



## LAB2 : Brushed DC Motor และ Stepper Motor

### Name

- นายณัฐพุทธ์ พุทธิวโรดม 66340500018
- นางสาวนันท์กัส คำสอน 66340500023
- นางสาวบุญวิลัย ปานทอง 66340500030

### Objectives

- สามารถออกแบบการทดลองโดยใช้ทักษะกระบวนการทางวิทยาศาสตร์ ในการ สืบเสาะพัฒนาระบบ ประภากារณ์ ทดลอง บันทึกผลการทดลอง สรุปผล และ อภิปรายผลการทดลอง เข้าใจหลักการทำงานของเซ็นเซอร์ และ อุปกรณ์ทั้งหมดที่ เกี่ยวข้องได้ ตลอดจนใช้หลักการทำงานทางวิทยาศาสตร์ คณิตศาสตร์ สถิติ และศาสตร์อื่นๆ ที่ เกี่ยวข้อง ประยุกต์ใช้ร่วมกับโปรแกรม MATLAB เพื่อเก็บผลการทดลอง วิเคราะห์ผล การทดลอง วิเคราะห์ความ เที่ยงตรง ความแม่นยำ ได้อย่างถูกต้อง และมีเหตุผลรองรับ ตรวจสอบความถูกต้องเทียบกับทฤษฎีที่น่าเชื่อถือ
- อธิบายความสามารถในการรับรู้ปริมาณทางฟิสิกส์ของเซ็นเซอร์ ทั้งหมดตั้งแต่ต้นจนจบกระบวนการได้ เช่น การ อธิบายการวัดกระแสไฟฟ้าของ Current Sensor (จากแรงดันไฟฟ้าแปลงเป็นกระแสไฟฟ้าได้อย่างไร)
- นักศึกษาต้องสามารถกำหนด ตัวแปรในการทดลองได้อย่างถูกต้องและสมเหตุสมผล ไม่ ว่าจะเป็น ตัวแปรต้น ตัว แปรตาม ตัวแปรควบคุม อธิบายจุดประสงค์การทดลอง อธิบายสมมติฐานให้สอดคล้องกับตัวแปรที่กำหนด นิยาม เชิงปฏิบัติการ และมีทฤษฎีที่ น่าเชื่อถือรองรับ เช่น ทฤษฎีทางฟิสิกส์ หรือข้อมูลจาก Datasheet
- นักศึกษาต้องสามารถออกแบบวิธีการทดลองเพื่อหาคำตอบ ให้สอดคล้องกับผลการ เรียนรู้ย่อทั้งหมดได้ด้วย ตนเองอย่างถูกต้องตามหลักวิทยาศาสตร์ รวมทั้ง บันทึกผล สรุปผล อภิปรายผล ตามข้อมูลที่บันทึกได้จริง มี กระบวนการทำซ้ำ อธิบายที่มาของผล การทดลองนั้นได้ โดยใช้อุปกรณ์ เครื่องมือ ไฟล์ Simulink, mlx ฯลฯ และ ชุดการ ทดลองพร้อมบอร์ด Microcontroller ที่ TA จัดเตรียมให้เบื้องต้น
- นักศึกษาต้องสามารถเขียนรายงาน สัญลักษณ์ และสมการทางวิทยาศาสตร์ ได้อย่าง ถูกต้อง ทั้งขนาด และ รูปแบบอักษร การเร้นซ่องไฟ การเร้นขอบกระดาษ การเร้นระยะ พิมพ์ ให้ได้ระยะที่เหมาะสมตามหลักสากล

## Experiment 1: การทดลอง ความสัมพันธ์ระหว่าง Speed, Torque, Current เมื่อมีการปรับ Duty Cycle, Frequency ของ PWM ที่จ่ายเข้า DC Motor เปลี่ยนแปลงไป

### จุดประสงค์

- ศึกษาความสัมพันธ์ระหว่างความเร็ว rob Speed และความถี่ของ PWM ที่ใช้ควบคุณ
- ศึกษาความสัมพันธ์ของค่า Torque และ Current ที่ DC Motor สามารถสร้างได้ในแต่ละความถี่ของ PWM
- สร้างกราฟ Speed vs Frequency และ Torque vs Current
- ศึกษาความสัมพันธ์ระหว่าง Duty Cycle, Current, และ Speed ของ DC Motor

### สมมติฐาน

- Speed vs Frequency: เมื่อความถี่ของ PWM เพิ่มขึ้น ความเร็ว rob (Speed) ของมอเตอร์จะเพิ่มขึ้นจนถึงจุดสูงสุด ซึ่งขึ้นอยู่กับคุณสมบัติของมอเตอร์และแรงดันไฟฟ้าสูงสุดที่มอเตอร์สามารถรองรับได้
- Torque vs Current: Torque ที่มอเตอร์สร้างได้จะเพิ่มขึ้นตามกระแสไฟฟ้า (Current) ที่จ่ายเข้าไป โดยความสัมพันธ์นี้คาดว่าจะเป็นเชิงเส้นตามสมบัติของ DC Motor จนถึงจุดที่กระแสเกินพิกัด (Saturation Point)
- Duty Cycle vs Current และ Duty Cycle vs Speed: Duty Cycle ที่เพิ่มขึ้นจะทำให้กระแสไฟฟ้า (Current) และความเร็ว rob (Speed) เพิ่มขึ้นตามไปด้วย โดยคาดว่าความสัมพันธ์นี้จะไม่เป็นเชิงเส้นเมื่อมอเตอร์เข้าใกล้จุดอิ่มตัว (Saturation) เมื่อ Duty Cycle สูงเกินจุดหนึ่ง กระแสและความเร็วอาจไม่เพิ่มขึ้นอีกเนื่องจากข้อจำกัดของมอเตอร์และแหล่งจ่ายไฟ

### ตัวแปร

ตัวแปรต้น : ความถี่ของ PWM (Frequency), โหลดบน DC Motor

ตัวแปรตาม : ความเร็ว rob ของมอเตอร์ (Speed), Torque ของ DC Motor, กระแสไฟฟ้า (Current)

ตัวแปรควบคุม : แรงดันไฟฟ้า (Voltage), อุณหภูมิและสภาพแวดล้อม

### นิยามศัพท์เฉพาะ

#### 1. DC Motor

มอเตอร์ที่เปลี่ยนพลังงานไฟฟ้ากระแสตรง (Direct Current) เป็นพลังงานกล โดยอาศัยหลักการเหนี่ยวนำแม่เหล็กไฟฟ้า มักถูกใช้งานในระบบที่ต้องการการควบคุมความเร็วและแรงบิดอย่างแม่นยำ

#### 2. Speed

อัตราการหมุนของมอเตอร์ที่แสดงในรูปของรอบต่อวินาที (RPS) หรือรอบต่อนาที (RPM) ซึ่งบ่งบอกว่ามอเตอร์หมุนแกนโดยเร็วเพียงใด โดยมีหน่วยเดียวกันคือ นิวตันเมตร (Nm) หรือ RPM

#### 3. Torque

แรงที่เกิดจากการหมุนรอบแกนกลางของมอเตอร์ ซึ่งเป็นผลคูณระหว่างแรงและระยะทางจากแกนหมุน มีหน่วยวัดเป็นนิวตัน-เมตร ( $N\cdot m$ ) โดยแรงบิดเป็นตัวบ่งชี้ความสามารถของมอเตอร์ในการรับภาระ หรือ เคลื่อนย้ายโหลด

#### 4. Current

ปริมาณการไหลของอิเล็กตรอนผ่านวงจรไฟฟ้าภายในมอเตอร์ มีหน่วยวัดเป็นแอมป์เร (A) โดยกระแสไฟฟ้าที่จ่ายเข้าสู่มอเตอร์จะสัมพันธ์กับแรงบิดและกำลังที่มอเตอร์ต้องการ

#### 5. No Load

สภาพที่มอเตอร์ทำงานโดยไม่มีการต่อเชื่อมกับโหลดหรือภาระใด ๆ ในสถานะนี้มอเตอร์จะทำงานด้วยความเร็วสูงสุด และใช้กระแสไฟฟ้าอยู่ที่สุด เนื่องจากไม่มีแรงต้านหรือโหลดที่ต้องเอาชนะแรงบิด

#### 6. Stall Torque

แรงบิดสูงสุดที่มอเตอร์สามารถสร้างได้ก่อนที่มอเตอร์จะหยุดหมุน โดยเป็นสภาพที่แรงบิดจากภาระหรือโหลดที่กระทำต่อตัวมอเตอร์มีค่ามากจนมอเตอร์ไม่สามารถเอาชนะแรงต้านได้อีกต่อไป

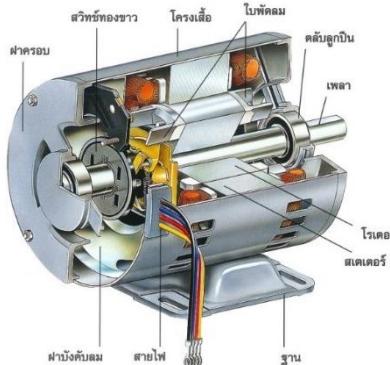
### นิยามเชิงปฏิบัติการ

1. DC Motor คือ Nidec Components Geared DC Geared Motor, 12 V dc, 20 Ncm, 70 rpm, 6mm Shaft Diameter
2. Load Cell คือ Single Point Load Cell YZC-131A ที่ถูกขยายสัญญาณด้วย INA125 Instrumentation Amplifier
3. แม่กัด Load Cell ขึ้นส่วน 3D print ที่ใช้ในการ Torque ซึ่งเป็นขึ้นส่วนที่สามารถนำมาติดตั้งกับเพลาของ DC Motor ด้วย Rigid Flange Coupling และวงปลาຍอีกด้านบน Load Cell เพื่อวัดน้ำหนัก
4. ความถี่ของ PWM คือค่าพารามิเตอร์ที่สามารถเปลี่ยนแปลงได้ในโปรแกรม MotorXploer.slx
5. MotorXploer.slx คือไฟล์โปรแกรม Simulink ที่ใช้ในการควบคุม บอร์ด MotorXploer
6. Log Signal การอ่านสัญญาณต่างๆใน โปรแกรม Simulink

## เอกสารและงานวิจัยที่เกี่ยวข้อง

## 1. Brushed DC Motor

มอเตอร์ที่มีแบร์จานสำหรับการส่งพลังงาน มีส่วนที่สัมผัสกันระหว่าง Rotor และ Stator โดยปกติ Brushed DC Motor ทั่วไปจะใช้ส่วนที่เป็นขดลวดเป็น Rotor จึงต้องจ่ายไฟไปให้ขดลวด เพื่อสร้างสนามแม่เหล็กมาผลักกับ Stator ให้เกิดการหมุน และเมื่อมีการจ่ายไฟ จะจ่ายไฟผ่านแบร์จาน ทำให้เกิดการเสียดสีกันจนสึกหรอ ส่งผลให้มอเตอร์ชนิดนี้มีอายุการใช้งานที่สั้น และมีโอกาสที่จะเกิดประกายไฟ หากใช้งานติดต่อกันนานเกินไป



โดยโครงสร้างจะประกอบไปด้วย 2 ส่วน ได้แก่

1) Stator คือ ส่วนที่อยู่กับที่ จะมีขิดลวดสนามชึ้งทำหน้าที่ในการสร้างสนามแม่เหล็กถาวร แต่บางครั้งมอเตอร์ขนาดเล็ก จะใช้แม่เหล็กถาวรแทนการใช้ขดลวด

2) Rotor คือ ส่วนที่เคลื่อนที่ จะมี Armature Coil และแปรรูปถ่านอยู่ ซึ่งจะจ่ายไฟฟ้ากระแสตรงเข้า Armature Coil ผ่าน Brush และชุด Commutator จะทำให้เกิดแรงบิดในการหมุนของ Rotor ที่เกิดจากการกระทำระหว่างขั้วแม่เหล็กของ Stator และ Rotor ที่ต่างขั้วกันจะผลักกันทำให้เกิดการหมุนขึ้น และทั้งสองขดลวดจะมีแหล่งจ่ายไฟฟ้ากระแสตรงจากแหล่งเดียวกัน

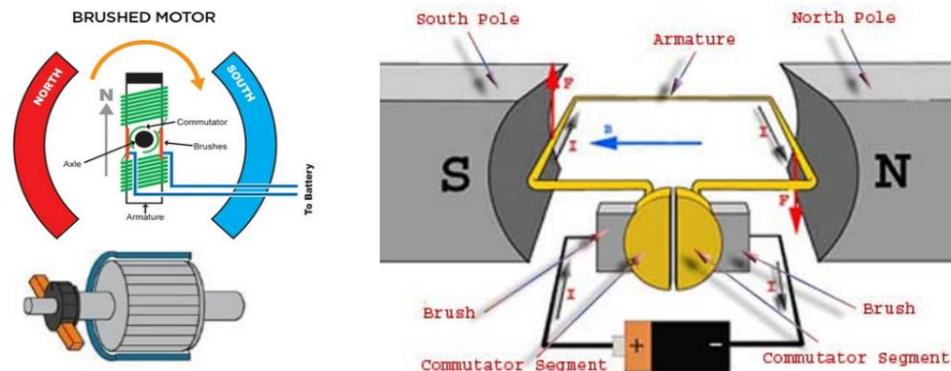
## 2. หลักการทำงานของ Brushed DC Motor

1) การหมุนของ Rotor เมื่อกระแสไฟฟ้าไหลผ่านขดลวดใน Rotor จะเกิดสนามแม่เหล็กขึ้น โดยสนามแม่เหล็กนี้จะมีปฏิกริยา กับ สนามแม่เหล็กของขั้วแม่เหล็กทราบที่อยู่ใน Stator ซึ่งทำให้เกิดแรงบิดที่ทำให้ Rotor หมุน

2) การแปลงทิศทางของกระแสไฟฟ้า Rotor ของมอเตอร์จะหมุนไปตามแรงบิดที่เกิดจากการติดต่อกันของชานมแม่เหล็ก แต่เมื่อ Rotor หมุนไปถึงจุดหนึ่ง กระแสไฟฟ้าภายในขดลวดของ Rotor จะต้องถูกกลับทิศทางเพื่อให้การหมุนดำเนินต่อไปได้ โดยระบบแปรรูปค่าและ Commutator จะทำหน้าที่สลับทิศทางกระแสไฟฟ้าในขดลวด เพื่อให้การหมุนของ Rotor มีทิศทางต่อเนื่อง

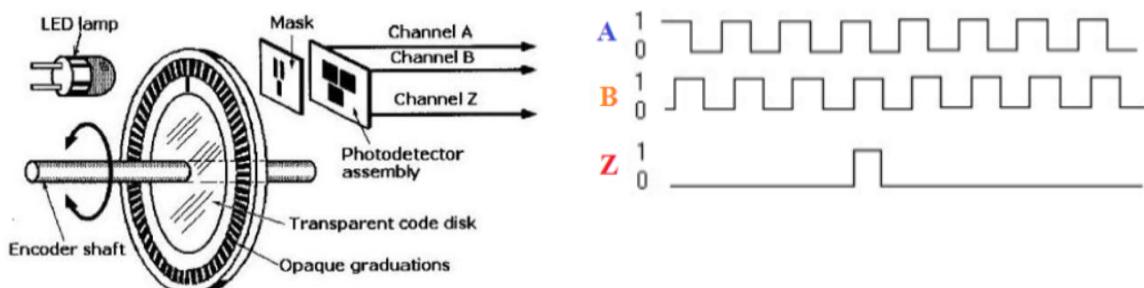
3) การทำงานของ Brushes จะทำการเชื่อมต่อกระแสไฟฟ้าไปยัง Commutator ซึ่งเป็นส่วนที่หมุนไปพร้อมกับ Rotor แปรรูปอิเล็กทรอนิกส์ให้กระแสไฟฟ้าไหลเข้าไปในชุดลวดของ Rotor และเมื่อ Rotor หมุน การสลับทิศทางกระแสไฟฟ้าจะเกิดขึ้นผ่าน Commutator

