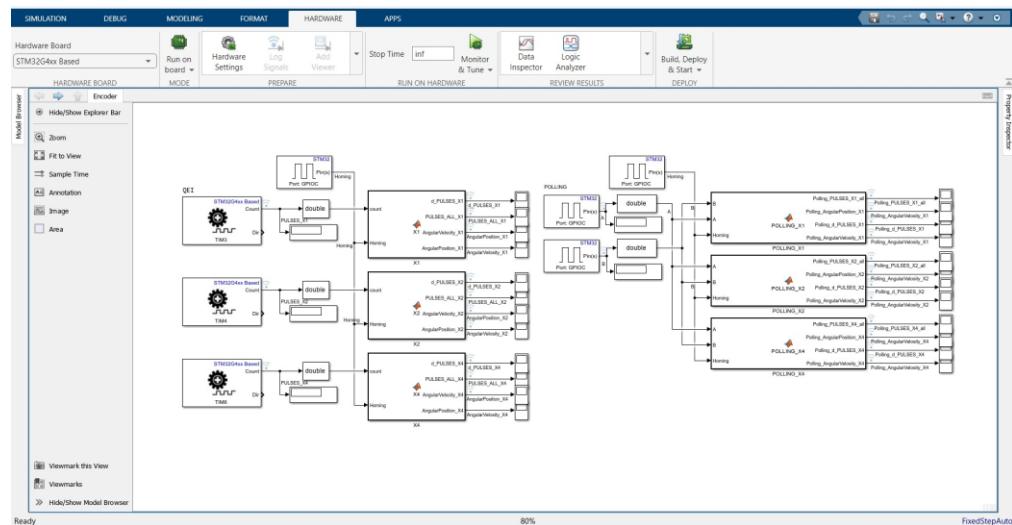
และเข้าไป Set ค่า Parameter Settings เป็น Encode Modex1_Tl1 ,Encode Mode_Tl1 และ Encode Mode Tl1and Tl2 ตามลำดับ



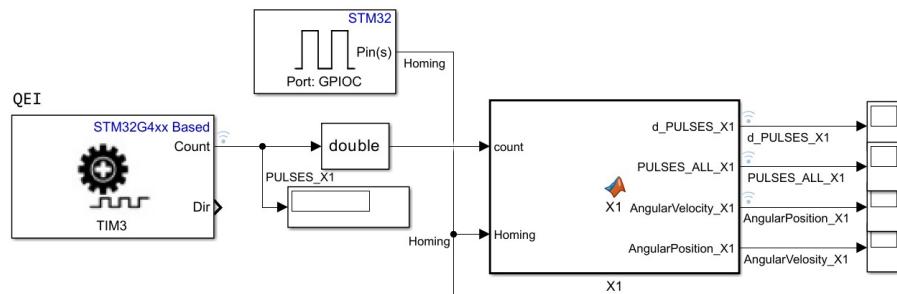
รูปที่ 47 การตั้งค่า Tim ใน CubeMX (ต่อ)

จากนั้นสุดท้ายไปที่ project manager advanced setting และเข้าไปตั้งค่าพิมพ์ให้กลายเป็น LL

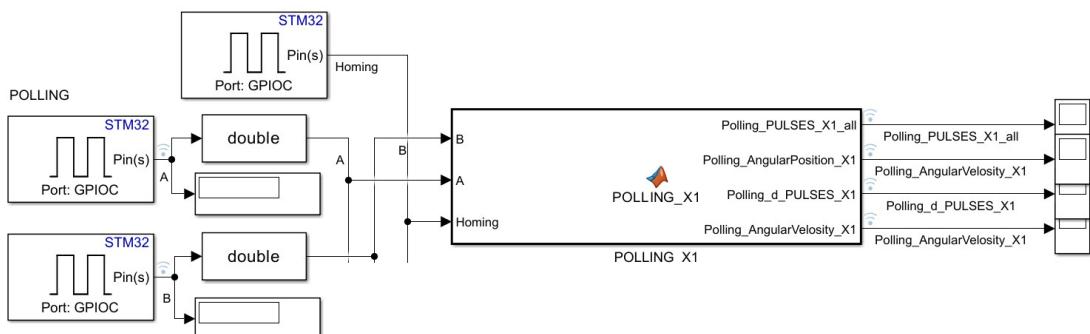
2.6.6 ทำการเขียน Simulink ที่ใช้สำหรับอ่านค่า



รูปที่ 48 Simulink รวมของ Encoder



รูปที่ 49 Simulink ของ การอ่านค่า Encoder แบบ QEI



รูปที่ 50 Simulink ของ การอ่านค่า Encoder แบบ polling

ใช้บล็อกฟังก์ชัน encoder ในการอ่านค่า count จาก encoder ทั้ง x1 x2 x4 ซึ่งจะสามารถอ่านค่าได้จากค่าที่เราตั้งไปในข้อก่อนหน้า ซึ่งจะส่งข่าวออกมากทั้งหมด 4 ค่าซึ่งก็คือ d_PULSES, PULSES_ALL_X2, AngularVelocity และ AngularPosition

2.6.7 ทำการเขียน Code อ่านค่าแบบ QEI โดยเริ่มจาก สร้าง Simulink เพื่อรับค่า Count ที่อ่านได้เข้ามา จากนั้นนำมาผ่าน code ทำการ Unwrap และแสดงผลออกมาเป็น pulse , Position และ Discrete Velocity ดังภาพ

```

function [d_PULSES_X1, PULSES_ALL_X1,AngularVelocity_X1,AngularPosition_X1] = X1(count,Homing)
    PPR = 24 ;
    DTime = 0.01 ;
    COUNT_MAX = 65536 - mod(65536,PPR);

    persistent PULSES_pass
    persistent PULSES_ALL

    if isempty(PULSES_pass)
        PULSES_pass = 0;
    end

    if isempty(PULSES_ALL)
        PULSES_ALL = 0;
    end

    d_PULSES_X1 = count - PULSES_pass;

    if d_PULSES_X1 > COUNT_MAX / 2
        d_PULSES_X1 = d_PULSES_X1 - COUNT_MAX;
    elseif d_PULSES_X1 < -COUNT_MAX / 2
        d_PULSES_X1 = d_PULSES_X1 + COUNT_MAX;
    end

    PULSES_ALL = PULSES_ALL + d_PULSES_X1;
    PULSES_ALL_X1 = PULSES_ALL;

    AngularPosition_X1 = (PULSES_ALL_X1*2*180)/PPR ;
    AngularVelocity_X1 = (((d_PULSES_X1)*2*180)/PPR)/DTime ;

    PULSES_pass = count;
    if Homing == 1
        d_PULSES_X1 = 0 ;
        PULSES_ALL_X1 = 0;
        PULSES_ALL = 0;
        PULSES_pass = count;
    end
end

```

รูปที่ 51 code การอ่านค่า Encoder แบบ QEI เพื่อทำการ "Unwrap" สัญญาณ

การเขียนโค้ดนี้มีจุดประสงค์หลักเพื่อจัดการกับการ "Unwrap" สัญญาณ encoder ที่มีการนับจำนวน pulses เกินค่าความละเอียดที่กำหนด โดยใช้ฟังก์ชัน X1 เพื่อกำหนณตำแหน่งเชิงมุม (Angular Position) และความเร็วเชิงมุม (Angular Velocity) จากข้อมูล pulses ที่ได้จาก encoder พร้อมจัดการกับกรณีที่จำนวน pulses เกินขอบเขตการนับสูงสุด (COUNT_MAX) ของ encoder

ในขั้นต้น โค้ดจะกำหนดค่าคงที่บางตัวที่ใช้ในการคำนวณ เช่น PPR (Pulses Per Revolution) ซึ่งแสดงจำนวน pulses ที่เกิดขึ้นต่อการหมุน 1 รอบของ encoder ค่า DTime กำหนดช่วงเวลาที่ใช้ในการคำนวณความเร็วเชิงมุม และ COUNT_MAX กำหนดขีดจำกัดของจำนวน pulses ที่สามารถอ่านได้โดยพิจารณาจากความละเอียดของ encoder ซึ่งในที่นี้มีค่าสูงสุดที่ 65536

จากนั้น โค้ดจะใช้ตัวแปร persistent คือ PULSES_pass และ PULSES_ALL เพื่อบันทึกค่า pulses ที่ผ่านไปในรอบก่อนหน้า (PULSES_pass) และค่า pulses สะสมทั้งหมด (PULSES_ALL) ซึ่งตัวแปรเหล่านี้จะไม่ถูกรีเซ็ตเมื่อออกจากฟังก์ชัน จึงสามารถใช้เก็บข้อมูลในรอบถัดไปได้ ในกรณีที่ยังไม่มีการกำหนดค่า โค้ดจะตั้งค่าเริ่มต้นให้เป็นศูนย์

หลังจากนั้น จะคำนวณความแตกต่างระหว่างค่า pulses ปัจจุบัน (count) และค่า pulses ก่อนหน้า (PULSES_pass) ซึ่งให้ผลลัพธ์เป็นจำนวน pulses ที่เพิ่มขึ้นหรือลดลง (d_PULSES_X1) หากค่า d_PULSES_X1 เกินขอบเขตที่กำหนด (คือ COUNT_MAX / 2) โค้ดจะปรับค่า d_PULSES_X1 โดยการ

เพิ่มหรือลบ COUNT_MAX เพื่อให้ค่า pulses ที่ได้ไม่ว่ากลับไปยังศูนย์แต่มีการนับต่อเนื่องเสมอ การ "Unwrap" ค่า encoder ทำให้สามารถคำนวณค่าตำแหน่งเชิงมุมได้อย่างต่อเนื่อง

ในขั้นตอนถัดไป โค้ดจะนำ d_PULSES_X1 ที่ได้มาบวกเพิ่มใน PULSES_ALL ซึ่งเป็นการบันทึกจำนวน pulses สะสมทั้งหมดใน PULSES_ALL_X1 ต่อจากนั้น ค่าที่ได้จะถูกใช้ในการคำนวณตำแหน่งเชิงมุม (AngularPosition_X1) และความเร็วเชิงมุม (AngularVelocity_X1) โดยตำแหน่งเชิงมุมคำนวณจากค่า pulses ที่สะสมและแบ่งเป็นองศา ส่วนความเร็วเชิงมุมคำนวณจากค่า pulses ที่เปลี่ยนแปลงในรอบนั้นๆ หารด้วยช่วงเวลาที่กำหนด (DTIME)

สุดท้าย โค้ดจะอัปเดตค่า PULSES_pass ให้เป็นค่าของ count ปัจจุบันเพื่อเก็บเป็นข้อมูลสำหรับรอบถัดไป และหาก Homing มีค่าเป็น 1 โค้ดจะรีเซ็ตค่า d_PULSES_X1, PULSES_ALL_X1, และ PULSES_ALL กลับไปที่ศูนย์ และบันทึกค่า count ใน PULSES_pass เพื่อเริ่มต้นการอ่านใหม่

2.6.8 ทำการเขียน Code อ่านค่าแบบ polling โดยเริ่มจาก สร้าง Simulink เพื่อรับค่า สัญญาณ pulse ที่จาก A และ B มาเทียบค่ากัน จากนั้นนำมาผ่าน code ทำการ อ่านค่าสัญญาณทั้ง 2 จากนั้นแสดงผลออกมา และแสดงผลออกมาเป็น pulse , Position และ Discrete Velocity ดังภาพ

```
function [Polling_PULSES_X1_all,Polling_AngularPosition_X1,Polling_d_PULSES_X1,Polling_AngularVelocity_X1] = POLLING_X1(A, B,Homing)
persistent pass_A pass_B PULSES_all_initialized Polling_PULSES_X1 Polling_PULSES_X1_d_all
% set zero first time
if isempty(PULSES_all_initialized)
    % เช็คเม็ดเงิน persistent
    pass_A = 0 ;
    pass_B = 0;
    Polling_PULSES_X1 = 0;
    Polling_PULSES_X1_d_all = 0 ;
    PULSES_all_initialized =true;
end
% ตรวจสอบค่าที่ร่อง A
if (pass_A == 0) && (A == 1)
    if (B == 0 && pass_B == 0)
        Polling_PULSES_X1 = Polling_PULSES_X1 + 1;
    elseif (B == 1 && pass_B == 1)
        Polling_PULSES_X1 = Polling_PULSES_X1 - 1;
    end
end
% Update pass A B
pass_A = A;
pass_B = B;

% ล้างออกค่าที่ร่อง Encoder
Polling_d_PULSES_X1 = Polling_PULSES_X1 - Polling_PULSES_X1_d_all;
Polling_PULSES_X1_d_all = Polling_PULSES_X1;
Polling_AngularPosition_X1 = (Polling_PULSES_X1_d_all*2*180)/24 ;
Polling_AngularVelocity_X1 = ((Polling_d_PULSES_X1*2*180)/24)/0.01 ;
Polling_PULSES_X1_all = Polling_PULSES_X1_d_all;

if Homing == 1]
    pass_A = 0 ;
    pass_B = 0;
    Polling_PULSES_X1 = 0;
    Polling_PULSES_X1_d_all = 0 ;
    Polling_d_PULSES_X1 = 0 ;
    Polling_PULSES_X1_all = 0 ;
    Polling_AngularPosition_X1 = 0 ;
    Polling_AngularVelocity_X1 = 0 ;
end
end
```