4) การสร้างแรงบิด สนามแม่เหล็กที่เกิดจากกระแสไฟฟ้าผ่านชุดลวดใน Rotor จะโต้ตอบกับสนามแม่เหล็กของขั้วแม่เหล็กภายใน Stator ทำให้เกิดแรงบิดที่หมุน Rotor ไปในทิศทางที่ต้องการ



### 3. Incremental Encoder

Incremental Encoder จะประกอบด้วย ajanหมุน และอุปกรณ์ตรวจจับ โดยajanหมุนจะมีช่องเล็ก ๆ เมื่อเพลาของมอเตอร์หมุนจะทำให้ajanหมุนไปตัดลำแสงของเซ็นเซอร์ (Sensor) ทำให้ชุดรับแสงได้รับสัญญาณเป็นช่วงๆ จึงทำให้สัญญาณเอาต์พุตออกมาเป็นสัญญาณพัลล์ต่อรอบ (PPR)



Incremental Encoder เป็น Encoder ที่ใช้หลักการเมื่อมีการหมุนของแกนเพลา จะทำให้มีสัญญาณเอาต์พุตที่เป็นสัญญาณลูกคลื่นพัลล์สี่เหลี่ยม (Square wave) มี 3 Tracks คือ A , B , Z โดยจะสัมพันธ์กับระยะการเคลื่อนที่และตำแหน่งสัญญาณเอาต์พุตของ Encoder A และ B มีมุมที่ห่างกัน 90 องศา ทางไฟฟ้า ส่วน Z จะมีสัญญาณ 1 พัลล์ ต่อ 1 รอบ หรือบางตัวจะเป็นพัลล์แบบ Invert เช่น A- , B- , Z- ซึ่งเป็นสัญญาณที่กลับเฟสกัน 90 องศา เพื่อเช็คทิศทางการหมุนของมอเตอร์ เป็นต้น

วิธีการใช้ Incremental Encoder

#### 1) การเชื่อมต่อ

- Vcc (Power Supply): แหล่งจ่ายไฟให้กับ encoder

- GND (Ground): การเชื่อมต่อสายกราวด์
- A, B (Output signals): สัญญาณพัลส์ที่ส่งออกจาก Encoder, ซึ่งมักจะใช้ในการคำนวณการหมุนและทิศทาง
- Optional Z (Index pulse): สัญญาณพัลส์หนึ่งที่สามารถใช้ในการรีเซ็ตตำแหน่งเริ่มต้นได้

### 2) การทำงานของสัญญาณ A และ B

- สัญญาณ A และ B: เป็นสัญญาณพัลส์ที่เกิดขึ้นในระหว่างการหมุน โดยสัญญาณ A และ B จะมีความส่วนพันธ์กันในลักษณะ Quadrature ซึ่งหมายถึง สัญญาณทั้งสองจะเลื่อนกันในเฟสต่างกัน 90 องศา ( $90^\circ$  phase shift) ซึ่งช่วยในการกำหนดทิศทางการหมุนได้
- หากสัญญาณ A เปลี่ยนก่อน B (หรือ A ล่าช้ากว่า B) จะบ่งบอกว่าการหมุนไปในทิศทางหนึ่ง (เช่น ทิศทางตามเข็มนาฬิกา)
- หากสัญญาณ B เปลี่ยนก่อน A (หรือ B ล่าช้ากว่า A) จะบ่งบอกว่าการหมุนไปในทิศทางตรงข้าม (ทิศทางทวนเข็มนาฬิกา)

### 3) Pulse Counting

ข้อมูลตำแหน่งที่ Encoder ส่งออกมาจะเป็นชุดของพัลส์ที่ต้องนับ เพื่อทราบจำนวนการหมุนหรือการเคลื่อนที่ที่เกิดขึ้น

- เมื่อสัญญาณ A และ B ส่งพัลส์ออกมาทั้งสองจะช่วยบอกถึงจำนวนการหมุน (Rotation Count) โดยนับจำนวนพัลส์ที่เกิดขึ้น
- ค่า Resolution ของ Encoder (จำนวนพัลส์ต่อรอบ) จะบอกว่าในแต่ละการหมุน 1 รอบจะส่งพัลส์ออกมากี่ครั้ง

#### 4) การคำนวณการหมุน:

- สามารถใช้สัญญาณ A และ B ในการคำนวณการหมุนทั้งหมดและทิศทางของการหมุน
- คำนวณจากจำนวนพัลส์ที่ได้ โดยใช้การนับจำนวนพัลส์ที่เกิดขึ้นเมื่อแปลงเป็นการหมุนจริง
- ตัวอย่างเช่น: ถ้า Encoder มี Resolution 1000 พัลส์ต่อรอบและได้รับพัลส์ 500 พัลส์จากสัญญาณ A หรือ B ในการหมุนครึ่งรอบ ก็จะรู้ได้ว่าเกิดการหมุนครึ่งรอบแล้ว

## 4. Unwrap

เป็นการแก้ไขมุม หรือการจัดการข้อมูลเกี่ยวกับมุมหรือเฟส ซึ่งจะใช้ในกรณีที่มุมมีค่าเกินช่วงที่ต้องการ และต้องการให้ค่าอยู่ในช่วงที่เหมาะสมสมหรือเป็นมุมที่ต้องเนื่อง

ถ้าจะทำการ Unwrap ข้อมูลจาก Incremental Encoder มักจะเกี่ยวข้องกับการแปลงการหมุนที่ได้จากสัญญาณพัลส์ของ Encoder ให้เป็นมุมที่มีหน่วยเป็น radians และคำนวณความเร็วเชิงมุม ในหน่วย rad/s ขั้นตอนในการทำ Unwrap ข้อมูลจาก Incremental Encoder เพื่อคำนวณในหน่วย rad/s

1) การรับสัญญาณจาก Encoder สัญญาณที่ Encoder ให้มักจะเป็นพัลส์ ซึ่งจะต้องคำนวณเพื่อแปลงเป็นมุม

2) แปลงพัลส์เป็นมุม (Conversion from pulses to angle)

- สำหรับ Encoder ที่มี 1000 PPR ทำได้โดยใช้สูตร

$$\text{Angle in radians} = \frac{\text{Number of pulses}}{\text{PPR}} \times 2\pi$$

3) การ Unwrap มุม

- การ Unwrap คือการทำให้มุมที่ได้ไม่ข้ามขอบเขต ( เช่น  $2\pi$ ) หรือ  $-\pi$  ถึง  $\pi$
- หากมุมที่คำนวณได้มีการกระโดดข้ามขอบเขต ( เช่น จาก  $360^\circ$  ไป  $-360^\circ$  หรือจาก  $2\pi$  ไป  $-2\pi$ ) จะทำการปรับให้มุมต่อเนื่อง โดยเพิ่มหรือลบ  $2\pi$  ให้เหมาะสม
- การ Unwrap นี้จะทำให้มุมมีการเปลี่ยนแปลงอย่างต่อเนื่อง ไม่เกิดการย้อนกลับ

4) การคำนวณความเร็วเชิงมุม

- หลังจากที่เรามีมุมที่ทำการ Unwrap แล้ว สามารถคำนวณความเร็วเชิงมุมได้โดยใช้การเปลี่ยนแปลงของมุมในช่วงเวลาหนึ่ง
- สูตรคือ

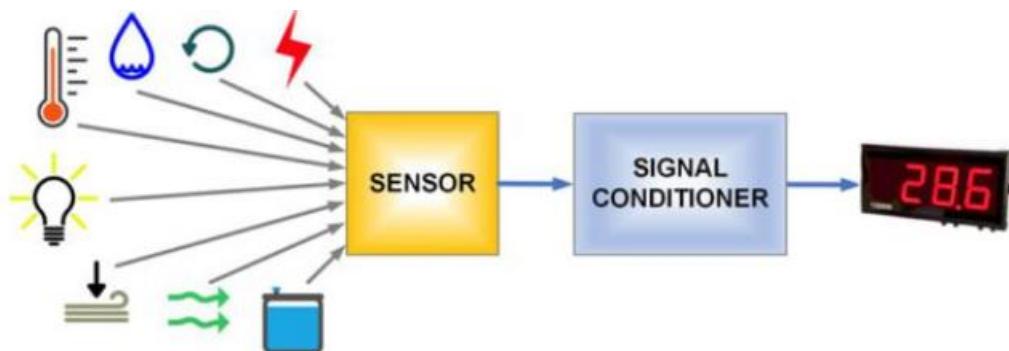
$$\omega = \frac{\Delta\theta}{\Delta t}$$

โดยที่  $\omega$  คือ ความเร็วเชิงมุมในหน่วย rad/s

$\Delta\theta$  คือ การเปลี่ยนแปลงของมุม ซึ่งได้จากการ Unwrap

$\Delta t$  คือ ช่วงเวลาที่ใช้ในการเปลี่ยนแปลงมุม

## 5. Signal Conditioner



อุปกรณ์ฮาร์ดแวร์ หรือ วงจรอิเล็กทรอนิกส์ ที่ใช้ในการดำเนินการ Signal Conditioning เป็นอุปกรณ์ที่ออกแบบมาเพื่อปรับสัญญาณจาก sensor ให้มีระดับแรงดันที่เหมาะสม หรือการกรองสัญญาณรบกวน

## พังก์ชันหลักของ Signal Conditioner

### 1) การขยายสัญญาณ (Amplification)

บางครั้งเซ็นเซอร์หรือแหล่งสัญญาณอาจผลิตสัญญาณที่มีแรงดันต่ำ ซึ่งไม่สามารถนำไปใช้งานได้โดยตรง การขยายสัญญาณทำให้ระดับแรงดันสูงขึ้น เพื่อให้สัญญาณสามารถนำไปประมวลผลหรือส่งไปยังระบบอื่น ๆ ได้

### 2) การกรองสัญญาณ (Filtering)

เครื่องปรับสัญญาณสามารถกรองสัญญาณรบกวนหรือสัญญาณที่ไม่ต้องการออกจากสัญญาณหลัก เช่น การกรองสัญญาณที่มีความถี่สูง (high-frequency noise) หรือการกรองสัญญาณที่มีความถี่ต่ำ (low-frequency noise) ซึ่งอาจส่งผลกระทบต่อความแม่นยำของการวัด

### 3) การแปลงสัญญาณ (Conversion)

เครื่องปรับสัญญาณบางประเภทสามารถทำการแปลงสัญญาณจากรูปแบบหนึ่งไปยังอีกรูปแบบหนึ่ง เช่น การแปลงจากสัญญาณอะนาล็อก (analog) เป็นสัญญาณดิจิทัล (digital) หรือการแปลงจากสัญญาณกระแส (current) เป็นสัญญาณแรงดัน (voltage)

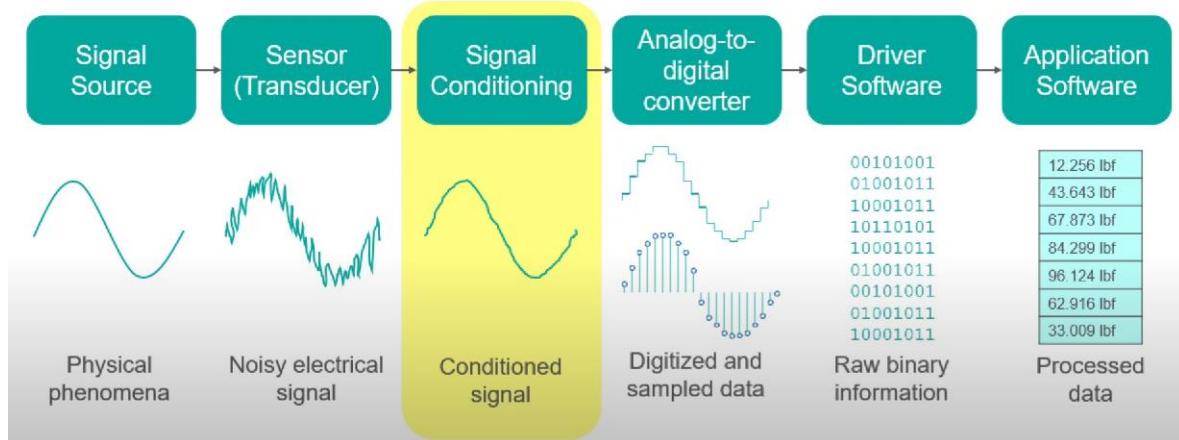
### 4) การแยกสัญญาณ (Isolation)

การแยกสัญญาณเป็นการป้องกันการส่งผลกระทบจากความแตกต่างของแรงดันหรือกระแสระหว่างระบบ โดยการใช้ isolator หรือ transformer เพื่อแยกสัญญาณออกจากกัน ซึ่งช่วยป้องกันความเสียหายจากการสัมผัสแรงดันสูง หรือป้องกันการรบกวนจากภายนอก

### 5) การปรับค่าเชิงเส้น (Linearization)

เซ็นเซอร์บางตัวอาจให้สัญญาณที่มีการตอบสนองไม่เป็นเส้นตรง (non-linear) การใช้ Signal Conditioner สามารถปรับค่าของสัญญาณให้เป็นเชิงเส้น (linearization) เพื่อให้การอ่านค่ามีความแม่นยำมากขึ้น

## 6. การ signal conditioning



กระบวนการที่ใช้ในการปรับแต่งสัญญาณที่ได้จากเซ็นเซอร์หรืออุปกรณ์วัดอื่น ๆ เพื่อให้สามารถนำไปใช้งานได้ในระบบวิเคราะห์หรือควบคุมต่าง ๆ กระบวนการนี้จะทำการปรับสัญญาณให้มีลักษณะที่เหมาะสมและสามารถใช้งานได้ในระดับที่ต้องการ โดยอาจจะเกี่ยวข้องกับการกรอง (filtering), การขยาย (amplification), การแปลง (conversion), หรือการปรับช่วง (scaling) เพื่อให้สัญญาณมีคุณสมบัติตรงตามความต้องการของระบบ

### วัตถุประสงค์ของ Signal Conditioning

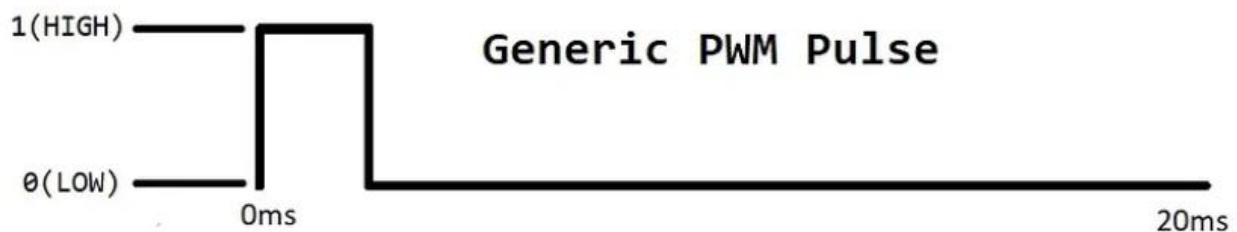
- 1) ปรับให้สัญญาณอยู่ในรูปแบบที่เหมาะสม: เช่น ปรับระดับแรงดันไฟฟ้าให้สามารถอ่านค่าจากอุปกรณ์วัดหรือเซ็นเซอร์ได้อย่างถูกต้อง
- 2) กรองสัญญาณรบกวน: เพื่อลบหรือบรรเทาสัญญาณรบกวน (noise) ที่อาจทำให้ผลลัพธ์ไม่ถูกต้อง
- 3) การแปลงสัญญาณ: เช่น การแปลงจากสัญญาณอะนาล็อก (analog) เป็นสัญญาณดิจิทัล (digital) หรือจากกระแส (current) เป็นแรงดัน (voltage)
- 4) เพิ่มความสามารถในการอ่านค่าของเซ็นเซอร์: เช่น การขยายสัญญาณ (amplification) เพื่อให้อ่านได้ในระดับที่ต้องการ

### ส่วนประกอบหลักของ Signal Conditioning

- 1) Amplifiers (ตัวขยายสัญญาณ): ใช้ขยายสัญญาณจากเซ็นเซอร์หรืออุปกรณ์ที่มีสัญญาณอ่อนเพื่อให้มีระดับที่สามารถประมวลผลได้
- 2) Filters (ตัวกรอง): ใช้กรองสัญญาณรบกวนหรือความถี่ที่ไม่ต้องการจากสัญญาณที่รับเข้ามา
- 3) Analog-to-Digital Converters (ADC) และ Digital-to-Analog Converters (DAC): ใช้ในการแปลงสัญญาณอะนาล็อกเป็นดิจิทัล (ADC) หรือแปลงสัญญาณดิจิทัลกลับเป็นอะนาล็อก (DAC)
- 4) Linearization: บางเซ็นเซอร์จะให้สัญญาณที่ไม่เป็นเชิงเส้น ซึ่งอาจต้องมีการปรับให้เป็นเชิงเส้นเพื่อให้การอ่านค่าถูกต้อง
- 5) Isolation: ใช้แยกสัญญาณที่รับเข้ามาจากภาคอื่นของระบบเพื่อป้องกันการรบกวนและความเสียหายที่อาจเกิดจากการสัมผัสกับแรงดันที่สูง

## 7. Pulse Width Modulation

เป็นเทคนิคที่ทำให้เราสามารถอ่าน/เขียนข้อมูลแบบ analog ด้วยสัญญาณ digital ได้ โดยตัวควบคุมการสร้างสัญญาณดิจิตอล (Digital control) จะสร้างสัญญาณคลื่นสี่เหลี่ยมออกมายอดังรูปที่ 1 โดยสัญญาณที่สร้างออกมายจะสลับกันระหว่าง เปิด(HIGH) กับ ปิด(LOW) รูปแบบสัญญาณเปิด-ปิดนี้สามารถจำลองเป็นแรงดันไฟฟาระหว่าง เปิด (5 Volts) กับ ปิด (0 Volts)



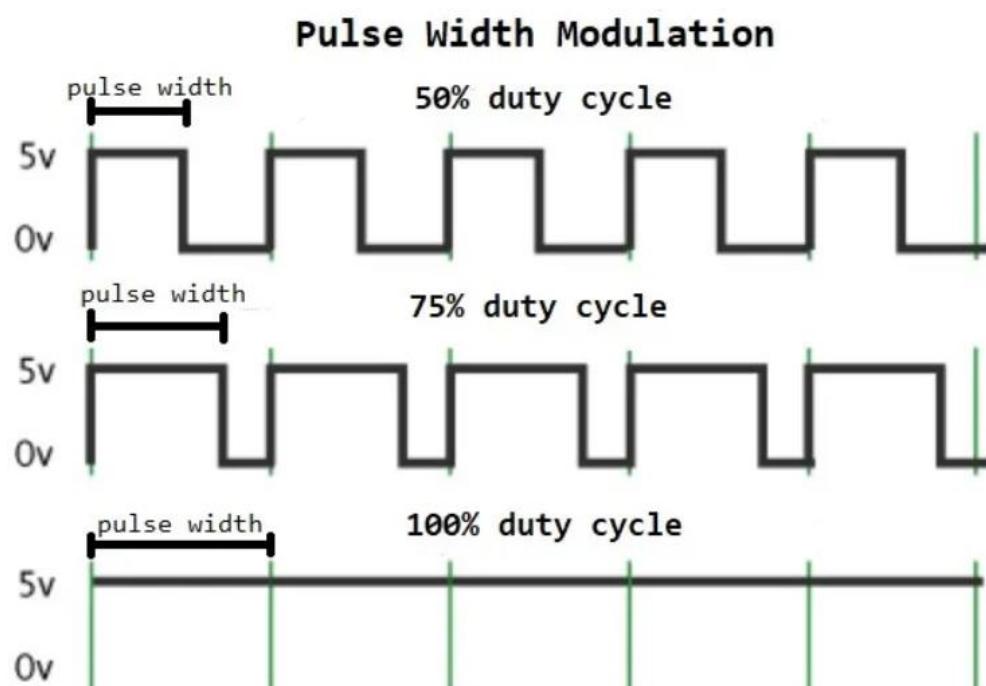
### หลักการทำงานของ PWM

PWM ใช้การสลับระหว่างสถานะ เปิด (high) และ ปิด (low) ของสัญญาณในระยะเวลาหนึ่ง (period) โดยการควบคุม ความกว้างของพัลส์ หรือ Duty Cycle เพื่อกำหนดพลังงานที่จ่ายให้กับโหลดในช่วงเวลาหนึ่ง

- Period: คือเวลาที่ใช้ในการทำหนึ่งรอบของพัลส์ (ช่วงที่มีทั้งการเปิดและปิด)
- Duty Cycle: คืออัตราส่วนระหว่างเวลาที่สัญญาณเป็น "เปิด" (high) ต่อเวลาเต็มหนึ่งรอบ (period) โดยจะมีหน่วยเป็นเปอร์เซ็นต์

หาก Duty Cycle เป็น 50% หมายถึงสัญญาณจะอยู่ในสถานะ "เปิด" ครึ่งหนึ่งของเวลาหนึ่งรอบ

หาก Duty Cycle เป็น 100% หมายถึงสัญญาณจะอยู่ในสถานะ "เปิด" ตลอดเวลา



## วิธีดำเนินการทดลอง

เริ่มจากการ ปรับปรุงค่าสัญญาณที่อ่านได้จาก sensor 3 ตัว ดังนี้ Current Sensor, Load Cell, Incremental Encoder ให้สามารถอ่านได้ในหน่วย SI และเริ่มวัดค่าความเร็วของ DC Motor พร้อม กับ current ที่ DC Motor ใช้โดยการต่อเพลา DC Motor ผ่านพูลเลอร์ เข้ากับเพลา ของ incremental Encoder แล้ว ทำการเพิ่ม PWM จากผังลบ สูงสุด จนถึง 0 แล้วจาก 0 ถึง ผังวงสูงสุด และหาค่า torque และ current โดยที่เพลา DC motor จะมีไม่สำหรับกด Load Cell อญี่ โดยจะสั่ง PWM จาก 0 ถึงสูงสุด ไปในทางทิศที่ไม่กด load cell

## วัสดุอุปกรณ์

1. Nidec Components Geared DC Geared Motor, 12 V dc, 20 Ncm, 70 rpm, 6mm Shaft Diameter จำนวน 1 อัน
2. Incremental Encoder AMT103-V จำนวน 1 อัน
3. Warner Electric Magnetic Particle Clutches MPB12 จำนวน 1 อัน
4. WCS1700 Hall Current Sensor จำนวน 1 อัน
5. Cytron MDD20A Motor Driver จำนวน 1 อัน
6. Nucleo STM32G474RE พร้อมสายอ้อปโหลด จำนวน 1 ชุด
7. MotorXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบ กับ DC Motor

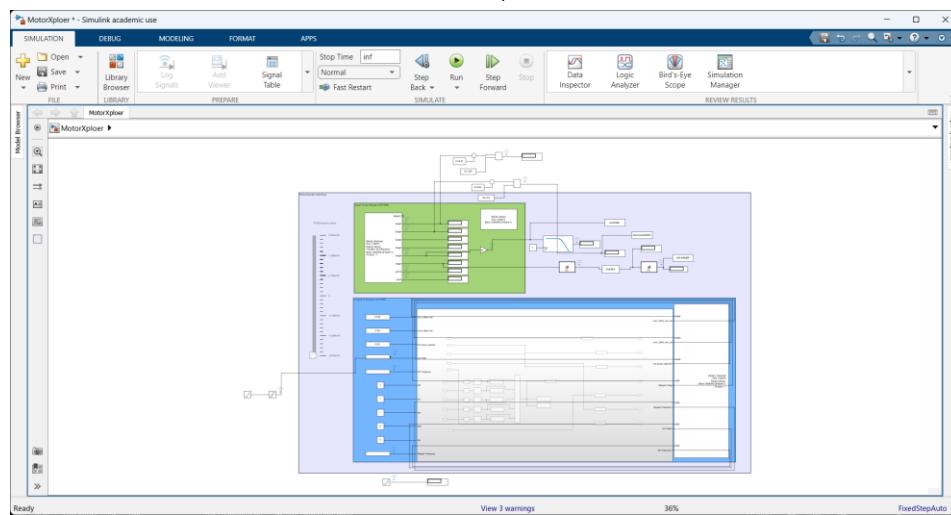
## ขั้นตอนการทดลอง

1. เชื่อมต่อ Nucleo STM32G474RE เข้ากับบอร์ด MotorXplorer ทำการจ่ายไฟ 24V และเปลี่ยนรูปแบบการใช้ไฟ เป็น E5V
2. จ่ายไฟให้ DC Motor โดยใช้ power supply ที่จ่ายแรงดัน 15 V และกระแส 10 A
3. เชื่อมต่อสายไฟของ sensor ต่างๆดังนี้
  - Incremental Encoder AMT103-V
    - 5V -> 3V3
    - GND -> GND
    - A -> PA6
    - B -> PA7
  - Magnetic Encoder
    - SCL -> PB8
    - SDA -> PB9

- DIR -> GND
- VCC -> 3V3
- GND -> GND
- Load Cell
  - OUT -> A3

หมายเหตุ: sensor ทุกตัวต้องต่อสายไฟอย่างถูกต้องไม่เช่นนั้น โปรแกรม MotorXploer.slx จะไม่สามารถทำงานได้

#### 4. เข้า Simulink ใน MATLAB เพื่อทำการเปิดไฟล์สำหรับควบคุมบอร์ด MotorXploer.slx



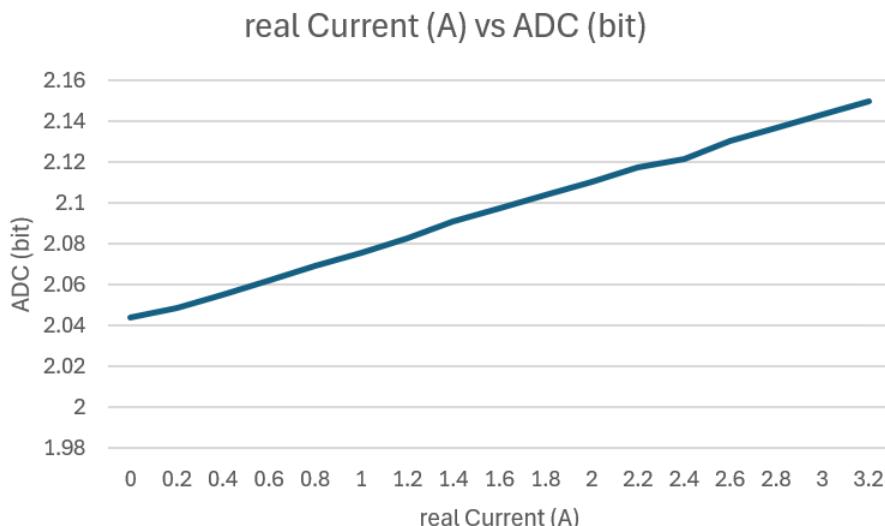
5. ติดตั้ง extension ที่ชื่อ Waijang ลงใน MATLAB
6. ทำการอัพโหลดไฟล์ bin ลงใน Nucleo STM32G474RE จากนั้นเลือก com port เพื่อการส่งข้อมูลให้ถูกต้อง
7. ปรับปรุงการรับค่าสัญญาณจากเซนเซอร์:

- WCS1700 Hall Current Sensor ผ่านการทำ Signal Conditioning ซึ่งแต่ละ sensor ก็จะมีค่าที่อ่านได้ไม่ตรงกัน เราจึงต้องใช้ กระแส ที่สามารถวัดได้จริง (ตั้งแต่ 0 A ไปจนถึง 3.2 A) และวัดค่า ADC ที่ sensor อ่านได้เพื่อสร้างสมการเชิงเส้น

$$ADC = m \times real\ Current + c$$

เพื่อแปลงค่าดิจิทัล (ADC Bit) ให้เป็นกระแส (Current) จริง

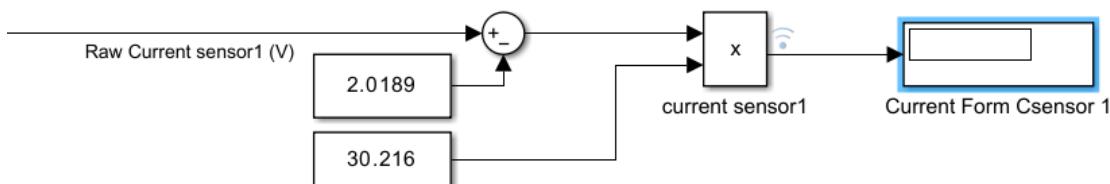
real	ADC
0	2.0437
0.2	2.0485
0.4	2.0552
0.6	2.062
0.8	2.0691
1	2.0758
1.2	2.0824
1.4	2.0911
1.6	2.0973
1.8	2.1038
2	2.1103
2.2	2.117
2.4	2.1215
2.6	2.1303
2.8	2.1366
3	2.1432
3.2	2.1497



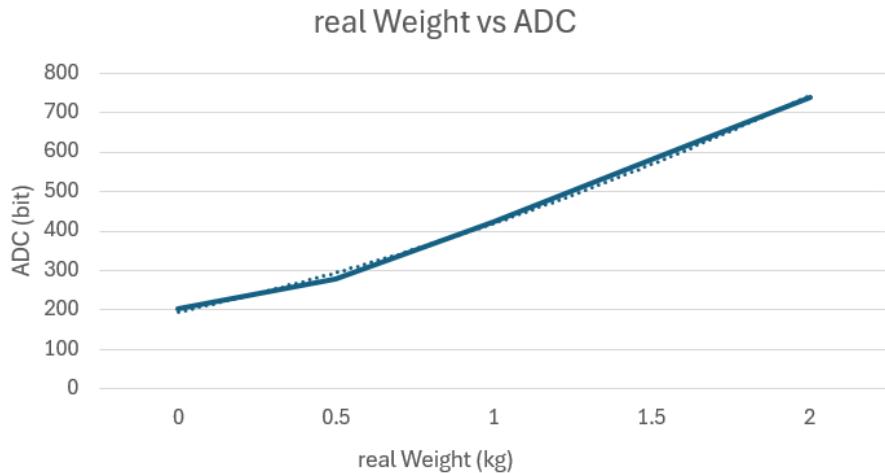
จึงได้สมการ

$$\text{real Current} = (\text{ADC} - 2.0189) \times 30.216$$

และทำการสร้างบล็อกสมการใน Simulink



- Single Point Load Cell YZC-131A สามารถทำวิธีเดียวกันกับ Current Sensor ได้แต่แม่ทදลงดูค่า น้ำหนักมาตรฐานที่เรารู้อยู่แล้ว กับค่า ADC จะเห็นว่ามีความสัมพันธ์ไม่เป็นสมการเชิงเส้น จึงต้องใช้ สมการ polynomial (สมการที่ตัวแปรมีมากกว่า หนึ่งตัวคงรี) เพื่อช่วยในการแปลงค่า ADC ให้เป็นน้ำหนัก ที่สามารถอ่านได้จาก Load Cell



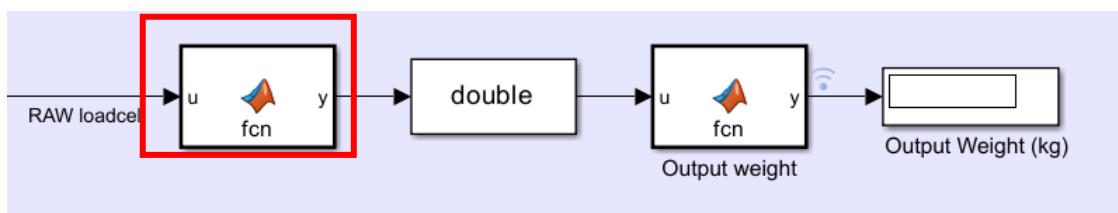
โดยสมการที่ได้คือ

$$Weight = (-2.2389 \times 10^{-6}) ADC^2 + 0.005698 ADC - 1.0055$$

นำสมการที่ได้มาใส่ในบล็อก MATLAB Function ใน Simulink

```
function y = fcn(u)
y = (- 2.2389*10^-6) * u^2 + 0.005698*u - 1.0055;

end
```