รูปที่ 52 code การอ่านค่า Encoder แบบ Polling

พึงշันจะตรวจสอบว่าตัวแปร PULSES_all_initialized ถูกกำหนดค่าแล้วหรือไม่ หากยังไม่ถูกกำหนด จะตั้งค่าเริ่มต้นให้กับตัวแปรต่าง ๆ เช่น pass_A, pass_B, Polling_PULSES_X1 และ Polling_PULSES_X1_d_all เป็นศูนย์

หลังจากนั้นฟังก์ชันจะตรวจจับการเปลี่ยนแปลงของสัญญาณจากช่อง A ของ encoder เมื่อสัญญาณจากช่อง A เปลี่ยนจาก 0 ไปเป็น 1 จะตรวจสอบสถานะของช่อง B เพื่อพิจารณาทิศทางการหมุน หากช่อง B อยู่ในสถานะ 0 ฟังก์ชันจะเพิ่มค่า Polling_PULSES_X1 แต่ถ้าช่อง B อยู่ในสถานะ 1 จะทำการลดค่า Polling_PULSES_X1 แทน กระบวนการนี้ช่วยให้ฟังก์ชันสามารถตรวจจับทิศทางการหมุนได้จากการเปลี่ยนแปลงของสัญญาณในช่อง A และ B

จากนั้น ฟังก์ชันจะคำนวณค่าความแตกต่างของพัลส์ (Polling_d_PULSES_X1), ตำแหน่งเชิงมุม (Polling_AngularPosition_X1), และความเร็วเชิงมุม (Polling_AngularVelocity_X1) โดยนำค่าพัลส์ที่อ่านได้มาคำนวณเป็นองศาและความเร็วเชิงมุมต่อเวลา ซึ่งการคำนวนนี้จะอัปเดตทุกครั้งที่มีการเปลี่ยนแปลงของพัลส์

สุดท้าย ฟังก์ชันมีการตรวจสอบค่า Homing หาก Homing มีค่าเป็น 1 หมายความว่ามีการสั่งให้รีเซ็ตค่าทั้งหมด ฟังก์ชันจะทำการรีเซ็ตค่าตัวแปรที่เกี่ยวข้องทั้งหมดกลับไปเป็นศูนย์ เพื่อให้สามารถเริ่มต้นการอ่านค่าใหม่ได้จากจุดเริ่มต้น

2.6.9 จากนั้นทำการเก็บค่า โดยหมุน Encoder ได้เรื่อย ๆ และเทียบสัญญาณแบบต่าง ๆ

2.7 วัสดุอุปกรณ์

2.7.1 Incremental Encoder AMT103-V	จำนวน 1 ตัว
2.7.2 BOURNS PEC11R-4220F-N0024	จำนวน 1 อัน
2.7.3 Nucleo STM32G474RE	จำนวน 1 ตัว
2.7.4 สายอัปโหลด	จำนวน 1 ชุด
2.7.5 EncoderXplorer	จำนวน 1 ชุด
2.7.6 Jumper Wire Cable	

2.8 ตัวแปร

ตัวแปรต้น : การหมุน Incremental Encoder

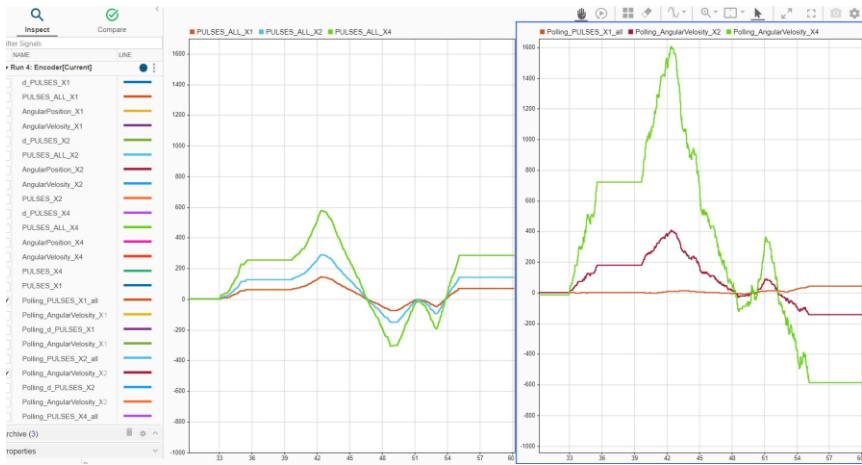
ตัวแปรตาม : Raw Signal, Angular Position, Angular Velocity

ตัวแปรควบคุม : ยี่ห้อและรุ่นของ Encoder ที่ใช้

2.9 ผลการทดลอง

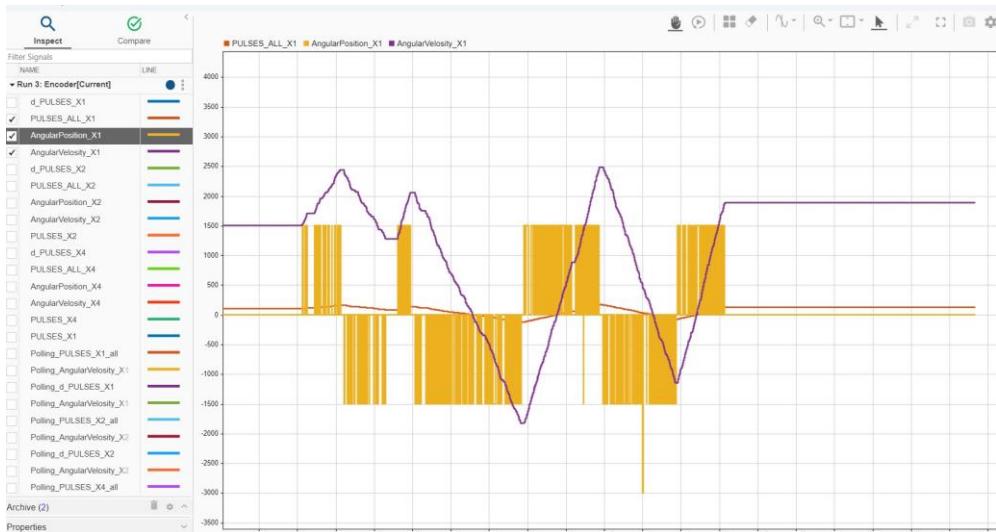
2.9.1 ทำการดูค่าที่อ่านได้จาก QEI และ polling method

ทำการ plot ค่าของ x1 x2 x4 ที่อ่านจาก QEI และแบบ polling method เพื่อมาเปรียบเทียบค่ากัน ซึ่งจากการทดลองจะเห็นว่าได้ผลลูกอกมาดังภาพด้านล่างจะเห็นว่าการอ่านค่าทั้งสองทำให้ค่าเผาที่อ่านได้มีความแตกต่างกันอย่างมาก

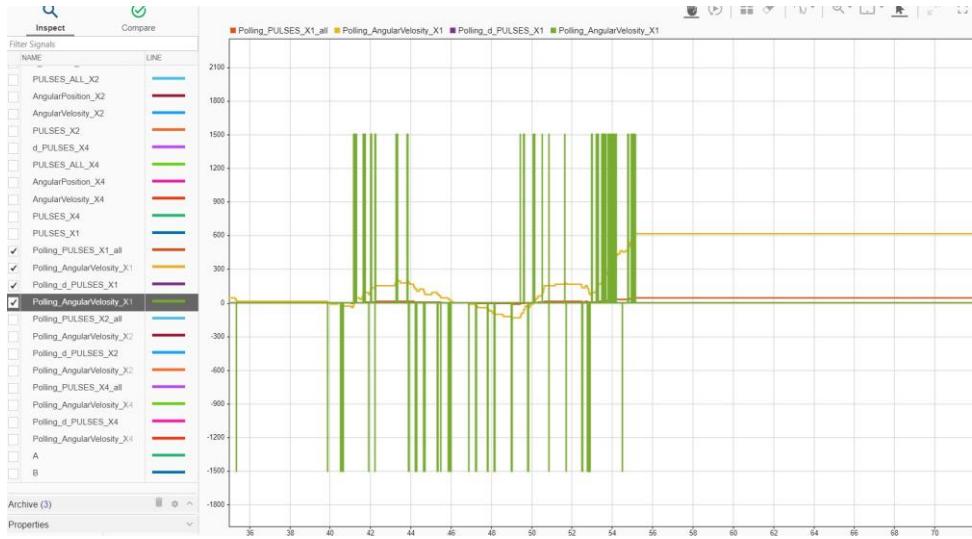


รูปที่ 53 ค่าที่อ่านได้จาก QEI และ polling method

2.9.2 ทำการเทียบข้อมูลจากการอ่านของ QEI เทียบ x1 ในการหาค่า Angular position และ Angular velocity และ และความต่างของ pulse กับ pulse ก่อนหน้า

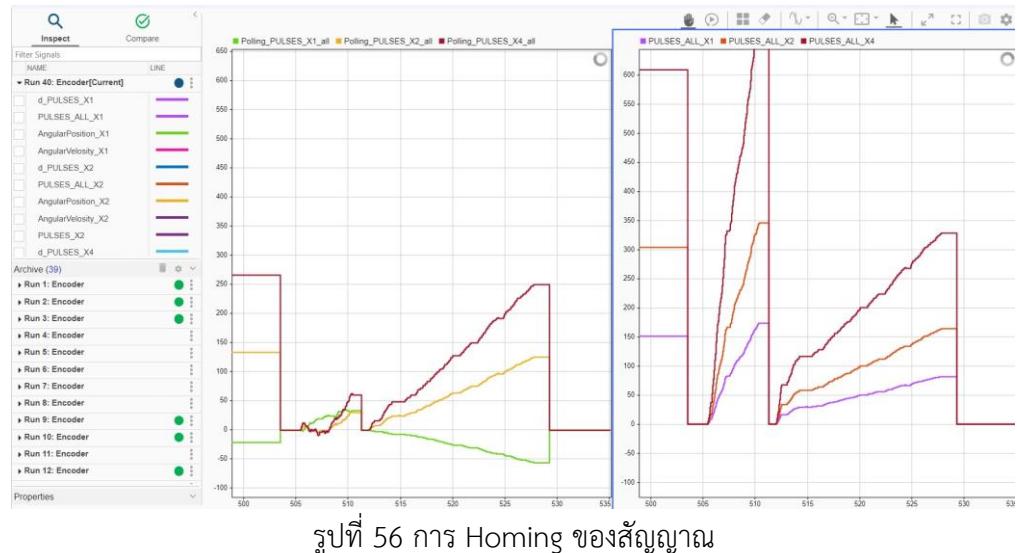


รูปที่ 54 การอ่านของ QEI เทียบ x1 ในการหาค่า Angular position และ Angular velocity และ และความต่างของ pulse กับ pulse ก่อนหน้า



รูปที่ 55 การอ่านของ polling เทียบ x1 ในการหาค่า Angular position และ Angular valuity และ และความต่างของ pulse กับ pulse ก่อนหน้า

2.9.3 การ Homing



รูปที่ 56 การ Homing ของสัญญาณ

จากการจะเห็นว่าเมื่อกดบุ่ม Homing ค่า ทำให้ค่าที่สัญญาณอุปกรณารีบันบ 0 ที่ position ใหม่

2.10 สรุปผล

2.10.1 ทำการดูค่าที่อ่านได้จาก QEI และ polling method

เมื่อทำการ plot ค่าของโหมดการอ่านแบบ x1, x2, และ x4 จาก QEI และ polling method พบร่วมค่าที่อ่านได้จากสองวิธีนี้มีความแตกต่างกันอย่างชัดเจน โดยการอ่านจาก QEI มักจะให้ค่าที่มีความเสถียรและแม่นยำมากกว่าเนื่องจาก QEI สามารถจับทิศทางและจำนวนพัลส์ได้อย่างแม่นยำ

ขณะที่การใช้ polling method อาจเกิดความล่าช้าและมีข้อจำกัดในเรื่องของความละเอียด ซึ่งอาจทำให้เกิดข้อผิดพลาดในการคำนวณค่า