- Incremental Encoder AMT103-V เนื่องจากในไฟล์ MotorXploer.slx ทำการ EQI มาแล้ว เราจึงสามารถอ่านค่า ความเร็วรอบ (rad/s) ที่อ่านมาจาก Incremental Encoder ได้เลย แต่เนื่องจาก สัญญาณที่ได้ยังมีสัญญาณรบกวนอยู่ จึงต้องทำการตัดสัญญาณที่เราไม่ได้สนใจออกโดยใช้ Varying Lowpass Filter โดยเราราสามารถหาความถี่ของสัญญาณที่เราต้องการได้จากการ สั่งงาน Motor ด้วย ความถี่ 1 KHz และนำค่าความเร็ว rad/s ที่อ่านได้ไปเข้าสูตรใน MATLAB โดยใช้คำสั่ง fft() เพื่อหา frequency ที่มีในสัญญาณที่อ่านมา

```

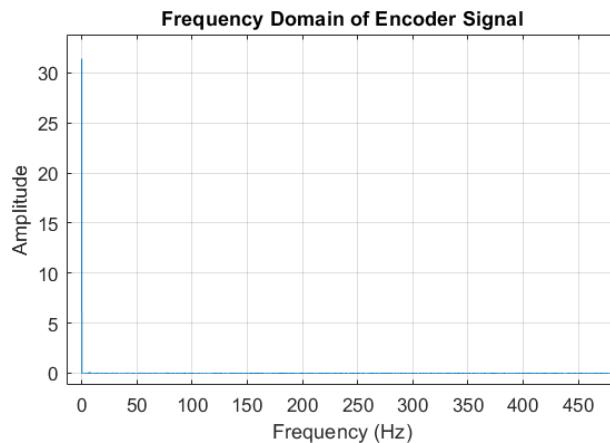
signal = out.simout; % Assuming the second column contains the signal
Fs = 1000; % Sampling frequency (Hz)
T = 1 / Fs; % Sampling period
L = length(signal); % Length of signal
t = (0:L-1) * T; % Time vector

% Perform FFT
Y = fft(signal);
P2 = abs(Y / L); % Two-sided spectrum
P1 = P2(1:L/2+1); % Single-sided spectrum
P1(2:end-1) = 2 * P1(2:end-1); % Adjust for symmetry
f = Fs * (0:(L/2)) / L; % Frequency vector

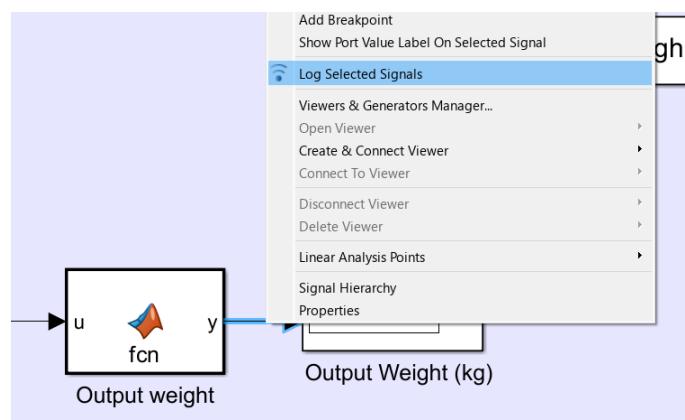
% Plot Frequency Spectrum
figure;
plot(f, P1);
title('Frequency Domain of Encoder Signal');
xlabel('Frequency (Hz)');
ylabel('|P1(f)|');
grid on;

```

เราจะได้กราฟแสดง frequency domain ที่จะเห็นว่าช่วงความถี่ที่ สูงกว่า 1 Hz ขึ้นไปเราไม่จำเป็นต้องสนใจจึงสามารถใช้ Low pass filter ที่มี cutoff frequency อยู่ที่ 1 Hz ได้



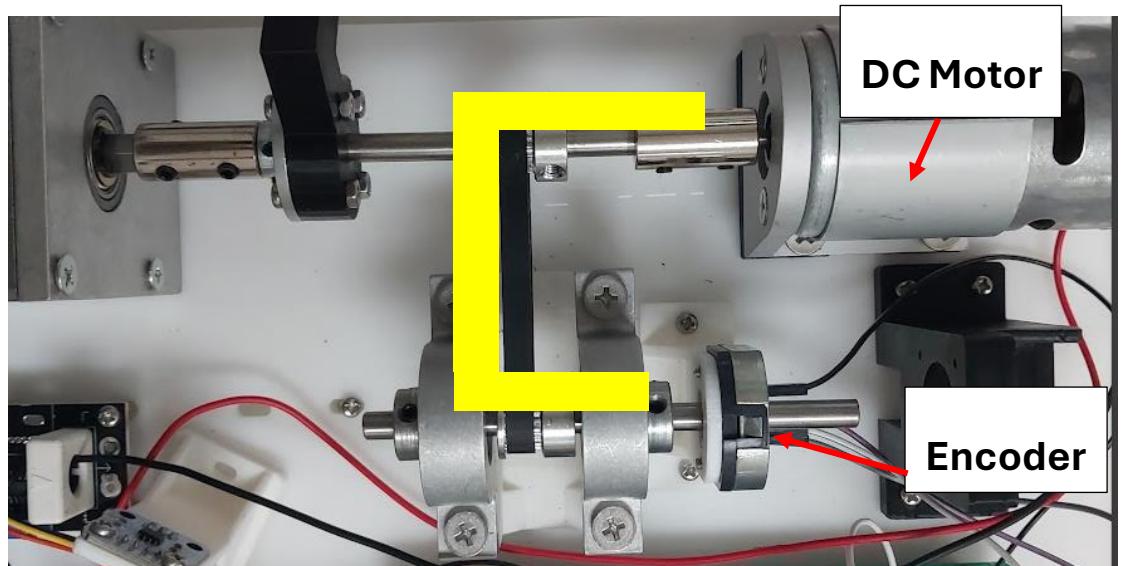
#### 8. เก็บข้อมูลเบื้องต้นด้วยการ Log Signal:



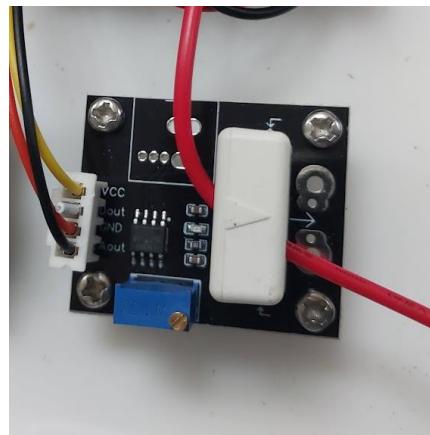
- เก็บค่าสัญญาณความเร็วรอบ (rad/s) จาก Incremental Encoder
- เก็บค่าสัญญาณกระแสไฟฟ้า (Current) ที่อ่านได้จาก Hall Current Sensor
- เก็บค่าแรงหนักหรือ Torque จาก Load Cell
- เก็บค่าความถี่ PWM ที่สั่ง DC Motor

9. วัดความเร็วของ DC Motor และกระแสไฟฟ้า (Current):

- เชื่อมต่อ DC Motor กับพูลเลย์และเพลา Incremental Encoder เพื่อสร้างสถานะ No Load แล้วสั่งงาน PWM จน DC Motor มีความเร็วสูงสุด

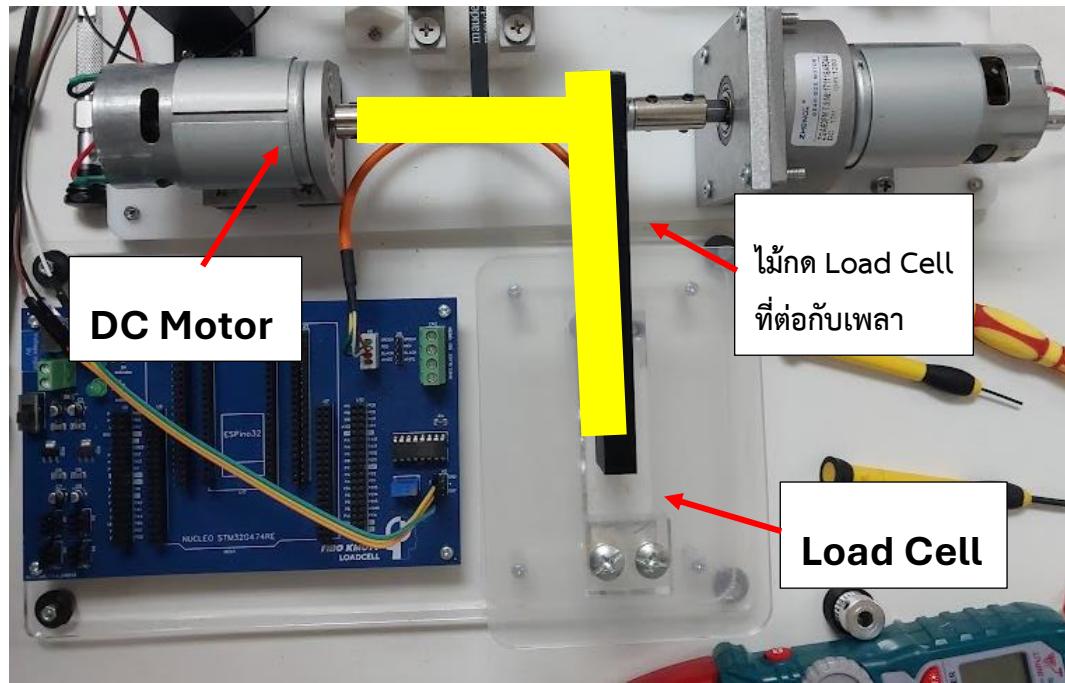


- สั่งความถี่ PWM จากค่าลับที่มากที่สุดจนถึง 0 และจาก 0 ไปจนถึงค่าบวกสูงสุด
- วัดค่า Speed (rad/s) และ Current (A) ในทุกค่าของความถี่ PWM โดยนำสายไฟขึ้นบวกที่เข้า Motor ผ่าน ด้านที่มีลูกศรบน Current sensor

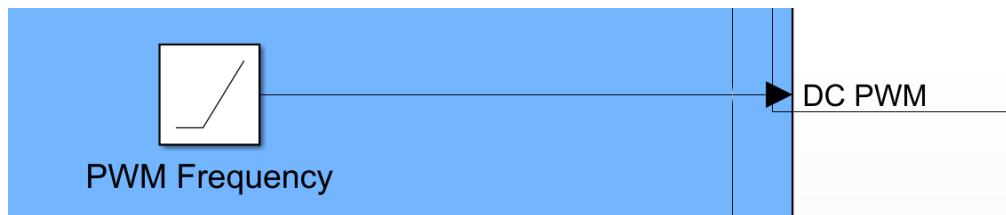


10. วัด Torque และกระแสไฟฟ้า (Current):

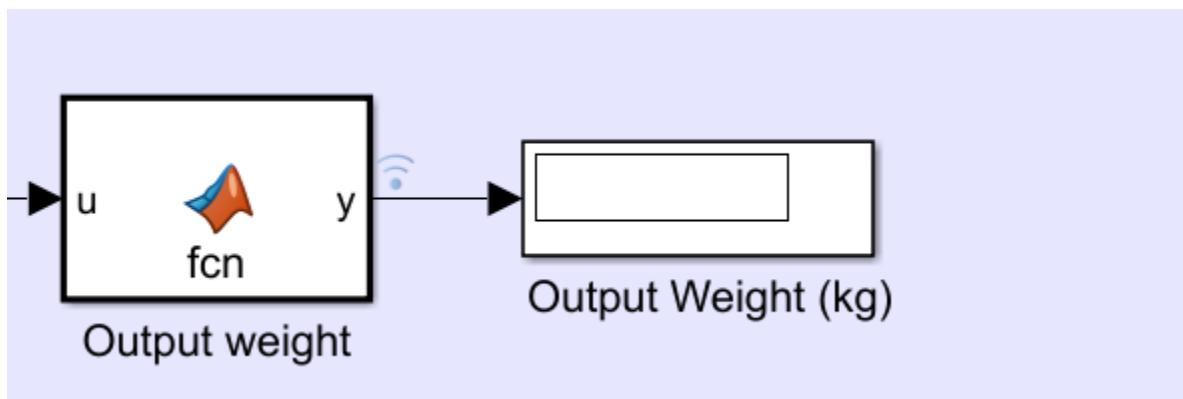
- ติดตั้งไม้กัด Load Cell เพื่อกด Load Cell เพื่อสร้างสภาวะ Stall Torque



- สั่งความถี่ PWM frequency จาก 0 ถึงค่าบวกสูงสุดในทิศทางที่ไม้กัด Load Cell ด้วยบล็อกคำสั่ง Ramp ใน Simulink



- วัดค่า Torque และ Current ที่แต่ละค่าของความถี่ PWM

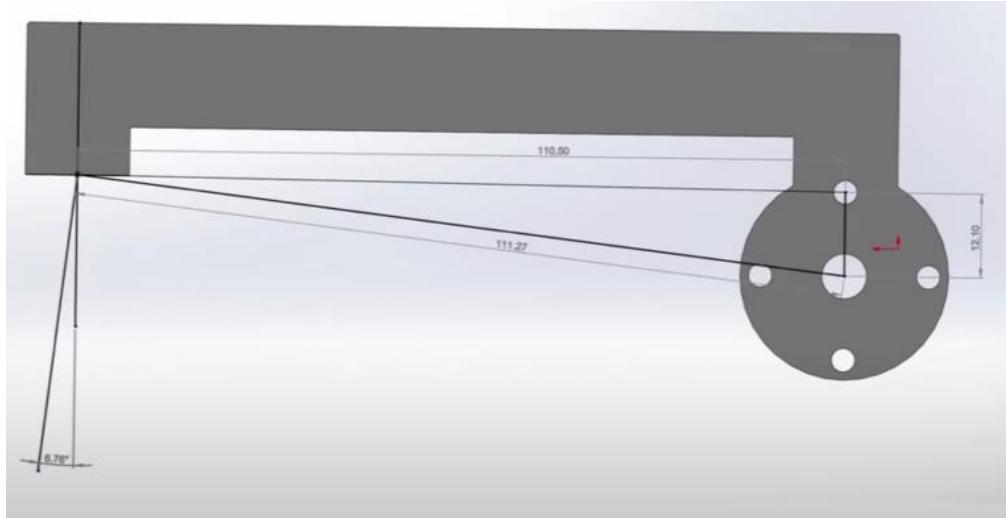


นำ Output Weight(kg) มาทำเป็น Torque โดยเริ่มจากเปลี่ยน กิโลกรัมเป็นนิวตัน โดยการคูณแรงโน้มถ่วง กับน้ำหนัก Output Weight(kg) ที่กดจากไม้ที่กดลงบน Load Cell

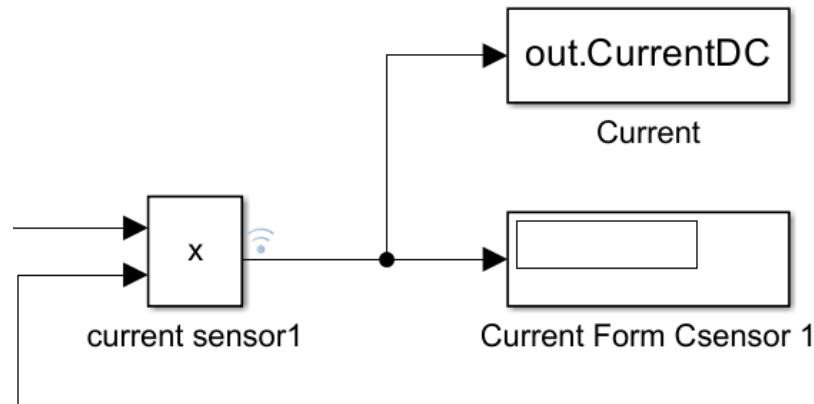
$$F = mg$$

และเข้าสูตร Torque ที่มีการใช้รัศมี จากจุดศูนย์กลางเพลาถึงจุดที่แรงเกิดการกด โดยรัศมี คือ 0.11127 เมตร และสูตรหา Torque คือ

$$\tau = Fr$$



และการอ่านค่ากระแสที่ DC Motor กำลังใช้จากการ Log Signal สัญญาณ Current Sensor 1



#### 11. วิเคราะห์และสร้างกราฟ Motor Characteristic:

- นำค่าที่ได้ทั้งหมดมาสร้างกราฟต่อไปนี้:
  - Torque vs Current

```
% Torque-Current Curve
xTorqueCurrent = [0, stallTorque]; % Torque range
yCurrent = [I_No_load, I_Stall]; % Current range
ratedCurrent = interp1(xTorqueCurrent, yCurrent, ratedTorque, 'linear'); % Current at rated torque
```

- Torque vs Speed

```
% Torque-Speed Curve
xTorqueSpeed = [0, stallTorque]; % Torque range
ySpeed = [maxSpeed, 0]; % Speed range
ratedSpeed = interp1(xTorqueSpeed, ySpeed, ratedTorque, 'linear'); % Speed at rated torque
```

- Power

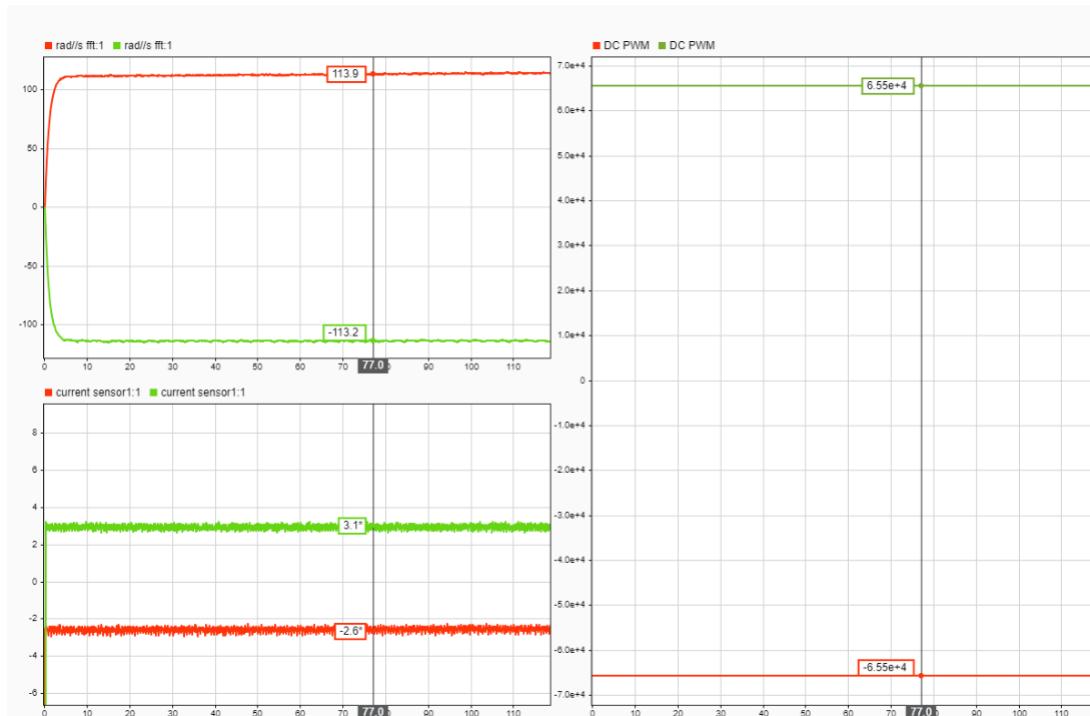
```
% Power Curve
Torque2 = linspace(0, stallTorque, 1000); % Torque from 0 to Stall Torque
P = -(maxSpeed/stallTorque)*((Torque2).^2) + maxSpeed*Torque2;
```

- Efficiency vs Torque

```
% Efficiency Curve
Torque1 = linspace(0, stallTorque, 1000); % Torque from 0 to Stall Torque
n = ((-(maxSpeed/stallTorque)*((Torque1).^2) + maxSpeed*Torque1) ./ ...
(((I_Stall - I_No_load)/stallTorque)*Torque1*Vin + I_No_load*Vin)) * 100;
```

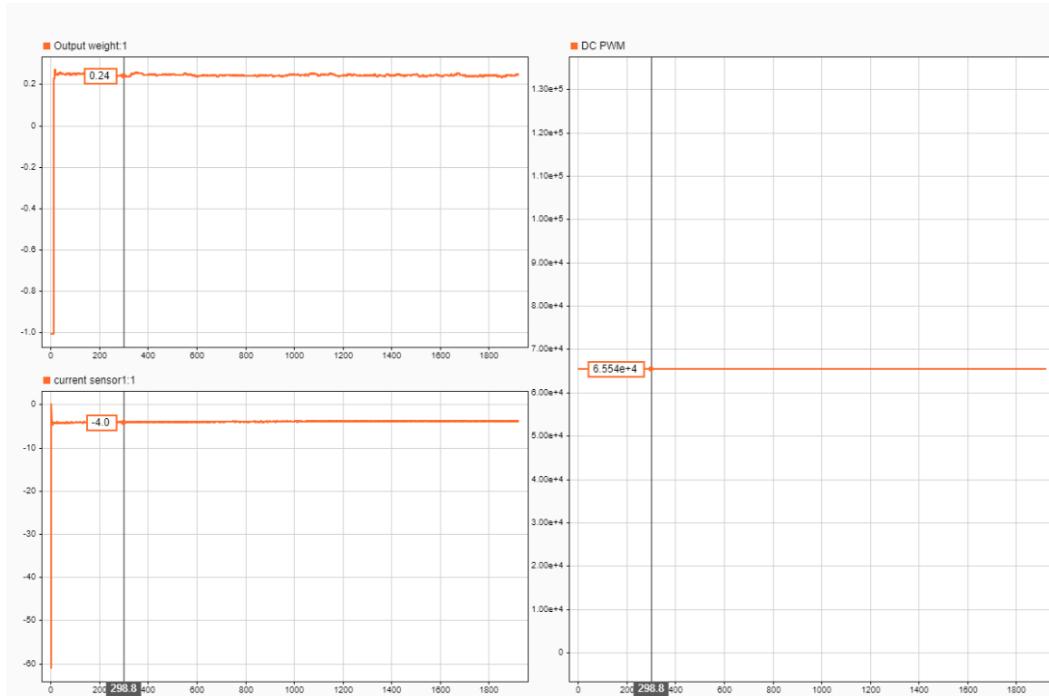
### ผลการทดลอง

ความเร็วที่วัดได้จาก Encoder และกระแสที่ใช้งานขณะ No load คือ 113.2 Rad/s และ 3.1 A ตามลำดับ เมื่อสั่ง PWM frequency ที่ 65,535 Hz



รูปแสดงกราฟความเร็วที่วัดได้จาก Encoder และกระแสที่ใช้งานขณะ No load

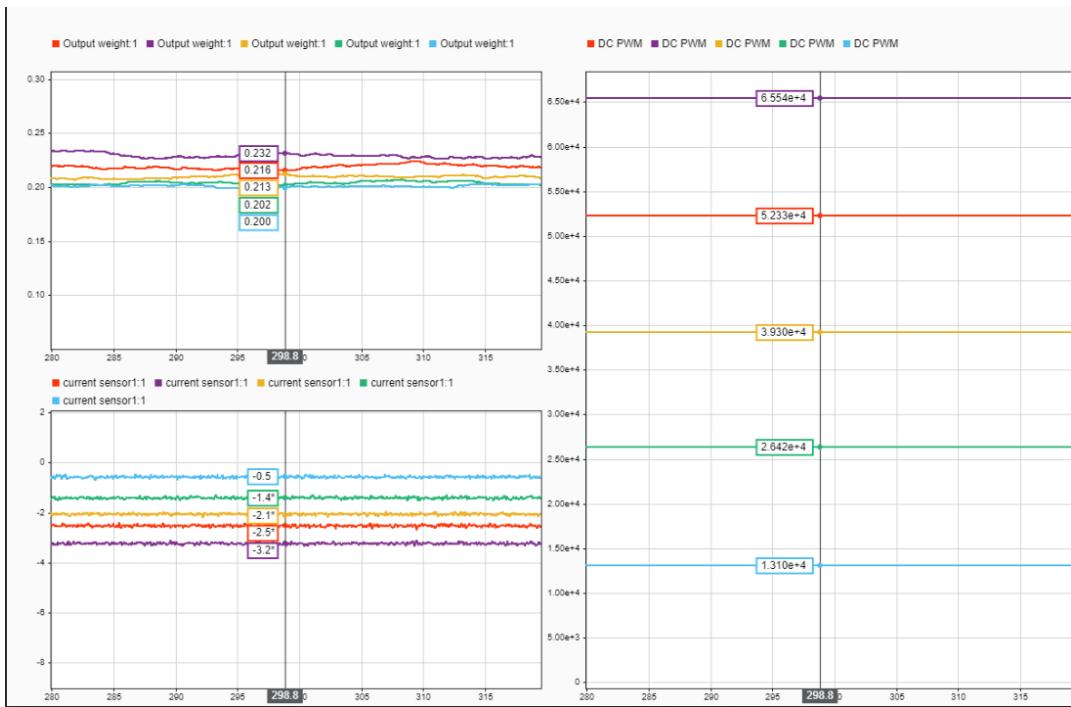
Stall Torque เมื่อสั่ง PWM frequency สูงสุดที่ 65,535 Hz วัดค่าที่ได้จากการ Signal conditioning Loadcell คือ 0.24 Nm และกระแสที่ใช้งานขณะ Stall Torque คือ 4 A



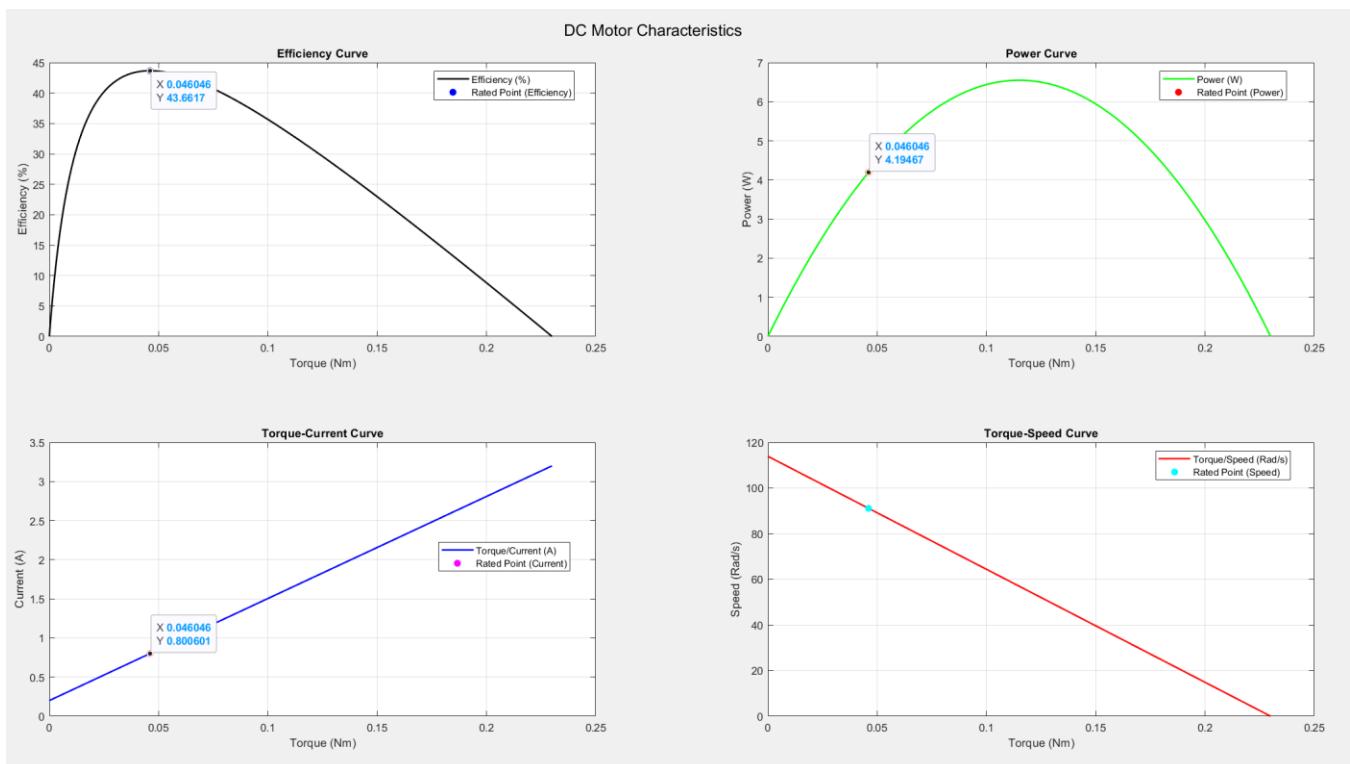
รูปแสดงกราฟ Stall Torque ที่วัดได้จากการ Signal conditioning Loadcell และกระแสที่ใช้งานขณะ Stall Torque

ในขณะ Stall Torque โดยเปลี่ยน Duty Cycle จะมี Torque และ Current ที่ทำได้ และ ตารางที่เปลี่ยนไปดังนี้

Duty Cycle (%)	Torque (Nm)	Current (A)
0	0	0
20	0.218	0.5
40	0.220	1.4
60	0.233	2.1
80	0.236	2.5
100	0.253	3.2



รูปแสดงกราฟกระแสที่ได้รับจาก Hall Current Sensor ที่ผ่านการ signal conditioning และ เทียบกับ  
ความถี่ที่ส่งงาน PWM



## สรุปผล

### 1. แรงดันไฟฟ้ากับความเร็วรอบ (Voltage vs Speed)

เมื่อเพิ่มแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ ความเร็วรอบของมอเตอร์เพิ่มขึ้นอย่างสม่ำเสมอจนถึงจุดที่แรงดันไฟฟ้าสูงสุดที่มอเตอร์รองรับได้

ความสัมพันธ์ระหว่างแรงดันไฟฟ้าและความเร็วเป็นเส้นตรง (Linear Relationship) ในช่วงแรงดันที่กำหนด

### 2. แรงบิดกับกระแสไฟฟ้า (Torque vs Current)

พบว่ามีความสัมพันธ์เชิงเส้นตรงระหว่างแรงบิดของมอเตอร์และกระแสไฟฟ้าที่จ่ายเข้า

กระแสไฟฟ้าเพิ่มขึ้นเมื่อมอเตอร์ต้องรับภาระโหลดที่มากขึ้น (Load Increasing)

### 3. ประสิทธิภาพของมอเตอร์ (Efficiency)

ที่แรงบิดต่ำ มอเตอร์มีประสิทธิภาพสูงสุด และเมื่อแรงบิดเพิ่มขึ้นจนถึงจุดที่มอเตอร์รับภาระมาก ประสิทธิภาพจะลดลงเนื่องจากการสูญเสียพลังงานในรูปของความร้อน

### 4. ผลกระทบของ PWM ต่อความเร็ว

การควบคุมความเร็วด้วย PWM ช่วยให้สามารถปรับความเร็วรอบของมอเตอร์ได้อย่างแม่นยำโดยไม่ทำให้แรงบิดลดลง