2.10.2 การเปรียบเทียบค่าที่อ่านได้จาก QEI และ Polling Method

เมื่อทำการ plot ค่าของโหมดการอ่านแบบ x_1 , x_2 , และ x_4 จาก QEI และ polling method พบว่าค่าที่อ่านได้จากสองวิธีนี้มีความแตกต่างกันอย่างชัดเจน โดยการอ่านจาก QEI มักจะให้ค่าที่มีความเสถียรและแม่นยำมากกว่าเนื่องจาก QEI สามารถจับทิศทางและจำนวนพัลส์ได้อย่างแม่นยำ ขณะที่การใช้ polling method อาจเกิดความล่าช้าและมีข้อจำกัดในเรื่องของความละเอียด ซึ่งอาจทำให้เกิดข้อผิดพลาดในการคำนวณค่า

2.10.3 การเปรียบเทียบข้อมูลจากการอ่านของ QEI และ Polling Method สำหรับโหมดการอ่านแบบ x_1

ในการทดลองนี้ได้ทำการคำนวณค่า Angular Position และ Angular Velocity โดยเปรียบเทียบข้อมูลจากการอ่านแบบ QEI กับแบบ polling ที่โหมด x_1 โดยพบว่าการอ่านค่าของ QEI ให้ค่า Angular Position และ Angular Velocity ที่แม่นยำกว่าเนื่องจาก QEI สามารถตรวจจับการเปลี่ยนแปลงของพัลส์อย่างต่อเนื่อง ขณะที่การอ่านด้วย polling method มีโอกาสพลาดพัลส์บางช่วง เนื่องจากเป็นการอ่านค่าตามจังหวะเวลา (time-based) ซึ่งอาจไม่ครอบคลุมการเปลี่ยนแปลงของพัลส์ที่เกิดขึ้นในช่วงสั้น ๆ ส่งผลให้ค่า Angular Position และ Angular Velocity มีความไม่สม่ำเสมอ

2.10.4 การ Homing ของสัญญาณ

การทดลองนี้แสดงการทำ homing เพื่อรีเซ็ตตำแหน่งของ encoder โดยใช้ QEI พบร่วมการทำ homing ทำให้ค่าพัลส์สะสม (PULSES_ALL) และตำแหน่งเชิงมุมถูกรีเซ็ตกลับสู่ศูนย์ ซึ่งมีประโยชน์ในการตั้งจุดอ้างอิงใหม่หลังจากการหมุนไปหลายรอบ ทั้งนี้ QEI สามารถทำ homing ได้อย่างแม่นยำกว่า polling เนื่องจากสามารถตรวจจับการเปลี่ยนแปลงของพัลส์ได้อย่างต่อเนื่องและตอบสนองต่อการรีเซ็ตตำแหน่งได้ดีกว่า

2.11 อภิรายผล

2.11.1 ค่าที่อ่านได้จาก QEI และ polling method

สัญญาณที่อ่านได้จากการอ่านค่า encoder ทั้ง 2 วิธี มีความแตกต่างกันมากเนื่องจากความแตกต่างของสรุปความต่างของ QEI และ Polling:

QEI : เก็บข้อมูลได้ต่อเนื่องและแม่นยำ เพราะใช้ Hardware ในการตรวจจับโดยตรง ลด noise ได้ดี หมายความว่าการรับการวัดที่ต้องการรายละเอียดสูง

Polling : เก็บข้อมูลเป็นช่วง ๆ ทำให้ข้อมูลไม่ต่อเนื่อง มีโอกาสเกิดความคลาดเคลื่อนและ noise มากกว่า เนื่องจากต้องรับการเก็บข้อมูลที่ไม่ต้องการความแม่นยำสูง

จากราฟ : QEI ให้กราฟที่ลื่นไหลและละเอียด ส่วน Polling ให้กราฟแบบขั้นบันได

2.11.2 ทำการเทียบข้อมูลจากการอ่านของ QEI เทียบ x1 ในการหาค่า Angular position และ Angular velocity และความต่างของ pulse กับ pulse ก่อนหน้า

Angular Position (ตำแหน่งเชิงมุม): กราฟเส้นสีม่วงแสดงค่าของตำแหน่งเชิงมุมที่มีการเปลี่ยนแปลงอย่างต่อเนื่องตามการหมุนของ encoder โดยสามารถสังเกตเห็นได้ว่าค่า Angular Position ค่อย ๆ เพิ่มขึ้นและลดลงในช่วงที่มีการหมุน ซึ่งแสดงถึงความต่อเนื่องในการอ่านตำแหน่ง

Angular Velocity (ความเร็วเชิงมุม): กราฟเส้นสีเหลืองแสดงค่าความเร็วเชิงมุมที่เปลี่ยนแปลงตามการเคลื่อนที่ของ encoder โดยการเปลี่ยนแปลงอย่างรวดเร็วในค่าความเร็วแสดงถึงการเร่งหรือชลอการหมุน ซึ่งค่าเหล่านี้มีการเปลี่ยนแปลงที่ค่อนข้างต่อเนื่อง หรือมีการกระโดดในการอ่านหรือไม่

การเปลี่ยนแปลงของ Pulse กับ Pulse ก่อนหน้า: เส้นสีส้มแสดงการเปรียบเทียบของ Pulse ที่เกิดขึ้นระหว่างตำแหน่งต่าง ๆ ของการหมุน โดยจะเห็นว่ามีการเปลี่ยนแปลงไปตามการหมุน ซึ่งการวิเคราะห์นี้ช่วยให้เราเข้าใจว่าการอ่าน Pulse มีการเปลี่ยนแปลงอย่างต่อเนื่อง หรือมีการกระโดดในการอ่านหรือไม่

จากการวิเคราะห์จะเห็นได้ว่า QEI 在การอ่านค่า X1 ให้ค่าที่ต่อเนื่องและแม่นยำสำหรับการหาค่าตำแหน่งเชิงมุมและความเร็วเชิงมุม ทั้งนี้ข้อมูลการเปลี่ยนแปลงของ Pulse ช่วยให้เห็นภาพชัดเจน ถึงการเคลื่อนที่ของ encoder

2.11.3 การ Homing

จากการทดลองแสดงให้เห็นว่าโหมดการอ่านพัลส์ x1, x2, และ x4 ส่งผลต่ออัตราการสะสมพัลส์ทั้งหมด (PULSES_ALL) โดยโหมด x4 มีความละเอียดสูงสุด ทำให้สะสมพัลส์ได้มากที่สุดในการหมุนเดียว กัน การทำ homing ช่วยรีเซ็ตค่า PULSES_ALL กลับสู่ศูนย์ ซึ่งเป็นประโยชน์ในการตั้งค่าจุดอ้างอิงใหม่หลังการทำงานของ encoder ผลลัพธ์ยืนยันว่า homing สามารถรีเซ็ตตำแหน่งได้อย่างมีประสิทธิภาพ และค่า PULSES_ALL เริ่มนับใหม่ตามโหมดการอ่าน

2.12 ข้อเสนอแนะ

ตรวจสอบสภาพของสายไฟก่อนใช้ต่อทุกครั้ง

2.13 เอกสารอ้างอิง

- 2.13.1 [What do X1, X2, and X4 position encoding mean for incremental encoders?](#)
- 2.13.2 [STM32G4-WDG_TIMERS-General_Purpose_Timer_GPTIM](#)
- 2.13.3 [BOURNS PEC11R-4220F-N0024 Datasheet](#)
- 2.13.4 [Incremental Encoder AMT103-V Datasheet](#)

3. Magnetic Sensor

3.1 จุดประสงค์

- 3.1.1 สามารถอธิบายการทำงานของ Ratiometric Linear Hall Effect Magnetic Sensor ได้
- 3.1.2 สามารถอธิบายลักษณะสัญญาณ Output ของ Ratiometric Linear Hall Effect Magnetic Sensor ได้ เมื่อ Magnetic Flux Density เป็น Raw Signal, Magnetic Flux Density ในหน่วย SI derived
- 3.1.3 สามารถอธิบายความสัมพันธ์ระหว่าง Magnetic Field Shielding และ Magnetic Flux Density
- 3.1.4 สามารถแสดงให้เห็นว่าสัญญาณ Output แปรผันตามสัญญาณ Input แบบ Real Time โดยมี Output เป็น Raw Signal, Magnetic Flux Density ในหน่วย SI derived

3.2 สมมติฐาน

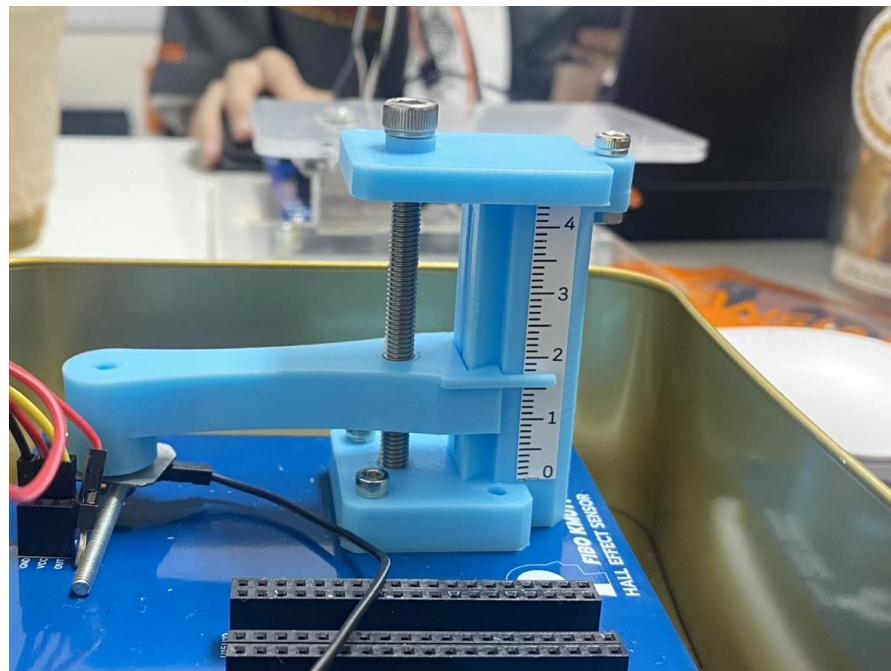
ข้าแม่เหล็กทั้งเหนือและใต้แต่ละขั้วนั้นส่งผลต่อแรงดันไฟฟ้า และแรงดันไฟฟ้าที่ได้มีอุปกรณ์มีค่าที่แตกต่างกัน เมื่อใส่ Sheild และไม่ใส่ Sheild

3.3 นิยามศัพท์เฉพาะ

- Magnetic Field Shielding หรือ การป้องกันสนามแม่เหล็ก คือกระบวนการป้องกันและควบคุมทิศทางของ สนามแม่เหล็กไม่ให้เข้าสู่อุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ เนื่องจากอาจได้รับผลกระทบจากสนามแม่เหล็กได้
- Magnetic Flux Density หรือ ความหนาแน่นฟลักซ์แม่เหล็ก คือค่าที่บอกถึงความหนาแน่นของ สนามแม่เหล็กในพื้นที่นั้น ๆ มีหน่วยเป็น เทสลา (Tesla – T)
- Magnetic Flux Direction คือทิศทางของฟลักซ์แม่เหล็กที่ออกจากขั้วเหนือ (North Pole) ไปยังขั้วใต้ (South Pole) ของสนามแม่เหล็ก
- Sensitivity Linearity คือความสามารถในการตอบสนองต่อการเปลี่ยนแปลงของสนามแม่เหล็กอย่าง สม่ำเสมอและแม่นยำของเซ็นเซอร์ โดยเมื่อสนามแม่เหล็กที่วัดมีการเปลี่ยนแปลงอย่างคงที่

3.4 นิยามเชิงปฏิบัติการ

- การวัดตามสเกล คือ การวัดตามสเกลที่ติดมากับบอร์ดของ Magnetic Sensor โดยมีระยะ 0 ถึง 5 เซนติเมตร หรือก็คือ 0 ถึง 50 มิลลิเมตร



รูปที่ X สเกลการวัดที่ติดมากับบอร์ด

3.5 เอกสารและงานวิจัยที่เกี่ยวข้อง

3.5.1 Datasheet ของ Magnetic Sensor

https://www.ti.com/lit/ds/symlink/drv5055.pdf?ts=1728890274202&ref_url=https%253A%252F%252Fwww.google.com%252F

DRV5055 Ratiometric Linear Hall Effect Sensor

1 Features

- Ratiometric Linear Hall Effect Magnetic Sensor
- Operates From 3.3-V and 5-V Power Supplies
- Analog Output With $V_{CC}/2$ Quiescent Offset
- Magnetic Sensitivity Options (At $V_{CC} = 5$ V):
 - A1/Z1: 100 mV/mT, ± 21 -mT Range
 - A2/Z2: 50 mV/mT, ± 42 -mT Range
 - A3/Z3: 25 mV/mT, ± 85 -mT Range
 - A4/Z4: 12.5 mV/mT, ± 169 -mT Range
- Fast 20-KHz Sensing Bandwidth
- Low-Noise Output With ± 1 -mA Drive
- Compensation For Magnet Temperature Drift for A1/A2/A3/A4 Versions and None for Z1/Z2/Z3/Z4 Versions
- Standard Industry Packages:
 - Surface-Mount SOT-23
 - Through-Hole TO-92

2 Applications

- Precise Position Sensing
- Industrial Automation and Robotics
- Home Appliances
- Gamepads, Pedals, Keyboards, Triggers
- Height Leveling, Tilt and Weight Measurement
- Fluid Flow Rate Measurement
- Medical Devices
- Absolute Angle Encoding
- Current Sensing

3 Description

The DRV5055 is a linear Hall effect sensor that responds proportionally to magnetic flux density. The device can be used for accurate position sensing in a wide range of applications.

The device operates from 3.3-V or 5-V power supplies. When no magnetic field is present, the analog output drives half of V_{CC} . The output changes linearly with the applied magnetic flux density, and four sensitivity options enable maximal output voltage swing based on the required sensing range. North and south magnetic poles produce unique voltages.

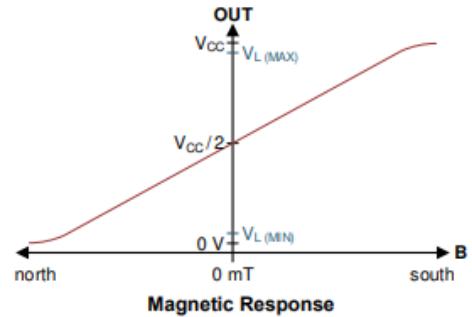
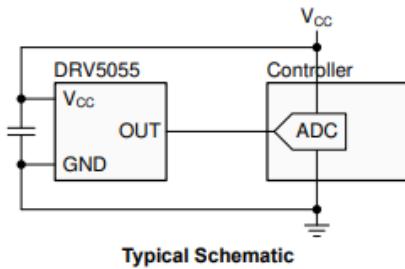
Magnetic flux perpendicular to the top of the package is sensed, and the two package options provide different sensing directions.

The device uses a ratiometric architecture that can eliminate error from V_{CC} tolerance when the external analog-to-digital converter (ADC) uses the same V_{CC} for its reference. Additionally, the device features magnet temperature compensation to counteract how magnets drift for linear performance across a wide –40°C to 125°C temperature range. Device options for no temperature compensation of magnet drift are also available.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
DRV5055	SOT-23 (3)	2.92 mm × 1.30 mm
	TO-92 (3)	4.00 mm × 3.15 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.



⚠ An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Magnetic flux that travels from the bottom to the top of the package is considered positive in this document. This condition exists when a south magnetic pole is near the top (marked-side) of the package. Magnetic flux that travels from the top to the bottom of the package results in negative millitesla values.

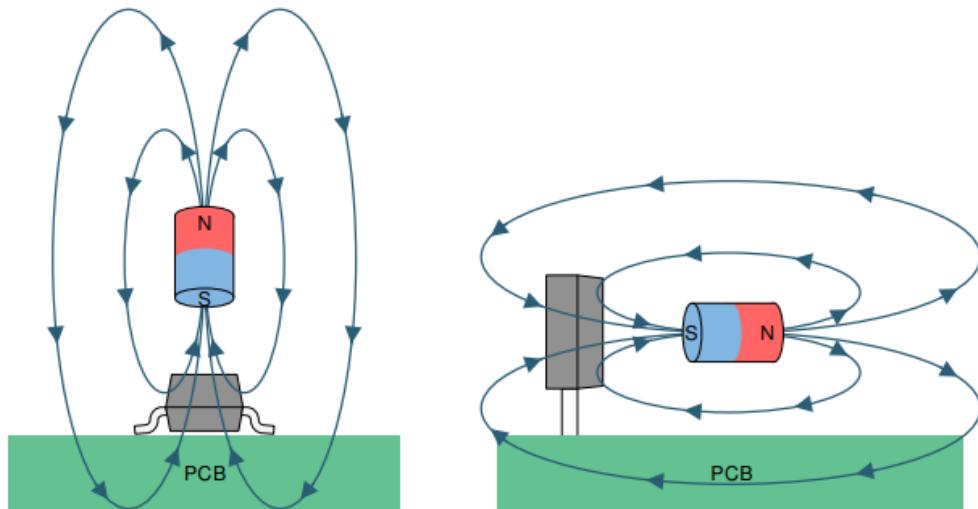


Figure 7-2. The Flux Direction for Positive B

7.3.2 Magnetic Response

When the DRV5055 is powered, the DRV5055 outputs an analog voltage according to [Equation 1](#):

$$V_{OUT} = V_Q + B \times (\text{Sensitivity}_{(25^\circ\text{C})} \times (1 + S_{TC} \times (T_A - 25^\circ\text{C}))) \quad (1)$$

where

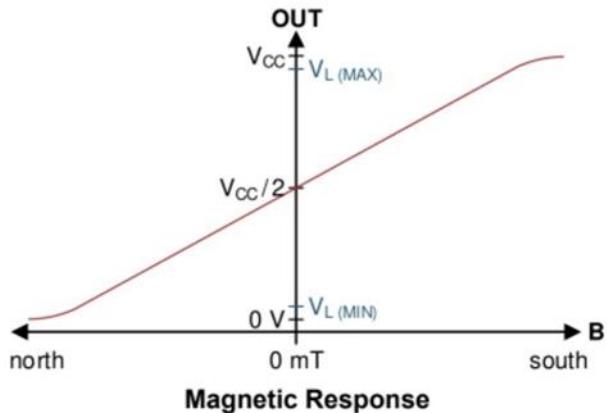
- V_Q is typically half of V_{CC}
- B is the applied magnetic flux density
- $\text{Sensitivity}_{(25^\circ\text{C})}$ depends on the device option and V_{CC}
- S_{TC} is typically $0.12\text{%/}^\circ\text{C}$ for device options DRV5055A1 - DRV5055A4 and is $0\text{%/}^\circ\text{C}$ for DRV5055Z1 - DRV5055Z4 options
- T_A is the ambient temperature
- V_{OUT} is within the V_L range

As an example, consider the DRV5055A3 with $V_{CC} = 3.3$ V, a temperature of 50°C , and 67 mT applied. Excluding tolerances, $V_{OUT} = 1650$ mV + 67 mT \times (15 mV/mT \times ($1 + 0.0012/\text{}^\circ\text{C} \times (50^\circ\text{C} - 25^\circ\text{C}))$) = 2685 mV.

รูปที่ X Datasheet ของ Magnetic Sensor

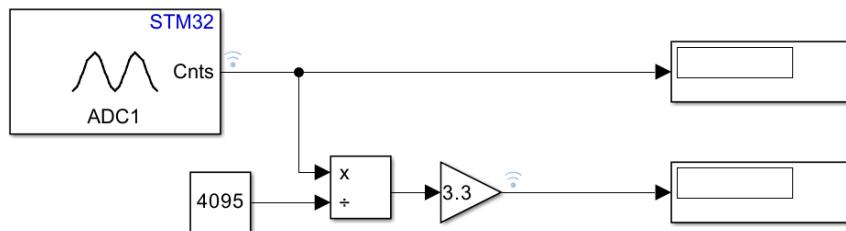
3.6 วิธีดำเนินการทดลอง

3.6.1 จากการศึกษา Datasheet DRV5055 พบร้าฟความสัมพันธ์ระหว่างสนามแม่เหล็ก และ Voltage มีความสัมพันธ์แบบ Linear โดยที่มี แกน Y คือ Voltage (Volt) แกน X สนามแม่เหล็ก (mT) ตามที่แสดงในภาพด้านล่าง



รูปที่ X กราฟความสัมพันธ์ระหว่างความหนาแน่นฟลักซ์แม่เหล็ก และ Voltage

3.6.2 ทางคณะผู้จัดทำจึงต้องการทำการทดลองกับ Magnetic Sensor ที่ได้จากบอร์ดเพื่อพิสูจน์ว่าสามารถอ่านค่าความสัมพันธ์ระหว่าง Voltage (Volt) และระยะห่างของแม่เหล็กได้ตาม Datasheet ซึ่งจากบอร์ดนั้นมีสเกลการวัดความสูงทั้งหมดอยู่ที่ 50 มิลลิเมตร ทางคณะผู้จัดทำจึงทำการเก็บค่าทั้งหมด 7 จุด จาก 0 ไปจนถึง 35 มิลลิเมตร โดยเพิ่มทีละ 5 มิลลิเมตรผ่าน Simulink ทำการเก็บค่า จุดละ 14 วินาที เพราะต้องการใช้ค่าที่ช่วง 2 -10 วินาที เพื่อป้องกันค่าคลาดเคลื่อน เนื่องจากค่าที่อ่านได้ในช่วงแรกอาจจะเป็นค่า yang ไม่เสถียร



รูปการออกแบบการเก็บค่าใน Simulink

จากการจะเห็นได้ว่า เมื่อรับค่าแบบ Analog เข้ามา ค่าเหล่านี้จะถูกหารด้วย 4095 ซึ่งเป็นค่าที่ได้จากการเทียบบัญชีโดยร่างค่าระหว่างจำนวนบิตกับค่าแรงดันไฟฟ้า จากนั้นจึงนำผลลัพธ์ไปคูณกับ 3.3 ซึ่งเป็นค่าแรงดันขาเข้า เพื่อแปลงค่าที่ได้จาก Analog ให้เป็นแรงดันไฟฟ้า

3.6.3 นำข้อมูลที่ได้จากข้อที่ 2 มาวัดเป็นกราฟ ด้วย MATLAB ซึ่งกราฟที่ได้ออกมาเป็นกราฟระหว่าง Voltage (Volt) และระยะทาง (cm)

```

% ค่าหนตื่อไฟล์ทั้งหมดที่ต้องการโหลด
files = {'SS100.mat', 'SS150.mat', 'SS200.mat', 'SS250.mat', 'SS300.mat', 'SS350.mat', 'SS400.mat'};
x = 1.00 : 0.5 : 4.00 ;
% สร้างตัวแปรเก็บค่าเฉลี่ย
mean_values = zeros(1, length(files));

% ค่าหนตื่อเวลาที่ต้องการให้ใช้
start_time = 2; % เวลาที่เริ่มต้น
end_time = 12; % เวลาสิ้นสุด

% สร้างตัวแปรเก็บค่าเฉลี่ย
mean_values = zeros(1, length(files));

for i = 1:length(files)
    % โหลดข้อมูลจากไฟล์
    data = load(files{i});

    % ตัวชี้อีฟิล์สจาก struct และเข้าถึงข้อมูล
    field_name = fieldnames(data);
    timeseries_data = data.(field_name{1});

    % เลือกช่วงเวลาที่ต้องการ
    time = timeseries_data.Time;
    values = timeseries_data.Data;

    % กรองข้อมูลตามช่วงเวลา
    selected_values = values(time >= start_time & time <= end_time);

    % คำนวณค่าเฉลี่ยของข้อมูลในช่วงเวลาที่เลือก
    mean_values(i) = mean(selected_values);
end

% ผลลัพธ์
figure;
plot(mean_values,x);
xlabel('Distance(cm.)');
ylabel('V');
title('HALL SENSOR (South)');

```

รูปที่ X ตัวอย่าง code การวาดกราฟในโปรแกรม MATLAB

3.6.4 นำข้อมูลจากข้อที่ 2 มาคิดคำนวนให้กล้ายเป็นค่าของสนามแม่เหล็ก สามารถหาได้โดยนำเอาค่าเฉลี่ยในวินาทีที่ 2-12 ซึ่งเป็นข้อมูล 1,000 ตัว มาวิเคราะห์เป็นกราฟจากสูตรตามภาพด้านล่าง

7.3.2 Magnetic Response

When the DRV5055 is powered, the DRV5055 outputs an analog voltage according to [Equation 1](#):

$$V_{\text{OUT}} = V_Q + B \times (\text{Sensitivity}_{(25^{\circ}\text{C})} \times (1 + S_{\text{TC}} \times (T_A - 25^{\circ}\text{C}))) \quad (1)$$

where

- V_Q is typically half of V_{CC}
- B is the applied magnetic flux density
- $\text{Sensitivity}_{(25^{\circ}\text{C})}$ depends on the device option and V_{CC}
- S_{TC} is typically 0.12%/°C for device options DRV5055A1 - DRV5055A4 and is 0%/°C for DRV5055Z1 - DRV5055Z4 options
- T_A is the ambient temperature
- V_{OUT} is within the V_L range

As an example, consider the DRV5055A3 with $V_{\text{CC}} = 3.3$ V, a temperature of 50°C , and 67 mT applied. Excluding tolerances, $V_{\text{OUT}} = 1650 \text{ mV} + 67 \text{ mT} \times (15 \text{ mV/mT} \times (1 + 0.0012/\text{°C} \times (50^{\circ}\text{C} - 25^{\circ}\text{C}))) = 2685 \text{ mV}$.