## อภิปรายผล

ผลการทดลองนี้สอดคล้องกับหลักการทำงานของ DC Motor ตามทฤษฎี และสามารถนำผลที่ได้ไปประยุกต์ใช้ในการออกแบบระบบควบคุมมอเตอร์หรือระบบขับเคลื่อนที่ใช้มอเตอร์กระแสตรง เช่น ในเครื่องจักรอุตสาหกรรมหรือหุ่นยนต์ โดยเน้นการจัดการพลังงานและการควบคุมโหลดเพื่อให้ได้ประสิทธิภาพสูงสุด

## ข้อเสนอแนะ

ควรเพิ่มการทดลองในช่วงแรงดันและกระแสที่สูงขึ้นเพื่อวิเคราะห์พฤติกรรมของมอเตอร์ในสภาวะที่ใกล้จุดวิกฤต

## อ้างอิง

1. <https://www.yilmazthailand.com/dcmotors>
2. <https://th.vsdmotor.com/info/brushed-dc-motor-basics-102180834.html>
3. [https://www.pm.co.th/%E0%B8%A3%E0%B8%B2%E0%B8%A2%E0%B8%A5%E0%B8%B0%E0%B9%80%E0%B8%AD%E0%B8%B5%E0%B8%A2%E0%B8%94/Increment\\_Und\\_Encoder\\_Und\\_%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B1%E0%B8%9A\\_Und\\_Absolute\\_Und\\_Encoder\\_Und\\_%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3](https://www.pm.co.th/%E0%B8%A3%E0%B8%B2%E0%B8%A2%E0%B8%A5%E0%B8%B0%E0%B9%80%E0%B8%AD%E0%B8%B5%E0%B8%A2%E0%B8%94/Increment_Und_Encoder_Und_%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B1%E0%B8%9A_Und_Absolute_Und_Encoder_Und_%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3)
4. <http://jwtech.co.th/activity/?p=1662>
5. <https://medium.com/icreativesystems/to-know-pwm-373efa1322ce>

## Experiment 2: การทดลอง: H-Bridge Drive Mode ศึกษาหลักการทำงานและการควบคุม H-Bridge Drive Mode

### จุดประสงค์

- ศึกษาหลักการทำงานของ H-Bridge Drive Mode ทั้ง 3 โหมด (Sign-Magnitude, Locked Anti-Phase, Async Sign-Magnitude)
- ทดสอบการควบคุม DC Motor ด้วย Sign-Magnitude และ Locked Anti-Phase
- วิเคราะห์ข้อดีข้อเสียของแต่ละ Mode เช่นการใช้พลังงาน การควบคุมทิศทาง ฯลฯ

### สมมติฐาน

การควบคุมแบบ Sign-Magnitude Mode คือการสั่ง PWM เพื่อควบคุม ความเร็ว ของมอเตอร์เท่านั้น โดย เมื่อ Duty Cycle ของ PWM เพิ่มขึ้น ความเร็วรอบ DC Motor จะเพิ่มขึ้น ส่วนการควบคุมแบบ Locked Anti-Phase Mode จะใช้ PWM ในการควบคุมทั้ง ความเร็วและทิศทาง

### ตัวแปร

- ตัวแปรต้น : โหมดการควบคุมของ H-Bridge, Duty Cycle ของ PWM (0% ถึง 100%)  
ตัวแปรตาม : ความเร็วรอบของมอเตอร์ (RPM), ทิศทางการหมุนของมอเตอร์  
ตัวแปรควบคุม : ค่าแรงดันไฟฟ้าที่ป้อนให้ DC Motor, ชนิดของมอเตอร์ (DC Brushed Motor ที่ใช้ทดสอบ), สภาพแวดล้อมในการทดลอง, sensor ที่ใช้วัดค่าต่างๆ

### นิยามศัพท์เฉพาะ

#### 1. H-Bridge

วงจรที่ประกอบด้วยสวิตซ์ (MOSFET หรือ Transistor) จำนวน 4 ตัว จัดเรียงในรูปตัวอักษร "H" ใช้สำหรับควบคุม การจ่ายแรงดันไฟฟ้าให้กับมอเตอร์เพื่อควบคุมทั้ง ความเร็วและทิศทางการหมุน

#### 2. Sign-Magnitude Mode

วิธีการควบคุมมอเตอร์ที่ใช้สัญญาณ PWM เพื่อกำหนด ความเร็ว ของมอเตอร์ และใช้สัญญาณ DIR (Direction) ในการกำหนด ทิศทางการหมุน

#### 3. Locked Anti-Phase Mode

วิธีการควบคุมมอเตอร์ที่ใช้สัญญาณ PWM เดียวในการกำหนดทั้ง ความเร็ว และ ทิศทาง โดย PWM ที่ Duty Cycle 50% หมายถึงมอเตอร์หยุดหมุน ส่วน Duty Cycle >50% หรือ <50% จะกำหนดทิศทางและความเร็ว

#### 4. PWM (Pulse Width Modulation)

สัญญาณดิจิทัลที่มีการปรับช่วงเวลาเปิดและปิดของพลังไฟฟ้าเพื่อควบคุมแรงดันเฉลี่ยที่จ่ายให้กับมอเตอร์ ส่งผลต่อความเร็วการหมุน

#### 5. Duty Cycle

อัตราส่วนระหว่างช่วงเวลาที่สัญญาณ PWM อยู่ในสถานะ "เปิด" (High) ต่อหนึ่งรอบค้างทั้งหมดของสัญญาณ โดยแสดงเป็นเปอร์เซ็นต์ เช่น Duty Cycle 70% หมายถึงสัญญาณเปิด 70% และปิด 30%

#### 6. Direction (DIR)

สัญญาณดิจิทัลที่มีค่า HIGH หรือ LOW ใช้เพื่อกำหนดทิศทางการหมุนของมอเตอร์ใน Sign-Magnitude Mode

#### 7. DC Motor

มอเตอร์ไฟฟ้ากระแสตรงที่ใช้ในการทดลอง โดยแรงดันไฟฟ้าที่ป้อนเข้าจะควบคุมความเร็วและทิศทางการหมุน

#### 8. ความเร็วรอบมอเตอร์ (RPM)

จำนวนรอบที่มอเตอร์หมุนต่อหนึ่งนาที วัดจากความถี่ของสัญญาณ PWM ที่จ่ายให้

#### 9. การตอบสนองของมอเตอร์ (Motor Response)

พฤติกรรมของมอเตอร์เมื่อสัญญาณ PWM หรือ DIR เปลี่ยน เช่น การเปลี่ยนความเร็วหรือทิศทางการหมุน

#### 10. การใช้พลังงาน (Power Consumption)

ปริมาณพลังงานที่มอเตอร์ใช้ในการทำงานในแต่ละโหมด วัดโดยการวิเคราะห์กระแสและแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์

### นิยามเชิงปฏิบัติการ

1. DC Motor คือ Nidec Components Geared DC Geared Motor, 12 V dc, 20 Ncm, 70 rpm, 6mm Shaft Diameter
2. Driver Motor คือ Cytron MDD20A Motor Driver
3. Duty Cycle คือ สัดส่วนเวลาที่สัญญาณ PWM อยู่ในสถานะ High เทียบกับรอบเวลารวมทั้งหมด

## เอกสารและงานวิจัยที่เกี่ยวข้อง

### 1. Sign-Magnitude Mode

เป็นรูปแบบหนึ่งของการแทนค่าเลขจำนวนเต็มในระบบดิจิทัล ที่ใช้สัญลักษณ์ในการระบุเครื่องหมาย (บวก หรือลบ) ของจำนวนเต็ม โดยแยกส่วน สัญญาณ (sign) และ ขนาด (magnitude) ของจำนวน

หลักการของ Sign-Magnitude Mode:

#### 1) ส่วนของสัญญาณ (Sign bit):

- 1 บิตแรก ใช้สำหรับระบุเครื่องหมายของจำนวน
- หากบิตสัญญาณเป็น 0 หมายถึงจำนวนเป็นบวก
- หากบิตสัญญาณเป็น 1 หมายถึงจำนวนเป็นลบ

#### 2) ส่วนของขนาด (Magnitude):

- บิตที่เหลือจะใช้สำหรับแทนค่า ขนาดของจำนวน โดยไม่คำนึงถึงเครื่องหมาย (เช่น ค่าเต็มที่เป็นบวกของจำนวน)

### 2. Locked Anti-Phase Mode

เป็นโหมดการทำงานของมอเตอร์ไฟฟ้านิดต่าง ๆ เช่น Brushed DC Motor หรือ Brushless DC Motor (BLDC) ซึ่งมักจะใช้ในระบบที่มีการควบคุมการหมุนในทิศทางตรงข้ามกันระหว่างสองมอเตอร์หรือสองขดลวด โดยในโหมดนี้ การหมุนของมอเตอร์จะเกิดขึ้นในทิศทางตรงข้ามกัน (anti-phase) แต่มีการควบคุมให้การหมุนของมอเตอร์ทั้งสองนั้นถูกซิงโครไนซ์ให้สอดคล้องกันในลักษณะที่เรียกว่า ล็อก (locked).

ลักษณะของ Locked Anti-Phase Mode

#### 1) การทำงานในทิศทางตรงข้าม (Anti-Phase):

- ในโหมดนี้ การหมุนของมอเตอร์สองตัวจะเกิดในทิศทางที่ตรงข้ามกัน (counter-rotating). ตัวอย่างเช่น, ถ้ามอเตอร์ตัวหนึ่งหมุนตามเข็มนาฬิกา, อีกตัวหนึ่งจะหมุนตามเข็มนาฬิกา.
- การหมุนในทิศทางตรงข้ามนี้จะถูกซิงโครไนซ์เพื่อให้มอเตอร์ทำงานร่วมกันได้อย่างมีประสิทธิภาพ และสอดคล้องกับการควบคุม.

#### 2) การควบคุมแบบล็อก (Locked):

- การล็อกหมายถึงการทำให้มอเตอร์ทั้งสองทำงานในรูปแบบที่สอดคล้องกัน โดยใช้การควบคุมที่ทำให้การหมุนของมอเตอร์ทั้งสองถูกปรับให้มีความสัมพันธ์ที่แน่นหนา
- ลักษณะนี้มักจะพบในระบบที่มีการใช้มอเตอร์หลายตัว เช่น ในระบบการขับเคลื่อนที่ต้องการให้มอเตอร์ทั้งสองทำงานร่วมกันอย่างมีประสิทธิภาพ เช่น ในระบบพวงมาลัยไฟฟ้า (Electric Steering Systems) หรือการควบคุมการหมุนของล้อในระบบการขับเคลื่อนของยานพาหนะ

## การใช้งานในระบบ

- การควบคุมมอเตอร์ในทิศทางที่ตรงข้ามกัน: โหมดนี้มักใช้ในระบบที่ต้องการให้มอเตอร์หมุนในทิศทางตรงข้าม เช่น ในมอเตอร์ที่ใช้ในหุ่นยนต์ หรือระบบที่ต้องการการเคลื่อนที่แบบคู่ขนานหรือเคลื่อนที่ด้วยความเร็วต่างกัน
- การใช้งานในระบบขับเคลื่อน (Drive Systems): ในการขับเคลื่อนที่ซับซ้อนที่ต้องการการหมุนทิศทางตรงข้าม เช่น ระบบการขับเคลื่อนของยานพาหนะที่ใช้มอเตอร์สองตัวที่ซิงโครไนซ์กัน

## วิธีดำเนินการทดลอง

ทดสอบการควบคุม DC Motor ด้วย Sign-Magnitude และ Locked Anti-Phase และใช้การปรับ Duty Cycle โดยเปลี่ยน Frequency ของ PWM จากค่าต่ำไปหาค่าสูง เพื่อทำการวัดความเร็วรอบ (RPM) ของมอเตอร์ที่มีเงื่อนไข No Load โดยสามารถทำตาม ขั้นตอนการทดลองในการทดลองที่ 1 ข้อที่ 1-8 เพื่อ เชื่อมต่อบอร์ด Nucleo STM32G474RE เข้ากับคอมพิวเตอร์ เชื่อมต่อสายไฟต่างๆของ Sensor และ DC Motor และสั่งงานผ่านโปรแกรม Simulink

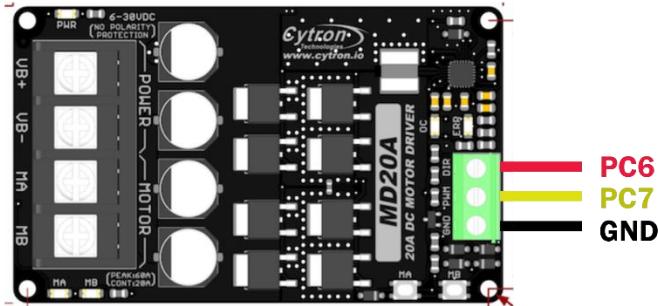
## อุปกรณ์

1. Nidec Components Geared DC Geared Motor, 12 V dc, 20 Ncm, 70 rpm, 6mm Shaft Diameter จำนวน 1 อัน
2. Incremental Encoder AMT103-V จำนวน 1 อัน 3. Warner Electric Magnetic Particle Clutches MPB12 จำนวน 1 อัน
3. WCS1700 Hall Current Sensor จำนวน 1 อัน
4. Cytron MDD20A Motor Driver จำนวน 1 อัน
5. Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
6. MotorXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ DC Motor

## ขั้นตอนการทดลอง

1. สั่งงาน Motor Driver ใน โหมด Sign-Magnitude Mode โดยการต่อสายดังนี้

### Sign-Magnitude Mode



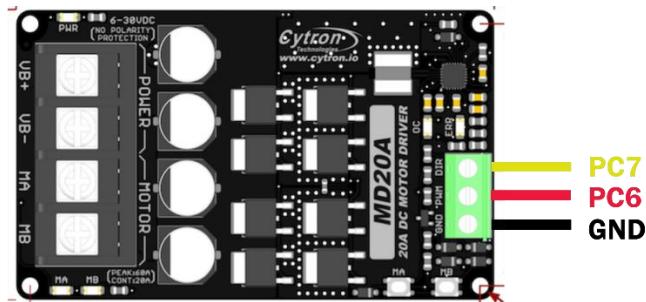
1. ทำการสั่งค่า PWM ในโปรแกรม MotorXploer.slx โดยใช้บล็อก Ramp เพื่อเพิ่มค่า PWM ไปเรื่อยๆ



ทำการเก็บค่าความเร็ว Speed และ กระแส Current จากการ Log Signal สัญญาณที่ออกมาจาก Hall Current Sensor และ Encoder

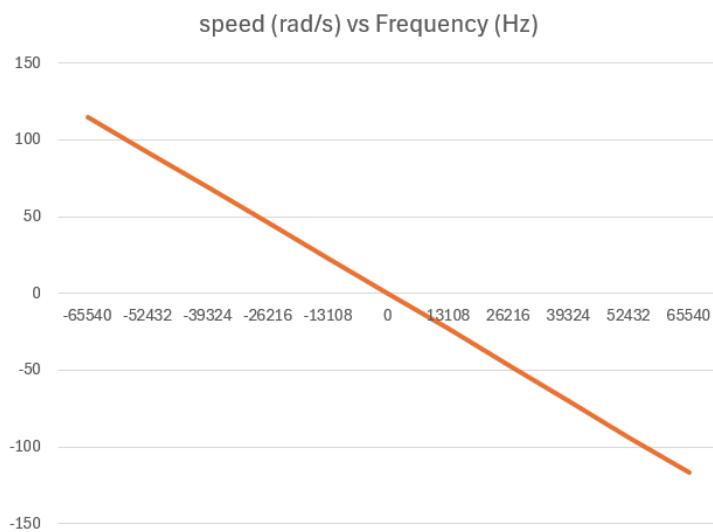
2. สั่งงาน Motor Driver ใน โหมด Locked Anti-Phase Mode โดยการต่อสายดังนี้