รูปที่ X สูตรหา V_{out} ของ Magnetic Response

```

% กำหนดชื่อไฟล์ทั้งหมดที่ต้องการโหลด (มảngเป็นกลุ่มตามชื่อ)
groups = {
    {'N100.mat', 'N150.mat', 'N200.mat', 'N250.mat', 'N300.mat', 'N350.mat', 'N400.mat'}, % Group_N
    {'NS100.mat', 'NS150.mat', 'NS200.mat', 'NS250.mat', 'NS300.mat', 'NS350.mat', 'NS400.mat'}, % Group_NS
    {'S100.mat', 'S150.mat', 'S200.mat', 'S250.mat', 'S300.mat', 'S350.mat', 'S400.mat'}, % Group_S
    {'SS100.mat', 'SS150.mat', 'SS200.mat', 'SS250.mat', 'SS300.mat', 'SS350.mat', 'SS400.mat'} % Group_SS
};

% กำหนดแกน x ตามที่ต้องการ
x = [-3.00,-2.50,-2.00,-1.50,-1.00,-0.50,0.00, 3.00,2.50,2.00,1.50,1.00,0.5,0.00];

start_time = 2; % เวลาที่เริ่มต้น
end_time = 12; % เวลาสิ้นสุด

% สร้างตัวแปรเก็บข้อมูลเฉลี่ยของแต่ละกลุ่ม
mean_values_N_NS = zeros(2, 7); % สำหรับกลุ่ม N และ NS มี 7 จุด
mean_values_S_SS = zeros(2, 7); % สำหรับกลุ่ม S และ SS มี 7 จุด

% ลูปผ่านแต่ละกลุ่ม
for g = 1:2
    files = groups{g};
    for i = 1:length(files)
        % โหลดข้อมูลจากแฟ้มไฟล์
        data = load(files{i});

        % ดึงชื่อไฟล์ต่างๆ
        field_name = fieldnames(data);
        timeseries_data = data.(field_name{1});

        % เลือกช่วงเวลาที่ต้องการ
        time = timeseries_data.Time;
        values = timeseries_data.Data;

        % กรองข้อมูลตามช่วงเวลา
        selected_values = values(time >= start_time & time <= end_time);

        % คำนวณค่าเฉลี่ยของข้อมูลในช่วงเวลาที่เลือก
        mean_values_N_NS(g, i) = mean(selected_values);
    end
end

for g = 3:4
    files = groups{g};
    for i = 1:length(files)
        % โหลดข้อมูลจากแฟ้มไฟล์
        data = load(files{i});

        % ดึงชื่อไฟล์ต่างๆ
        field_name = fieldnames(data);
        timeseries_data = data.(field_name{1});

        % เลือกช่วงเวลาที่ต้องการ
        time = timeseries_data.Time;
        values = timeseries_data.Data;

        % กรองข้อมูลตามช่วงเวลา
        selected_values = values(time >= start_time & time <= end_time);

        % คำนวณค่าเฉลี่ยของข้อมูลในช่วงเวลาที่เลือก
        mean_values_S_SS(g-2, i) = mean(selected_values);
    end
end

% ผลลัพธ์ที่ได้
figure;
plot(x(1:7), mean_values_N_NS(1, :), '-o', 'DisplayName', 'Group N'); % ครึ่งแรกของ Group N
hold on;
plot(x(1:7), mean_values_N_NS(2, :), '-s', 'DisplayName', 'Group NS'); % ครึ่งแรกของ Group NS
hold on;
plot(x(8:14), mean_values_S_SS(1, :), '^', 'DisplayName', 'Group S'); % ครึ่งหลังของ Group S
hold on;
plot(x(8:14), mean_values_S_SS(2, :), '-d', 'DisplayName', 'Group SS'); % ครึ่งหลังของ Group SS
hold off;

% ผลลัพธ์ที่ได้
figure;
plot(mean_values,x);
xlabel('Distance(cm.)');
ylabel('V');
title('HALL SENSOR (South)');

```

รูปที่ X ตัวอย่าง Code ที่ใช้ในการหาค่าของสนามแม่เหล็ก

3.6.5 นำกราฟที่ได้จากการทดลองมาเปรียบเทียบกับกราฟที่ปรากฏใน Datasheet เพื่อช่วยให้เห็นภาพรวมของข้อมูล ซึ่งสามารถบ่งชี้ถึงประสิทธิภาพและข้อแตกต่างของอุปกรณ์หรือระบบที่กำลังศึกษาอยู่ทั้งหมด

3.6.6 ทำการวิเคราะห์และสรุปผลการทดลอง เพื่อให้เข้าใจถึงประสิทธิภาพและข้อจำกัดของอุปกรณ์ที่ศึกษาอยู่

3.7 วัสดุอุปกรณ์

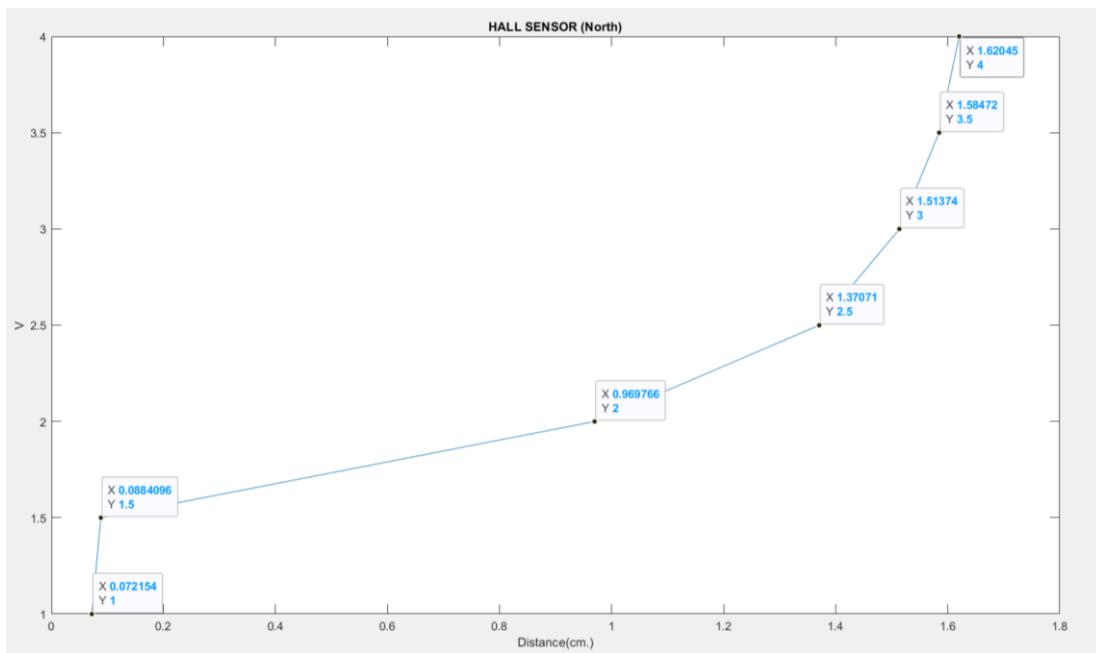
3.7.1	DRV5055 Ratiometric Linear Hall Effect Magnetic Sensor	จำนวน 1 ตัว
3.7.2	Nucleo STM32G474RE	จำนวน 1 ตัว
3.7.3	สายอับໂໂລດ	จำนวน 1 ชุด
3.7.4	EncoderXplorer	จำนวน 1 ชุด
3.7.5	Jumper Wire Cable	

3.8 ตัวแปร

- ตัวแปรต้น : ระยะการรับ และการใส่ shield หรือ ไม่ใส่ shield Magnetic Sensor แต่ละชนิด
ตัวแปรตาม : แรงดันไฟฟ้าที่วัดได้ของ Magnetic Sensor ที่วัดได้ในแต่ละแบบ
ตัวแปรควบคุม : ชนิดของ Magnetic Sensor ,วิธีการทดลอง

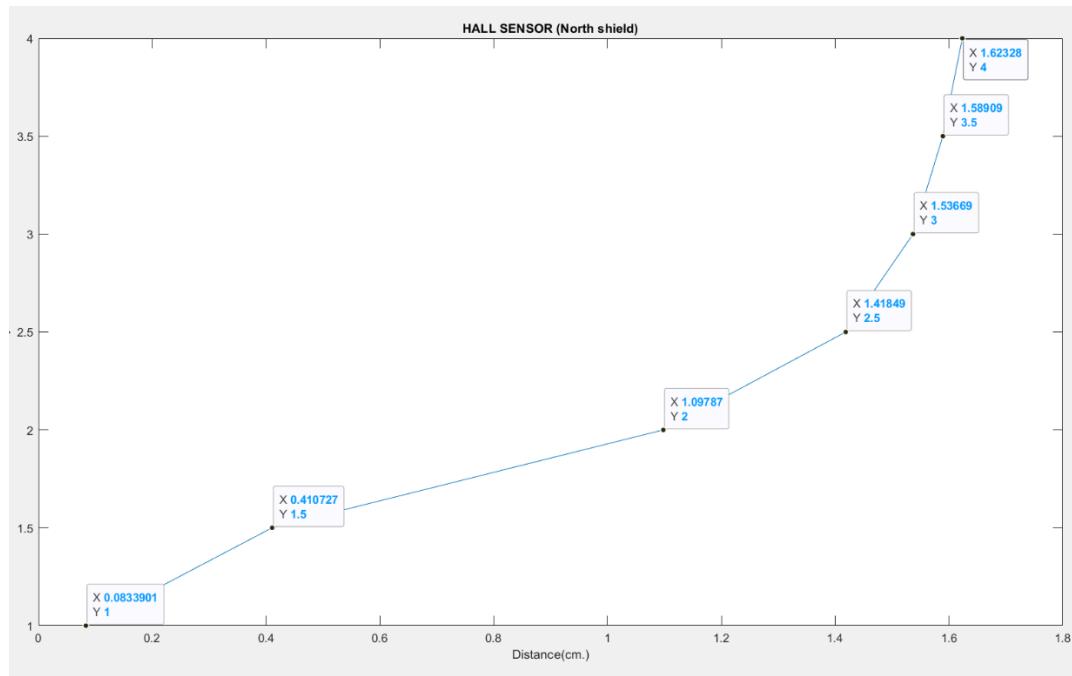
3.9 ผลการทดลอง

จากการนำค่าที่ Magnetic Sensor แต่ละชิ้นวัดได้ทั้งตอนที่ใส่ shield และไม่ได้ใส่ shield ไปวาดเป็นกราฟใน MATLAB พบว่าทั้งตอนที่ใส่ shield และไม่ได้ใส่ shield Magnetic Sensor ขึ้นเนื้อให้กราฟออกมารูปแบบ Exponential ที่ค่อย ๆ เพิ่มขึ้น และ Magnetic Sensor ขึ้นใต้ให้กราฟออกมารูปแบบ Exponential ที่ค่อย ๆ ลดลงดังภาพด้านล่าง กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์ของ Magnetic Sensor ขึ้นเนื้อและใต้ ทั้งตอนที่ใส่ shield และไม่ได้ใส่ shield เป็นกราฟแบบสมการเชิงเส้น (Linear)