### Locked Anti-Phase Mode

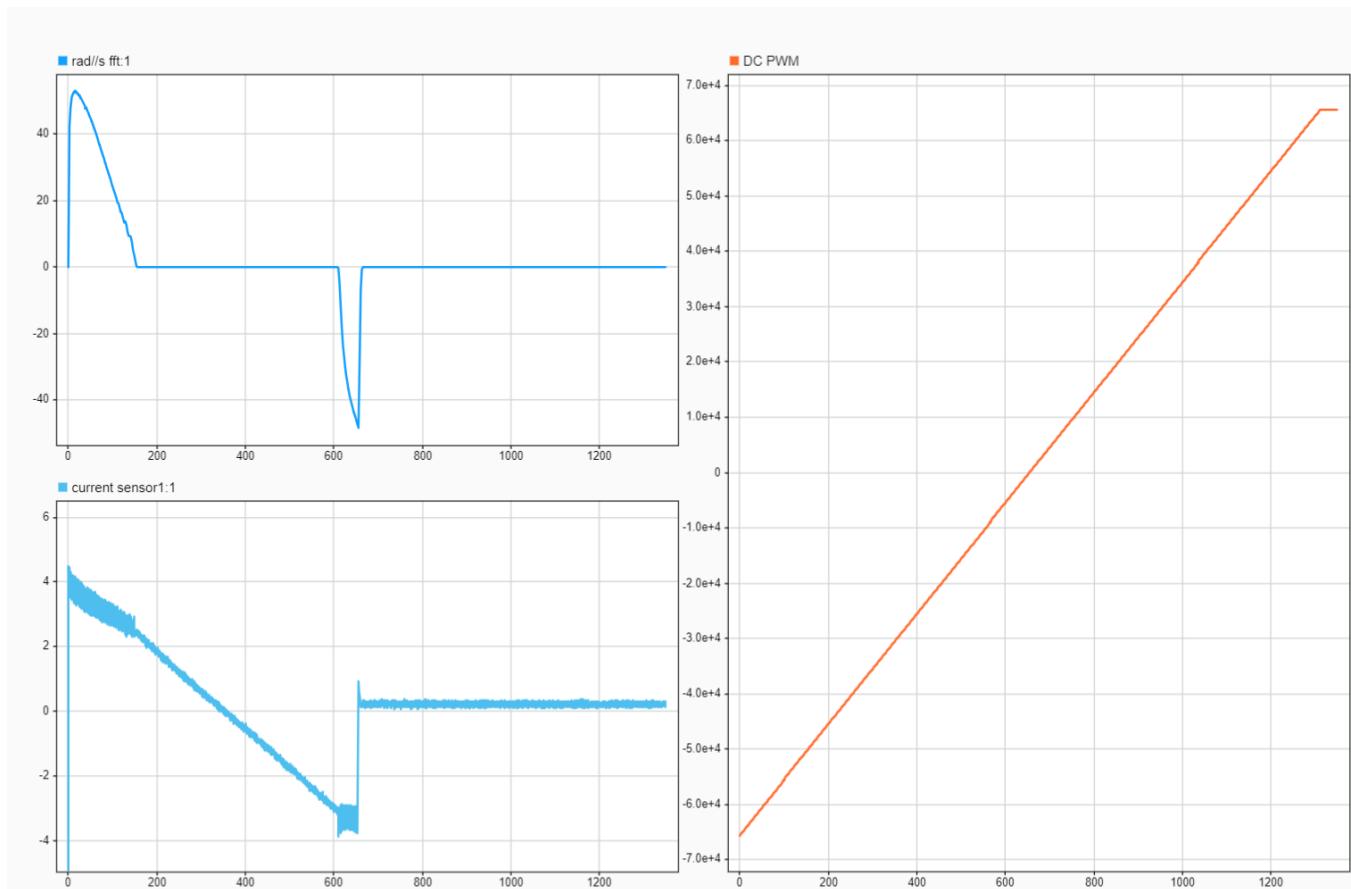


3. และทำการสั่ง PWM เหมือนในข้อที่ 2. และเก็บค่าเหมือนในข้อที่ 3.

## ผลการทดลอง



รูปแสดงกราฟ ความเร็วรอบ ในโหมด Locked Anti-Phase Mode



รูปแสดง ความเร็วรอบ และกระแสที่ใช้ ในโหมด Locked Anti-Phase Mode

## สรุปผลการทดลอง

### 1. การควบคุมมอเตอร์ด้วย Sign-Magnitude Mode

- สามารถควบคุมทิศทางและความเร็วได้อย่างมีประสิทธิภาพ เนื่องจากใช้สัญญาณ PWM สำหรับความเร็ว และสัญญาณ DIR หรือการให้ความถี่ PWM ติดลบ สำหรับการเปลี่ยนทิศทาง
- การใช้พลังงานในโหมดนี้จะน้อยกว่า Locked Anti-Phase เนื่องจากสวิตซ์บางตัวใน H-Bridge อยู่ในสถานะปิดขณะทำงาน

### 2. การควบคุมมอเตอร์ด้วย Locked Anti-Phase Mode

- มีการตอบสนองที่รวดเร็วต่อการเปลี่ยนแปลงทิศทาง เพราะใช้ PWM อย่างเดียวในการควบคุมทั้งทิศทาง และความเร็ว
- การใช้พลังงานในโหมดนี้จะมากกว่า Sign-Magnitude เนื่องจากมอเตอร์ทำงานในลักษณะ "active" ทั้งสองทิศทาง

### 3. Async Sign-Magnitude Mode

- โหมดนี้จะช่วยลดการสูญเสียพลังงานเมื่อเปรียบเทียบกับ Sign-Magnitude ปกติ โดยการปรับการทำงานของสวิตซ์เพื่อเพิ่มประสิทธิภาพการใช้พลังงาน
- โหมดนี้อาจมีการตอบสนองช้ากว่า Locked Anti-Phase ใน การเปลี่ยนทิศทาง

### 4. การควบคุม DC Motor ด้วยโหมดที่แตกต่างกัน (Sign-Magnitude vs Locked Anti-Phase)

- Sign-Magnitude Mode จะเหมาะสมกับการใช้งานที่ต้องการการประยัดพลังงาน เช่น การควบคุมความเร็วคงที่
- Locked Anti-Phase Mode จะเหมาะสมกับการใช้งานที่ต้องการการตอบสนองเร็วและการเปลี่ยนทิศทางบ่อยครั้ง

## 5. ข้อดีข้อเสียของแต่ละ Mode

- Sign-Magnitude: ประหยัดพลังงาน แต่การเปลี่ยนทิศทางอาจช้ากว่า
- Locked Anti-Phase: ตอบสนองเร็ว แต่ใช้พลังงานมากกว่า
- Async Sign-Magnitude: ประสิทธิภาพการใช้พลังงานสูงสุด แต่ซับซ้อนกว่าในการออกแบบจริง

### อภิปรายผล

จากการทดลองและการเปรียบเทียบ พบว่าโหมดที่เหมาะสมจะขึ้นอยู่กับลักษณะการใช้งาน หากต้องการลดการใช้พลังงาน ควรเลือก Sign-Magnitude Mode หรือ Async Sign-Magnitude Mode หากต้องการการตอบสนองที่รวดเร็ว ควรเลือก Locked Anti-Phase Mode

### ข้อเสนอแนะ

ควรเพิ่มการทดลองภายใต้เงื่อนไขทดสอบที่หลากหลายเพื่อยืนยันการเลือกโหมดที่เหมาะสมในสถานการณ์จริง  
อ้างอิง

1. <https://th.cytron.io/p-20amp-6v-30v-dc-motor-driver?srsltid=AfmBOoqASBhQ2ZhsMY8iefGBwz5PQbocqs6K5iPbZ9blogtFllL9d7R3>

### Experiment 3: การทดลองเพื่อวิเคราะห์ความถี่ที่ Stepper Motor เกิด Loss Step ในแต่ละโหมดการควบคุม (Full, Half, Micro-Stepping)

#### จุดประสงค์

- ศึกษาหลักการทำงานของ Stepper Motor และหาความสัมพันธ์ระหว่าง Speed กับ Frequency ของสัญญาณที่จ่ายเข้า Stepper Motor
- ศึกษาความสัมพันธ์ระหว่างความถี่ของสัญญาณที่จ่ายให้ Stepper Motor กับการเกิด Loss Step และเปรียบเทียบกับ Drive Modes ในโหมดการควบคุมที่แตกต่างกัน ได้แก่ Full-Step, Half-Step, และ Micro-Step
- ศึกษาผลกระทบของ Ramp Acceleration ต่อ Stepper Motor ในการเปลี่ยนแปลงความเร็วและการเกิด Loss Step

#### สมมติฐาน

ความเร็ว (Speed) ของ Stepper Motor มีความสัมพันธ์เชิงเส้นกับความถี่ของสัญญาณ (Frequency) ที่จ่ายเข้า Stepper Motor โดยตรง กล่าวคือ เมื่อความถี่เพิ่มขึ้น ความเร็วของ Stepper Motor จะเพิ่มขึ้นในลักษณะเชิงเส้นภายในช่วงความถี่ที่ไม่มีการ Loss Step

โหมดการควบคุมแบบ Full-Step Drive จะเกิด Loss Step ที่ความถี่ต่ำกว่าโหมดการควบคุมแบบ Half-Step Drive และ Micro-Step Drive เนื่องจากความละเอียดของการเคลื่อนที่ในแต่ละ Step ต่ำกว่า ส่งผลให้ Stepper Motor สูญเสียการควบคุมสามารถแม่เหล็กจนไม่สามารถตอบสนองต่อความถี่ที่สูงขึ้นได้อย่างมีประสิทธิภาพ

Ramp Acceleration ที่มีค่าการเร่ง (Acceleration) สูงจะเพิ่มโอกาสเกิด Loss Step มาขึ้น เนื่องจาก Stepper Motor อาจไม่สามารถตอบสนองต่อการเปลี่ยนแปลงของความเร็วได้ทัน ส่วน Ramp Acceleration ที่มีค่าการเร่งต่ำจะช่วยลดการเกิด Loss Step และเพิ่มความสามารถในการเปลี่ยนแปลงความเร็วของ Stepper Motor

#### ตัวแปร

ตัวแปรต้น : โหมดการควบคุม (Drive mode), ความถี่ของสัญญาณ (Frequency) ที่จ่ายเข้า Stepper Motor (Hz), Ramp Acceleration ระดับการเปลี่ยนแปลงความถี่ที่เพิ่มขึ้นในแต่ละช่วงเวลา

ตัวแปรตาม : ความเร็ว (Speed) ของ Stepper Motor (RPM), ความถี่สูงสุด (Maximum Frequency) ก่อนเกิด Loss Step ในแต่ละโหมดการควบคุม

ตัวแปรควบคุม : กระแสไฟฟ้าที่จ่ายให้มอเตอร์ (A), แรงดันไฟฟ้าที่จ่ายให้มอเตอร์ (V), แรงบิด荷载 (Nm)

#### นิยามศัพท์เฉพาะ

1. Stepper Motor

มอเตอร์ไฟฟ้าประเภทหนึ่งที่หมุนได้เป็นขั้นๆ (Steps) โดยตำแหน่งของมอเตอร์ถูกกำหนดโดยจำนวนและลำดับของสัญญาณควบคุม ทำให้สามารถควบคุมความเร็วและตำแหน่งได้อย่างแม่นยำ

## 2. Loss Step

การที่ Stepper Motor สูญเสียตำแหน่งที่ตั้งไว้ (Target Position) เนื่องจากไม่สามารถติดตามลำดับสัญญาณควบคุมได้ทัน โดยเกิดจากข้อจำกัดของแรงบิด ความถี่ หรือแรงเฉื่อย

## 3. Stepping Mode

รูปแบบการควบคุม Stepper Motor โดยการส่งสัญญาณไฟฟ้าไปยังขดลวดของมอเตอร์ในลักษณะต่างๆ เพื่อควบคุมการเคลื่อนที่ของโรเตอร์ (Rotor) ให้เกิดการหมุนในมุมและตำแหน่งที่ต้องการ

## 4. Full-Step Drive

โหมดการควบคุม Stepper Motor ที่ขับเคลื่อนมอเตอร์ให้หมุนหนึ่งขั้น (Step) ต่อการจ่ายกระแสไฟฟ้าผ่านขดลวดในแต่ละเฟส ซึ่งให้การเคลื่อนที่ 1.8° ต่อสัญญาณควบคุมหนึ่งครั้ง

## 5. Half-Step Drive

โหมดการควบคุม Stepper Motor ที่ขับเคลื่อนมอเตอร์ให้หมุนครึ่งขั้น (Half Step) โดยการจ่ายกระแสไฟฟ้าสลับระหว่างขดลวดหนึ่งเฟสและสองเฟสพร้อมกัน ทำให้มอเตอร์เคลื่อนที่ครั้งละ 0.9° ต่อสัญญาณควบคุม

## 6. Micro-Step Drive

โหมดการควบคุม Stepper Motor ที่แบ่งการหมุนในแต่ละขั้นย่อยลงไปอีกหลายส่วน (เช่น 1/4, 1/8, 1/16, 1/32) โดยการปรับค่ากระแสไฟฟ้าที่จ่ายให้กับขดลวดในแต่ละเฟสอย่างแม่นยำ

## 7. Acceleration Profile

รูปแบบของการเปลี่ยนแปลงความเร็วของมอเตอร์ตามเวลา ซึ่งกำหนดการเพิ่ม/ลดความเร็วเพื่อให้เหมาะสมกับความสามารถของมอเตอร์ มักแสดงในรูปกราฟอัตราเร่งเชิงมุม ( $\alpha = \frac{d\omega}{dt}$ ) ต่อเวลา หรือเป็นฟังก์ชันใน MATLAB

## 8. Frequency

จำนวนครั้งที่สัญญาณควบคุมเปลี่ยนแปลงในหนึ่งวินาที มีหน่วยเป็นเฮิรตซ์ (Hz) ซึ่งมีผลโดยตรงต่อความเร็วเชิงมุมของ Stepper Motor

## 9. Incremental Encoder

อุปกรณ์ตรวจวัดตำแหน่งเชิงมุมที่ให้สัญญาณดิจิทัลเป็นจำนวน Pulse เพื่อบอกตำแหน่งและทิศทางการหมุนของมอเตอร์

## 10. Hall Current Sensor

เซ็นเซอร์วัดกระแสไฟฟ้าบนขดลวดของ Stepper Motor โดยอาศัยหลักการ Hall Effect เพื่อวัดและแปลงสัญญาณกระแสเป็นแรงดันไฟฟ้า

## 11. Signal Conditioning

กระบวนการปรับแต่งสัญญาณจากเซ็นเซอร์ให้เหมาะสมกับการประมวลผล เช่น การกรองสัญญาณรบกวน หรือการปรับสัญญาณให้อยู่ในช่วงที่ต้องการ

## 12. Signal Processing

การวิเคราะห์และแปลงสัญญาณที่ได้จากเซ็นเซอร์เพื่อให้ได้ข้อมูลที่สามารถนำไปใช้งานหรือวิเคราะห์ได้

## 13. SI Derived Units

หน่วยที่ได้จากการแปลงหน่วยมาตรฐาน เช่น ความเร็วเชิงมุม ( $\text{rad/s}$ ) หรือกระแสไฟฟ้า ( $\text{A}$ ) เพื่อการวิเคราะห์ผลการทดลอง

## 14. No Load

สภาพที่ Stepper Motor ทำงานโดยไม่มีการต่อต้านจากแรงบิดภายนอก (Load) ทำให้แรงบิดทั้งหมดที่มอเตอร์ผลิตขึ้นใช้ในการหมุนตัวเอง

## 15. Angular Velocity

ความเร็วเชิงมุมที่แสดงถึงอัตราการหมุนของมอเตอร์ มีหน่วยเป็นเรเดียนต่อวินาที ( $\text{rad/s}$ ) หรือรอบต่อนาที (RPM)