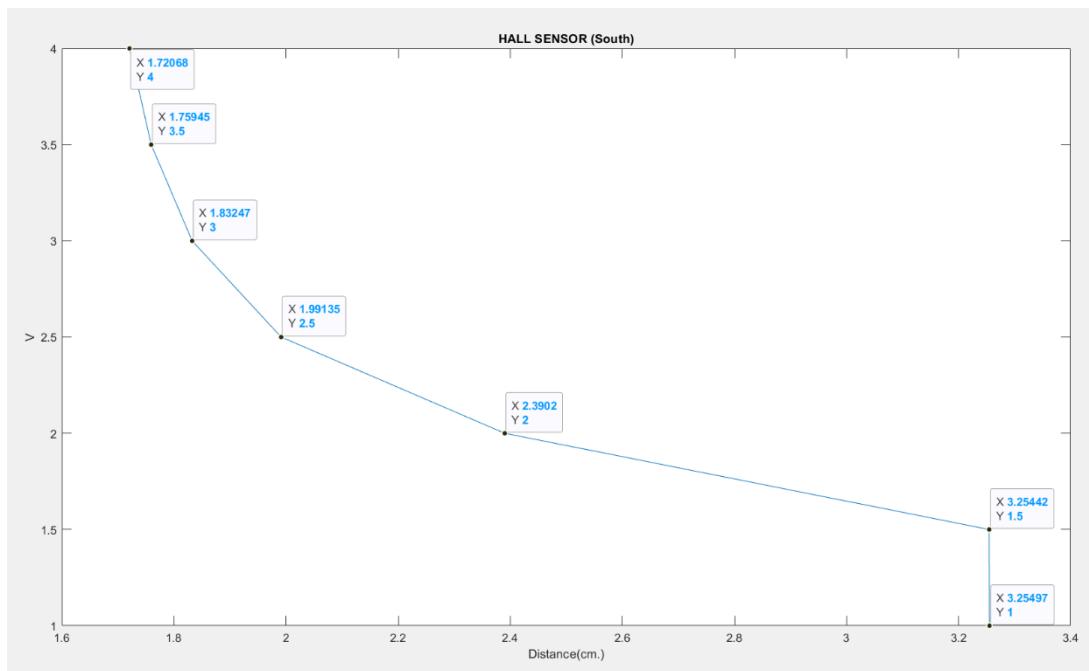
กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์

ของ Magnetic Sensor ขั้วเหนือ ตอนที่ไม่ได้ใส่ shield



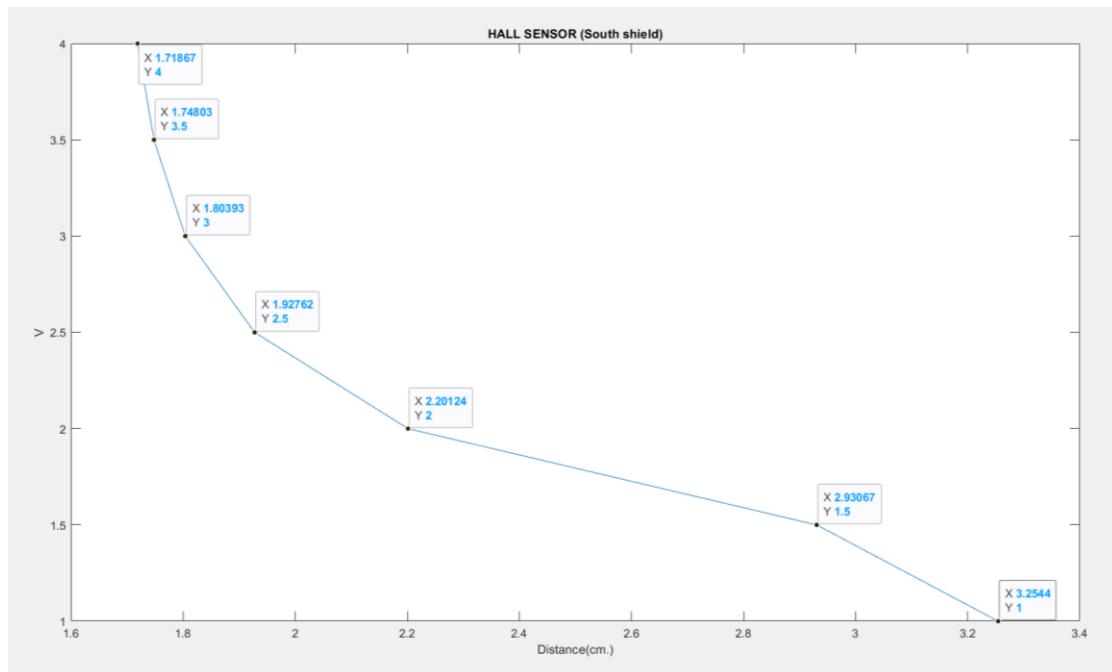
กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์

ของ Magnetic Sensor ขั้วเหนือ ตอนที่ใส่ shield



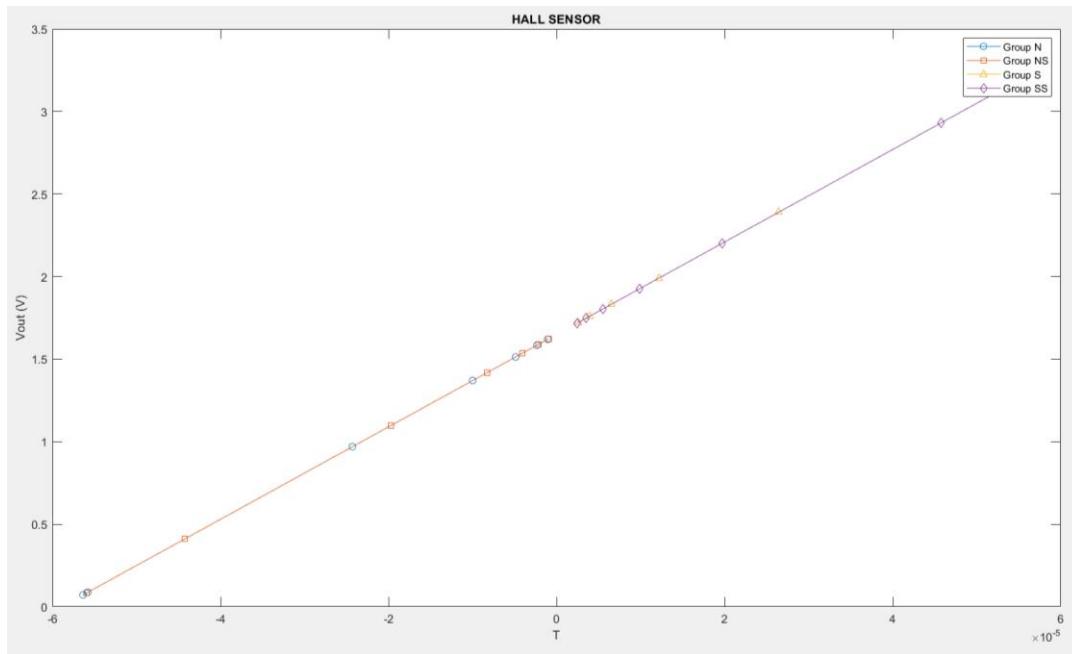
กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์

ของ Magnetic Sensor ข้าวตี้ ตอนที่ไม่ได้ใส่ shield



กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์

ของ Magnetic Sensor ข้าวตี้ ตอนที่ใส่ shield



กราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้า และความหนาแน่นฟลักซ์
ของ Magnetic Sensor ข้างหนึ่งและใต้ทั้งตอนที่ใส่ shield และไม่ได้ใส่ shield

ทางคณะผู้จัดทำได้นำเสนอการคำนวณค่า Magnetic Response จาก Datasheet ของ Ratiometric Linear Hall Effect Magnetic Sensor ดังสมการด้านล่าง

$$V_{out} = V_Q + B \times (Sensitivity_{25^\circ C} \times (1 + S_{TC} \times (T_A + 25^\circ C)))$$

เข้าโปรแกรม Matlab คำนวณหาค่า Ideal ของ Magnetic Flux Density ที่อยู่บนบอร์ด และนำค่าที่ได้ไปหากราฟค่าความคลาดเคลื่อน

```

Vq = 1650; %half Vcc
B = zeros(1, 3301); %magnetic flux density
Sen = 0.03; %sensitivity
Stc = 0.0012; %0.12%
Ta = 25; %ambient temp
Step = zeros(1, 3301);
%Loop calculate magnetic flux density
for i = 0:20:3301
    keep = ((i-Vq)/(Sen*(1+Stc)))*0.001;
    B(i+1) = keep;
    Step(i+1) = i*0.001;
end

Step = Step';
err_B = 1/100*B;
%Plot Graph
hold on;
errorbar(B,Step,err_B,'or','LineWidth',1,'MarkerSize',2,'MarkerEdgeColor','r','MarkerFaceColor','r');
title("Magnetic Response Error","FontSize",16);
grid("on");
xlabel("Magnetic flux density","FontSize",14);
ylabel("Vout","FontSize",14);

```

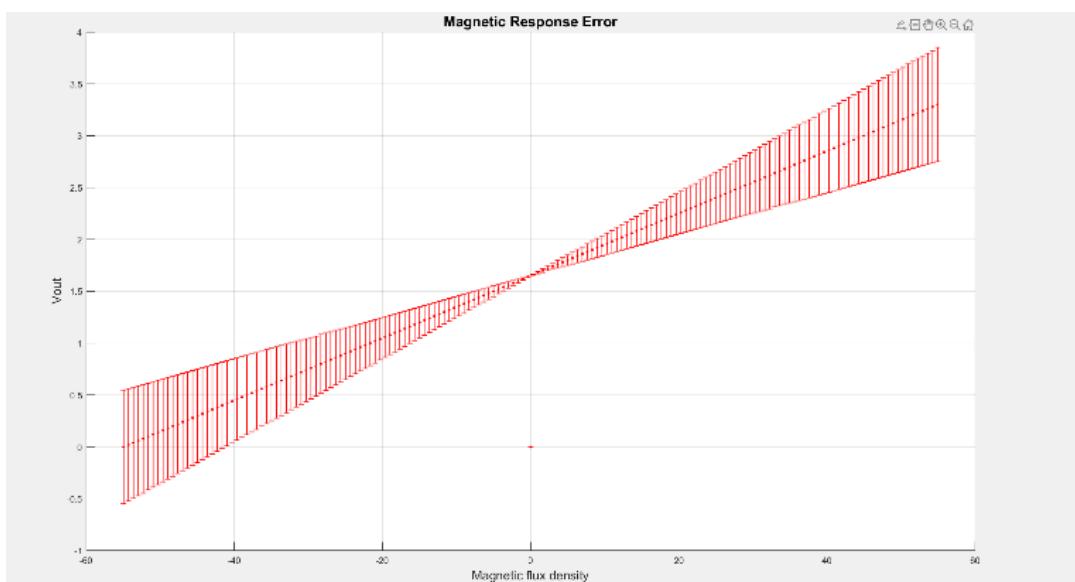
รูปที่ X Code ของการวาดกราฟและคำนวณค่าความคลาดเคลื่อนของ Magnetic Flux Density

3.10 สรุปผล

จากการทดลองพบว่า เมื่อนำกราฟที่แสดงความสัมพันธ์ระหว่างแรงดันไฟฟ้าและความหนาแน่นฟลักซ์แม่เหล็กของ Magnetic Sensor ทั้งในข้อเหนือ และข้อใต้ โดยเปรียบเทียบทั้งในกรณีที่ใส่ และไม่ใส่ shield มาวัดรวมกันพบว่ากราฟมีลักษณะเป็นเส้นตรง (Linear) ซึ่งบ่งชี้ถึงความสัมพันธ์เชิงเส้นระหว่างแรงดันและความหนาแน่นฟลักซ์แม่เหล็ก เมื่อเปรียบเทียบกับกราฟใน Datasheet ของ Magnetic Sensor พบว่ากราฟจากการทดลองและกราฟใน Datasheet มีรูปแบบที่ใกล้เคียงกัน แสดงให้เห็นว่าเซนเซอร์ทำงานได้ตามที่ระบุไว้ใน Datasheet

3.11 อภิปรายผล

จากการทดลองจะพบได้ว่า ค่าที่ได้จากการทดลองนั้น มีทั้งค่าที่อยู่ในเกณฑ์ และบางค่าที่ไม่อยู่ในเกณฑ์ Datasheet แต่ละชนิด ซึ่งสาเหตุที่อาจทำให้เกิดค่าคลาดเคลื่อนได้ก็มีอยู่หลายอย่าง เช่น การขยับระยะห่างของแม่เหล็กไม่ตรงตามสเกลที่อยู่บนบอร์ด หรือสเกลที่อยู่บนบอร์ดเกิดการคลาดเคลื่อนขณะติด หรือแม้แต่การใช้สายจ้มที่มีปัญหาทำให้ค่าที่เก็บเกิดการคลาดเคลื่อนได้เข่นกัน



รูปที่ X กราฟแสดงค่าความคลาดเคลื่อนของ Ratiometric Linear Hall Effect Magnetic Sensor

3.12 ข้อเสนอแนะ

มีการเก็บค่าโดยมีระยะห่างระหว่างค่าที่น้อยลง เพื่อให้ค่าที่ได้มีความละเอียดมากขึ้น กราฟที่ออกมาก็จะสามารถสังเกตความเปลี่ยนไปของกราฟได้ชัดเจนยิ่งขึ้น

3.13 เอกสารอ้างอิง

- 3.13.1 Hall effect sensor Datasheet
- 3.13.2 Magnetic shielding
- 3.13.3 What is Magnetic Flux?
- 3.13.4 Accuracy, Precision, Resolution, and Sensitivity

4. Load cell

4.1 จุดประสงค์

- 4.1.1 อธิบายหลักการทำงาน: เข้าใจการทำงานของ Load Cell, วงจร Wheatstone Bridge, Strain Gauge, และ Differential Amplifier แบบ 2 Op-amps
- 4.1.2 การตอบสนองของ Load Cell วิเคราะห์ Output เมื่อแรงที่กระทำต่อ Load Cell เปลี่ยนแปลง
- 4.1.3 อธิบายความสัมพันธ์ระหว่าง ค่า Gain และ ค่าความต้านทานของ Single External Resistor ส่งผลอย่างไรกับการวัดค่าของ Load Cell
- 4.1.4 Signal Conditioning อธิบายตั้งแต่การอ่านค่า การจัดการข้อมูล และการปรับค่า Calibrate นักศึกษาต้องสามารถอธิบายกระบวนการ Signal Conditioning ทั้งหมด ว่าค่าที่อ่านได้จาก Load Cell มีที่มาอย่างไร อธิบายให้เห็นถึงวิธีคิดและขั้นตอนทั้งหมด ทั้งก่อนและ หลัง Calibrate Sensor หรือ วิธีจัดการข้อมูลที่ได้มา จัดการอย่างไร รวมถึงหาสมการความสัมพันธ์ระหว่างแรงที่กระทำต่อ Load Cell แรงดันไฟฟ้าที่ออกมาจาก Load Cell ทั้งแบบที่ยังไม่ผ่าน INA125 และผ่าน INA125
- 4.1.5 สามารถเขียนโปรแกรม ในการสั่งการร่วมกับบอร์ด Nucleo STM32G474RE โดยใช้สัญญาณจาก INA125 Instrumentation Amplifier ที่ต่อเข้ากับ Single Point Load Cell เป็น Input และ แสดงสัญญาณ Output จากการ Log สัญญาณ และผลเป็นกราฟจาก Data Inspector ใน MATLAB Simulink และแสดงให้เห็นว่าสัญญาณ Output แปรผันตามสัญญาณ Input แบบ Real Time โดยมี Output เป็นหน่วย SI

4.2 สมมติฐาน

- 4.2.1 การทำงานของ Load cell นั้น ตัว Load Cell จะสร้างสัญญาณตามแรงที่กระทำ โดยวงจร Wheatstone Bridge และ Strain Gauge ช่วยแปลงแรงเป็นแรงดันไฟฟ้า ส่วน Differential Amplifier ขยายสัญญาณแรงดันให้พร้อมสำหรับการอ่านค่าสามารถอธิบายความสัมพันธ์ระหว่าง ค่า Gain และ ค่าความต้านทานของ Single External Resistor ได้ และความสัมพันธ์เหล่านี้ส่งผลอย่างไรกับการวัดค่าของ Load Cell
- 4.2.2 การตอบสนองของ Load Cell คือ แรงที่กระทำมากขึ้นควรทำให้แรงดัน Output สูงขึ้นตามสัดส่วนของแรงที่วัดได้
- 4.2.3 การปรับค่า Gain และความต้านทานจะส่งผลโดยตรงต่อการขยายสัญญาณจาก Load Cell
- 4.2.4 การปรับตั้งค่าและกระบวนการ Signal Conditioning มีผลสำคัญในการเพิ่มความแม่นยำของข้อมูลที่ได้จาก Load Cell โดยสัญญาณที่อ่านได้จะต้องผ่านกระบวนการจัดการเพื่อให้ได้ค่าที่ถูกต้อง ซึ่งการ Calibrate ช่วยลดข้อผิดพลาดจากการอ่านข้อมูล โดยที่ค่าสัญญาณหลังจากการปรับและขยายผ่าน INA125 จะมีความสอดคล้องกับแรงที่กระทำต่อ Load Cell และสามารถนำมาหาความสัมพันธ์ที่ชัดเจนระหว่างแรงที่กระทำกับแรงดัน Output ที่ได้

4.2.5 สามารถใช้ MATLAB และ Simulink ร่วมกับบอร์ด Nucleo STM32G474RE รับสัญญาณจาก Load Cell เมื่อเชื่อมต่อกับ INA125 และบอร์ด Nucleo STM32G474RE จะสามารถนำมาแสดงผลเป็นหน้าหนักในหน่วย SI แบบ Real-Time

4.3 นิยามคัพท์เฉพาะ

- 4.3.1 Load cell คือ อุปกรณ์ที่ใช้ในการเปลี่ยนจากแรงหรือน้ำหนักที่กระทำต่อตัวโหลดเซลล์ เป็นสัญญาณทางไฟฟ้า
- 4.3.2 Wheatstone bridge เป็นวงจรที่ใช้สำหรับการหาค่าความต้านทานที่ไม่ทราบค่าโดยใช้หลักการเปรียบเทียบค่าความต้านทานที่ต้องการทราบค่ากับความต้านทานมาตรฐานที่ทราบค่าแล้ว
- 4.3.3 Strain Gauge คือ เครื่องมือที่ใช้ในการตรวจวัดแรงตึงเครียด (Strain) ของวัสดุ ในรูปแบบของการยืดหรือการบีบตัวของวัสดุเมื่อมีแรงกระทำต่อมัน โดยความต้านทานของเซ็นเซอร์จะเปลี่ยนแปลงตามอัตราส่วนของแรงตึงเครียดที่เกิดขึ้น
- 4.3.4 Op-amps Differential Amplifier เป็นวงจรที่ใช้เพื่อยายความแตกต่างระหว่างสัญญาณสองตัวที่เข้ามา โดยมีการลดสัญญาณรบกวนที่อาจจะมาจากสัญญาณทั่วไป
- 4.3.5 Gain คือ ค่าคงตัวที่ทำการปรับ ให้สัญญาณจาก Load Cell มีการขยายให้สอดคล้องกับแรงที่รับได้มากขึ้น

4.4 นิยามเชิงปฏิบัติการ

- 4.4.1 Signal Conditioning คือ กระบวนการจากการอ่านสัญญาณจนถึงการปรับและแปลงให้อยู่ในหน่วยที่ต้องการ รวมถึงการ Calibrate เพื่อลดความคลาดเคลื่อน
- 4.4.2 Calibrate คือการ Calibrate ช่วยให้ค่าที่ได้จาก Load Cell สอดคล้องกับแรงจริง โดยลดข้อผิดพลาดที่เกิดจากการวัด

4.5 เอกสารและงานวิจัยที่เกี่ยวข้อง

- 4.5.1 Datasheet ของ Load cell

FEATURES

- LOW QUIESCENT CURRENT: 460 μ A
- PRECISION VOLTAGE REFERENCE: 1.24V, 2.5V, 5V or 10V
- SLEEP MODE
- LOW OFFSET VOLTAGE: 250 μ V max
- LOW OFFSET DRIFT: 2 μ V/ $^{\circ}$ C max
- LOW INPUT BIAS CURRENT: 20nA max
- HIGH CMR: 100dB min
- LOW NOISE: 38nV/ \sqrt{Hz} at f = 1kHz
- INPUT PROTECTION TO $\pm 40V$
- WIDE SUPPLY RANGE
Single Supply: 2.7V to 36V
Dual Supply: $\pm 1.35V$ to $\pm 18V$
- 16-PIN DIP AND SO-16 SOIC PACKAGES

DESCRIPTION

The INA125 is a low power, high accuracy instrumentation amplifier with a precision voltage reference. It provides complete bridge excitation and precision differential-input amplification on a single integrated circuit.

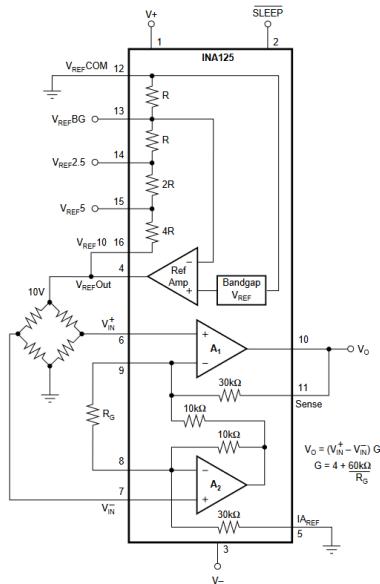
A single external resistor sets any gain from 4 to 10,000. The INA125 is laser-trimmed for low offset voltage (250 μ V), low offset drift (2 μ V/ $^{\circ}$ C), and high common-mode rejection (100dB at G = 100). It operates on single (+2.7V to +36V) or dual ($\pm 1.35V$ to $\pm 18V$) supplies.

The voltage reference is externally adjustable with pin-selectable voltages of 2.5V, 5V, or 10V, allowing use with a variety of transducers. The reference voltage is accurate to $\pm 0.5\%$ (max) with $\pm 35ppm/^{\circ}$ C drift (max). Sleep mode allows shutdown and duty cycle operation to save power.

The INA125 is available in 16-pin plastic DIP and SO-16 surface-mount packages and is specified for the -40° C to $+85^{\circ}$ C industrial temperature range.

APPLICATIONS

- PRESSURE AND TEMPERATURE BRIDGE AMPLIFIERS
- INDUSTRIAL PROCESS CONTROL
- FACTORY AUTOMATION
- MULTI-CHANNEL DATA ACQUISITION
- BATTERY OPERATED SYSTEMS
- GENERAL PURPOSE INSTRUMENTATION



Datasheet Load cell

4.6 วิธีดำเนินการทดลอง

4.6.1 เริ่มจากการคำนวณหาค่า Gain และค่า Single External Resistor

โดยให้ $G = \text{Gain}$

$R_G = \text{Single External Resistor}$

$V_0 = 2.5 V$

จากสูตร

$$V_0 = (V_{in}^+ - V_{in}^-)G$$

แล้ว

$$G = 4 + \frac{60000}{R_G}$$

ซึ่งนำหนักบน Load cell 10 kg และวัดค่าจะได้

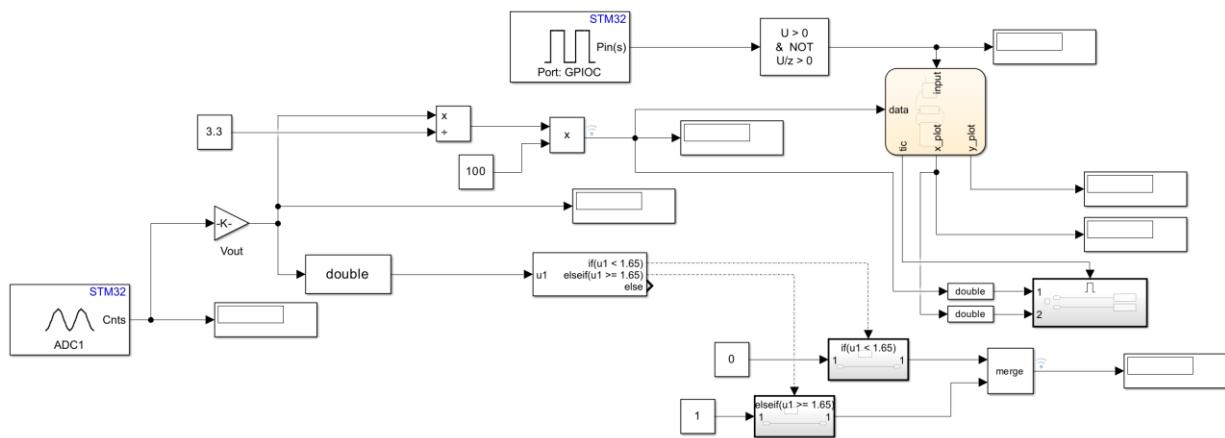
$$(V_{in}^+ - V_{in}^-) = 5 mV$$

แทนค่าลงในสูตรเพื่อหาค่า R_G

$$2.5 = (5 \times 10^{-3})(4 + \frac{60000}{R_G})$$

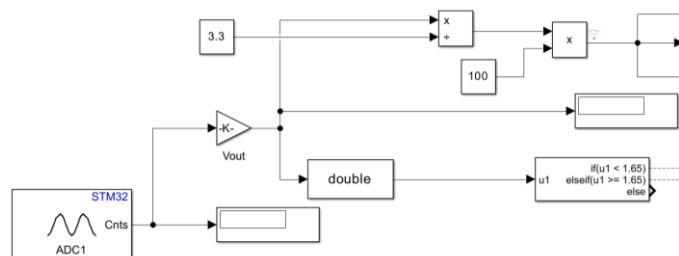
จะได้ค่า $R_G \approx 120 \Omega$

4.6.2 ทำการเขียนโปรแกรมและสร้าง Simulink เพื่อเก็บค่าข้อมูล 500 ค่า ตามที่ได้ออกแบบไว้ ซึ่งจะได้ diagram ออกรมาตามภาพด้านล่าง



รูปที่ X การออกแบบการทดลองใน Simulink

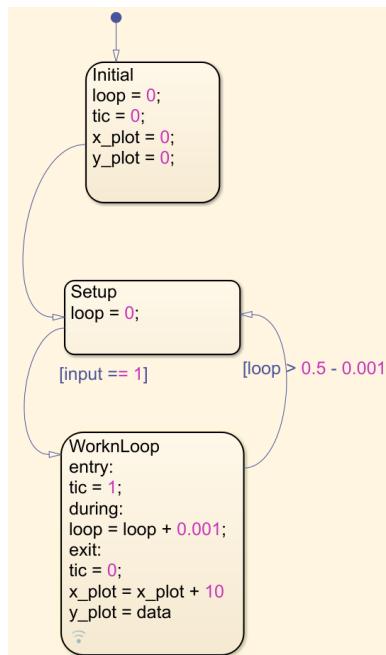
รับค่าสัญญาณจาก Load cell และทำการคำนวณเพื่อหาค่า V_{out} ของ Load cell เพื่อนำค่าที่ได้ส่งเข้า chart



รูปที่ x การออกแบบการทดลองเพื่อนำค่าที่ได้ส่งเข้า chart

จากภาพด้านล่างเป็นแผนผังการทำงานแบบ Stateflow ใน Simulink ซึ่งแสดงถึงการทำงานของระบบโดยมี state หลัก ๆ อยู่ 3 state คือ Initial, Setup และ WorknLoop โดยในแต่ละสถานะมีการทำงานดังนี้

- Initial State เป็นสถานะเริ่มต้น จะทำการกำหนดค่าเริ่มต้นของตัวแปร loop, tic, x_plot, และ y_plot ให้เป็น 0 จากนั้นจะเปลี่ยนไปที่สถานะ Setup
- Setup State กำหนดให้ตัวแปร loop เป็น 0 เพื่อเริ่มการทำงานใหม่ มีการเปลี่ยนแปลงสถานะได้ 2 ทาง ขึ้นอยู่กับเงื่อนไขที่ตั้ง ซึ่งก็คือ
 - ถ้า input == 1 จะเปลี่ยนไปที่ WorknLoop State
 - ถ้า loop > 0.5 - 0.001 จะวนกลับไปที่ Setup State
- WorknLoop State เป็นการทำงานหลักของระบบ โดยมี state ย่อยออกเป็น 3 ส่วน
 - entry เมื่อเข้าสู่สถานะนี้ tic = 1
 - during ขณะที่อยู่ในสถานะนี้ ตัวแปร loop จะถูกเพิ่มขึ้นด้วยค่า 0.001 ทุกรัง
 - exit เมื่อออกจากสถานะนี้ tic = 0 และปรับค่า x_plot โดยเพิ่มขึ้นทีละ 10 และกำหนดค่า y_plot ให้เท่ากับ data



รูปที่ X การออกแบบของ chart ใน Simulink

จากภาพ block In หรือ input ด้านบนจะทำการรับข้อมูลที่ผ่านการคำนวณสัญญาณเป็นค่า Voltage ส่วน block In หรือ input ด้านล่างจะทำการรับค่าน้ำหนักของวัตถุบน Load cell และทำการส่งข้อมูลทั้ง 2 ส่วน ไปที่ workspace ในโปรแกรม MATLAB ผ่าน block To Workspace ใน Simulink



รูปที่ X การออกแบบของ Subsystem ใน Simulink

นำข้อมูลที่ได้มาวัดเป็นกราฟ ด้วย MATLAB ซึ่งกราฟที่ได้ออกมาเป็นกราฟระหว่าง V_{out} (Volt) และน้ำหนัก (kg)

```

Data = out.simout1;
chunkSize = 500; % ขนาดของแต่ละกลุ่ม
nChunks = floor(length(Data) / chunkSize); % จำนวนกลุ่มทั้งหมด
averages = zeros(1, nChunks); % สร้าง array เพื่อเก็บค่าเฉลี่ยแต่ละ step
schmitt = zeros(1,nChunks);
step = linspace(0,10,11);

for i = 1:nChunks
    % หา index เริ่มต้นและสิ้นสุดของแต่ละช่วง
    startIndex = (i - 1) * chunkSize + 1;
    endIndex = i * chunkSize;

    % หาค่าเฉลี่ยของแต่ละช่วงและเก็บไว้ใน averages
    averages(i) = mean(Data(startIndex:endIndex));
end
y = [0,averages];

%Plot Graph
hold on;
plot(step,y);
title("Load Cell Lab","FontSize",16);
grid("on");
xlabel("Mass (kg)","FontSize",14);
ylabel("Vout","FontSize",14);
hold off;

```

รูปที่ X Code การวิเคราะห์ค่า และ Plot Graph

4.6.3 ทำการวิเคราะห์และสรุปผลการทดลอง

4.7 วัสดุอุปกรณ์

4.7.1 Single Point Load Cell YZC-131A	จำนวน 1 ตัว
4.7.2 INA125 Instrumentation Amplifier	จำนวน 1 อัน
4.7.3 Trimpot 100 K 25 Turns	จำนวน 1 อัน
4.7.4 Resistor 4.7K Ohm	จำนวน 1 อัน
4.7.5 สายอัปโหลด	จำนวน 1 ชุด

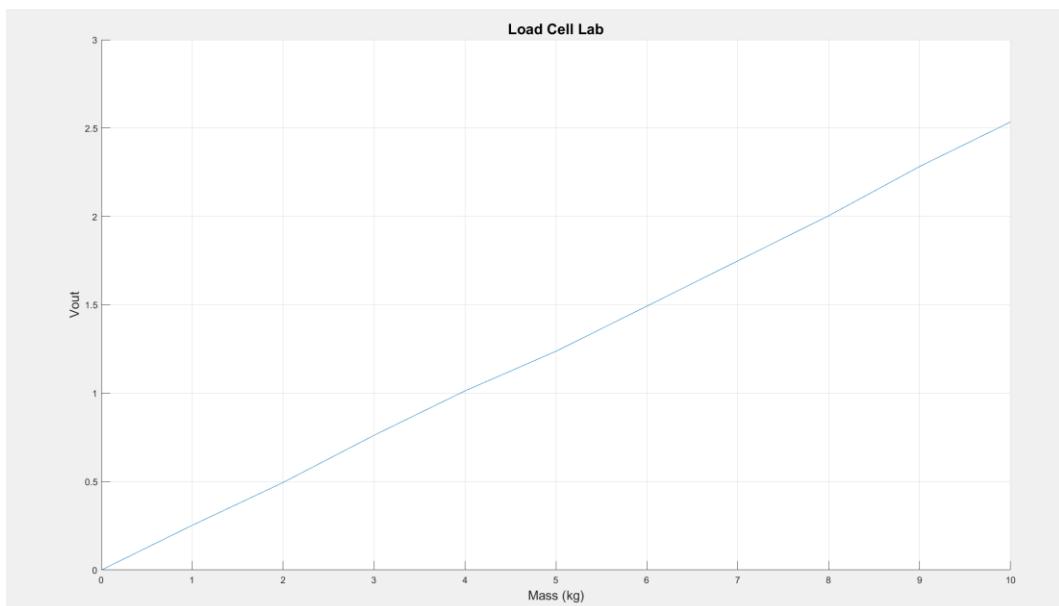
4.7.6	LoadcellXplorer	จำนวน 1 ชุด
4.7.7	Nucleo STM32G474RE	จำนวน 1 ตัว
4.7.8	Jumper Wire Cable	

4.8 ตัวแปร

- ตัวแปรต้น : น้ำหนักที่ซึ่งบน Load cell
 ตัวแปรตาม : ค่าความต่างศักย์ที่เปลี่ยนแปลงไป
 ตัวแปรควบคุม : ชนิดของ Load cell , การวางแผนของน้ำหนัก, ตำแหน่งที่วาง load cell

4.9 ผลการทดลอง

หลังจากที่คณะผู้จัดทำได้เขียนโปรแกรมวัดกราฟในโปรแกรม MATLAB แล้ว โดยกำหนดให้แกน x คือ น้ำหนักที่ซึ่งบน Load cell และแกน y คือ ค่า V_{out} ที่ได้



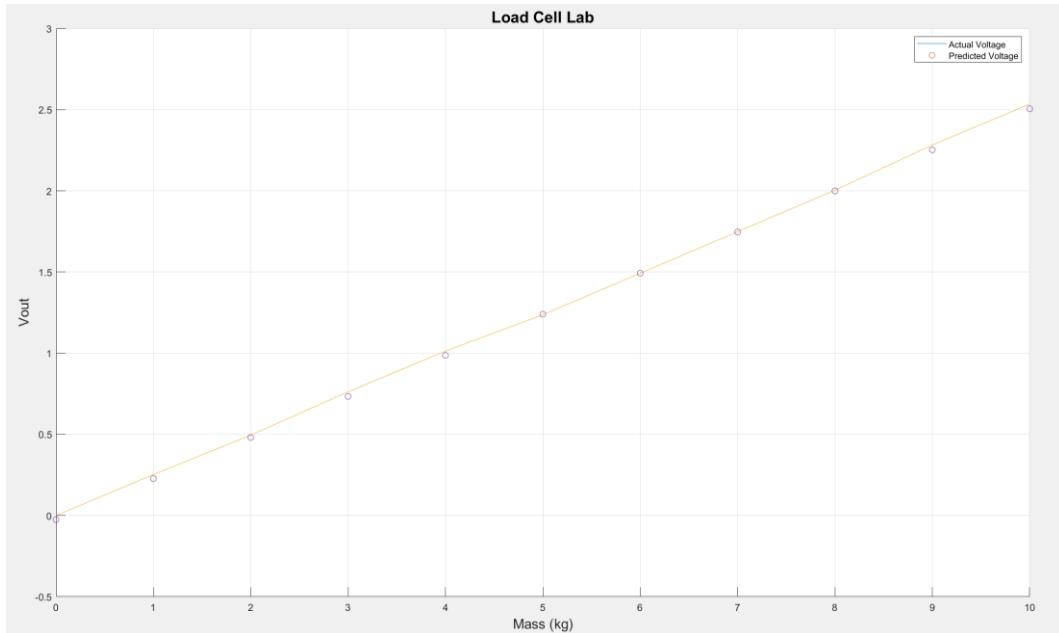
รูปที่ X กราฟของ Single Point Load Cell

4.10 สรุปผล

จากการทดลองพบว่า เมื่อนำกราฟที่แสดงความสัมพันธ์ระหว่างแรงดันไฟฟ้าและน้ำหนักที่ซึ่งอยู่บน Load cell มาวัดพบว่ากราฟมีลักษณะเป็นเส้นตรง (Linear) ซึ่งบ่งชี้ถึงความสัมพันธ์เชิงเส้นระหว่างค่าของแรงดันไฟฟ้าและค่าน้ำหนักที่ซึ่งอยู่บน Load cell และแสดงให้เห็นว่าเซนเซอร์ทำงานได้ตามที่ระบุไว้ใน Datasheet

4.11 อกีปรายผล

จากการทดลองจะพบได้ว่า กราฟที่ได้จากการทดลองนั้นเป็นกราฟเส้นตรง (Linear) โดยมีค่า Predicted Voltage มาจากสมการ $V_{out} = (0.2531 \times Mass) + (-0.02525)$



รูปที่ X กราฟแสดงค่าความคลาดเคลื่อนของ Load cell

```
%Error
err=zeros(1,11);
for j = 0:10
    Ex_force = 0.2531*j-0.02525;
    err(j+1) = Ex_force;
end
plot(step,err,'o');
legend("Actual Voltage", "Predicted Voltage");
```

รูปที่ X Code ของการวัดกราฟและคำนวณค่าความคลาดเคลื่อนของ Load cell

4.12 ข้อเสนอแนะ

ทำการเก็บค่าให้ละเอียดมากขึ้น โดยแบ่งช่วงให้ละเอียดขึ้น จากการเก็บค่าที่ละ 1 กิโลกรัม เป็น 0.5 กิโลกรัม

4.13 เอกสารอ้างอิง

4.13.1 ประเภทของ Load cell

4.13.2 วงศ์บริจ์แบบวีตส์โตน

4.13.3 Strain gauge

4.13.4 The Differential Amplifier

4.13.5 Load cell Datasheet