## 16. Angular Acceleration

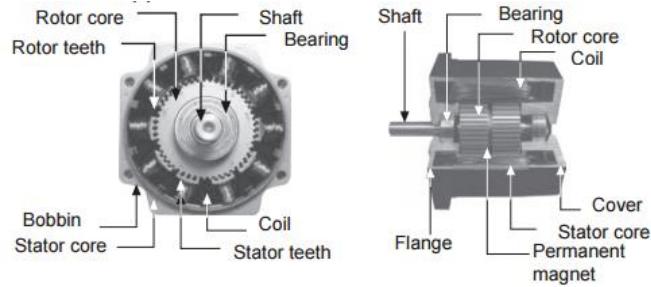
อัตราการเปลี่ยนแปลงของความเร็วเชิงมุมต่อหน่วยเวลา มีหน่วยเป็นเรเดียนต่อวินาทียกกำลังสอง ( $\text{rad/s}^2$ )  
นิยามเชิงปฏิบัติการ

1. Stepper Motor คือ HANPOSE - Stepper Motor (Nema11) แรงบิด 10 N.cm 1.0A (11HS3410)
2. บอร์ดควบคุม คือ Nucleo STM32G474RE
3. Motor driver คือ DRV8825

## เอกสารและงานวิจัยที่เกี่ยวข้อง

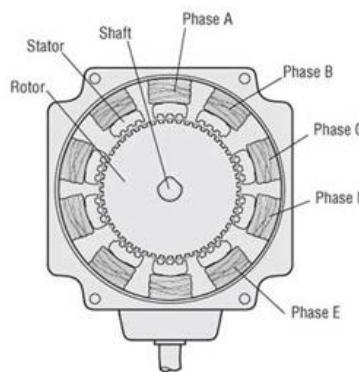
### 1. หลักการทำงานของ Stepper motor

Stepper Motor จะแปลงพลังงานไฟฟ้าให้เป็นการหมุน การทำงานของมันคือการหมุนทีละขั้น (step) ครั้งที่มีกระแสไฟฟ้าถูกจ่ายเข้าไป การทำงานของมอเตอร์จะถูกควบคุมโดยไดรเวอร์ที่ส่งสัญญาณพัลส์เข้าไปในมอเตอร์ ทำให้บันทุนในแต่ละขั้น จำนวนพัลส์ที่ส่งเข้าไปจะตรงกับจำนวนขั้นที่มอเตอร์หมุน และอัตราการหมุนของมอเตอร์จะเท่ากับความถี่ของพัลส์ที่ป้อนเข้าไป โดยโครงสร้างภายในนั้นจะประกอบไปด้วยขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำงานจากกระแสไฟฟ้า จะมีจุดอ่อนคือการหมุนไม่ต่อเนื่อง โดยแต่ละขั้นที่หมุนจะมีจุดลาก (คอยล์) พ้นอยู่ เมื่อมีกระแสผ่านคอยล์จะเกิดสนามแม่เหล็กไฟฟ้าขึ้น



Stepper Motor มี 3 ชนิดดังนี้

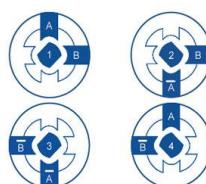
- Permanent Magnet Stepper Motor (PM): มีแม่เหล็กถาวรเป็นโรเตอร์ แต่ไม่มีฟัน มักมีมุ่ง step ที่ไม่ละเอียดนัก แต่มีแรงดึงดูดแน่น (Detent Torque)
- Variable Reluctance Stepper Motor (VR): มีฟันบนโรเตอร์และสเตเตอร์ แต่ไม่มีแม่เหล็ก ดังนั้นจึงไม่มีแรงดึงดูดแน่น
- Hybrid Stepper Motor: รวมคุณสมบัติของแม่เหล็กจากมอเตอร์ PM และฟันจากมอเตอร์ VR เข้าด้วยกัน โดยแม่เหล็กถูกทำให้มีขั้วเหนือ (North Pole) และใต้ (South Pole) ในแนวแกน โรเตอร์มีถ่วงโรเตอร์ที่มีฟัน 2 ส่วน โดยแต่ละส่วนมีฟัน 50 ชี และถูกเลื่อนตำแหน่งกัน  $3.6^\circ$  ตั้งนั้น หากมองผ่านฟันบนถ่วงขั้วเหนือ จะเห็นฟันบนถ่วงขั้วใต้ตรงกลางระหว่างฟันของถ่วงขั้วเหนือ



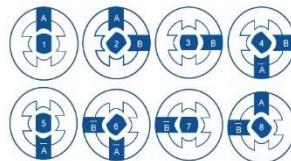
Motor Structural Diagram: Cross-Section Perpendicular to Shaft

โหมดการทำงานของ Stepper Motor แบ่งเป็น 3 แบบ

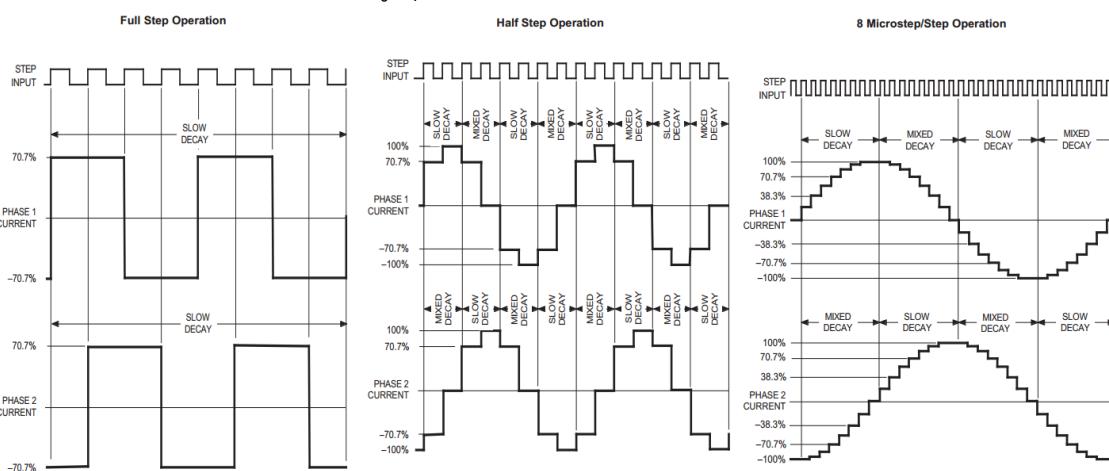
- Full-Step Mode: หมุนทีละหนึ่งสเต็ปเต็ม หมายความว่ารับงานที่ไม่ต้องการความละเอียดสูง



- Half-Step Mode: สลับการเปิดเฟสเดียวและสองเฟส ทำให้มุมก้าวลดลงครึ่งหนึ่ง ( $0.9^\circ$ ) ลดการสั่นสะเทือน โดยมีลำดับการทำงานทางไฟฟ้า 8 ขั้นตอน ให้การเคลื่อนที่ละเอียดมากขึ้น

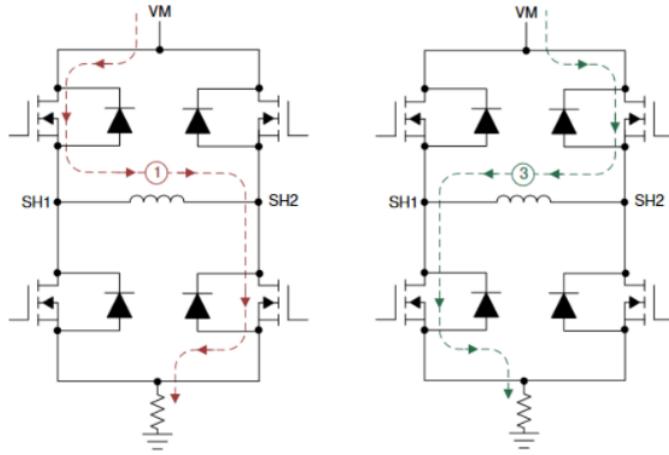


- Micro-step Mode: แบ่งการเคลื่อนที่ออกเป็นส่วนย่อยมากที่สุด เช่น  $1/8$ ,  $1/16$ ,  $1/32$  เหมาะสำหรับงานที่ต้องการความแม่นยำสูงสุด



## 2. การทำงานของ Stepper Motor Driver

Stepper Motor จำเป็นต้องใช้แรงดันไฟฟ้าและกระแสไฟฟ้าที่สูงกว่าความสามารถของคอนโทรลเลอร์ที่ไว้ในตั้งนั้นจึงต้องอาศัย Driver ในการแปลงสัญญาณจากคอนโทรลเลอร์ให้เหมาะสมกับการทำงานของมอเตอร์ Driver มีบทบาทสำคัญในการสร้างสัญญาณควบคุมที่เหมาะสมจากข้อมูลอินพุต เพื่อใช้ขั้บเคลื่อนแกนมอเตอร์ให้ทำงานตามที่กำหนด หนึ่งในวิธีการควบคุมมอเตอร์สเต็ปที่ได้รับความนิยมอย่างแพร่หลาย คือการใช้งาน H-bridge ซึ่งเป็นวงจรที่ประกอบด้วยทรานซิสเตอร์ FET จำนวน 4 ตัว โดยทรานซิสเตอร์แต่ละตัวมีค่าความด้านทานต่ำระหว่างขา Drain และ Source (RDSON) เมื่อยื่นในสถานะทำงาน เพื่อให้สามารถรองรับกระแสไฟฟ้าที่จำเป็นต่อการขับเคลื่อนมอเตอร์ได้ โดยทั่วไป Stepper Motor จะต้องใช้ H-bridge อย่างน้อย 2 ชุด เนื่องจากมอเตอร์มีขดลวดอย่างน้อย 2 ชุด ควบคุมกระแสไฟฟ้าที่ไหลผ่านขดลวดเหล่านี้สามารถทำได้โดยการส่งสัญญาณควบคุมที่เหมาะสมไปยัง Gate ของทรานซิสเตอร์ FET ในวงจร H-bridge การกระตุ้นขดลวดทั้งสองสลับกันตามลำดับที่เหมาะสมจะทำให้แกนมอเตอร์เคลื่อนที่ไปยังตำแหน่งที่ต้องการอย่างมีประสิทธิภาพ



### 3. Loss of Step in Stepper Motors

หมายถึงการที่มอเตอร์สเต็ปเปอร์ไม่สามารถหมุนได้ตามจำนวนก้าวที่คำนวณไว้ ส่งผลให้ตำแหน่งของโรเตอร์ไม่ตรงกับตำแหน่งที่ต้องการ หรือการเคลื่อนที่ไม่เป็นไปตามคำสั่ง ซึ่งอาจเกิดจากหลายปัจจัย

- โหลดมากเกินไป (Overloading): ถ้ามีโหลดที่มากเกินไป มอเตอร์อาจไม่สามารถหมุนได้ตามจำนวนขั้นที่กำหนด การเพิ่มแรงบิดที่จำเป็นเกินไปจะทำให้มอเตอร์ไม่สามารถย้ายไปยังตำแหน่งที่ต้องการได้
- ความเร็วสูงเกินไป (Excessive Speed): การขับมอเตอร์ที่ความเร็วสูงเกินไปจะทำให้มอเตอร์ไม่สามารถตามคำสั่งได้ทัน เนื่องจากมอเตอร์สเต็ปเปอร์มีลักษณะการทำงานแบบเป็นก้าว (Discrete Steps) และอาจไม่สามารถคุ้มจังหวะการหมุนที่เร็วมากได้
- กระแสไฟไม่เพียงพอ (Insufficient Current): หากมีการขับเคลื่อนมอเตอร์ด้วยกระแสไฟที่ไม่เพียงพอ มอเตอร์อาจขาดแรงบิดที่จำเป็นในการหมุนตามจำนวนก้าวที่คำนวณไว้ ส่งผลให้การเคลื่อนที่ไม่ถูกต้อง
- ปัญหาจากระบบควบคุม (Controller Issues): ปัญหาทางด้านฮาร์ดแวร์หรือซอฟต์แวร์ในตัวควบคุมอาจทำให้การควบคุมลำดับการเปิดปิดเฟสไม่ถูกต้อง และทำให้มอเตอร์เสียตำแหน่งได้
- การตั้งค่าไม่เหมาะสม (Incorrect Step Sequence or Microstepping Setting): การตั้งค่าลำดับการเปิดเฟส (Step Sequence) หรือการใช้โหมด Microstepping ที่ไม่เหมาะสมอาจทำให้มอเตอร์ไม่สามารถติดตามการเคลื่อนที่ได้ตามคำสั่ง

### 4. Stepper Motor Systems

- Open Loop Systems: จะมีการส่งพัลส์จากตัวสร้างพัลส์ไปยังวงจรการกำหนดลำดับเฟส (Phase Sequencing Circuit) ซึ่งจะกำหนดว่าเฟสใดต้องเปิดหรือปิด มีข้อเสียคือไม่มีการยืนยันตำแหน่งและไม่สามารถตรวจสอบได้ว่ามอเตอร์ทำการเคลื่อนไหวตามคำสั่งหรือไม่

- Closed Loop Systems: วิธีที่นิยมในการปิดวงจร (Closed Loop) คือการติดตั้งตัวเข้ารหัส (Encoder) ที่แกนหลังของมอเตอร์ ซึ่งจะเป็นajanบางๆ ที่มีเส้นบนมัน เมื่อ詹หมุน เส้นจะผ่านระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ ทำให้เกิดพลังส์ออกแบบสายสัญญาณ พลังส์เหล่านี้จะถูกส่งกลับไปยังตัวควบคุม ซึ่งจะนับจำนวนพลังส์ เมื่อการเคลื่อนที่เสร็จสิ้น ตัวควบคุมจะเปรียบเทียบจำนวนพลังส์ที่ส่งไปยังไดเรเวอร์กับจำนวนพลังส์ที่ได้รับจากตัวเข้ารหัส หากจำนวนพลังส์ไม่ตรงกัน การแก้ไขจะถูกทำที่ตอนท้ายของการเคลื่อนที่ มีข้อดีคือเพิ่มความแม่นยำในการเคลื่อนที่ และสามารถแก้ไขข้อผิดพลาดได้
- Servo Systems: ระบบเซอร์โวใช้การควบคุมตำแหน่งแบบปิดวงจรเช่นเดียวกับระบบ Closed Loop แต่จะมีการใช้มอเตอร์และตัวควบคุมที่มีความสามารถในการปรับแรงบิดและความเร็วอย่างแม่นยำ ระบบนี้จะปรับพลังงานที่ให้กับมอเตอร์ตามโหลดและความต้องการ

## 5. Stepper Motor Specifications (HANPOSE - Stepper Motor (Nema11) แรงบิด 10 N.cm 1.0A (11HS3410))

STEP ANGLE	$1.8^\circ \pm 10\%$	CERTIFICATION	ROHS, CE , FCC
NUMBER OF THE PHASE	2	RATED CURRENT	1.0 A
INSULATION RESISTANCE	100M ohm MIN. (500V DC)	RESISTANCE PER PHASE	4.0Ohm $\pm 10\%$
INSULATION CLASS	CLASS B	INDUCTANCE PER PHASE	4.2 MH $\pm 20\%$
Step angle accuracy	$\pm 5\%$	HOLDING TORQUE	0.1 N.m
WEIGHT	120G	STEPPER MOTOT TYPE	28BYGH34P100

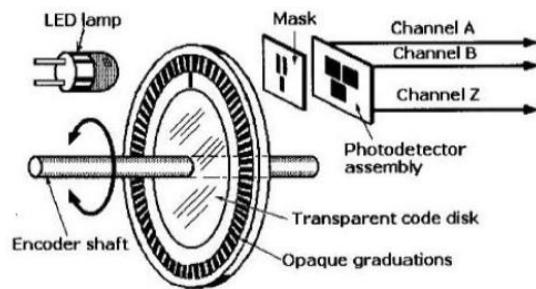
**HANPOSE**

Drawing ownership are "HANPOSE" , Please indicate the source if use.

Type: 28BYGH34P100 . 2-phase hybrid stepper motors

	(°)	(mm)	(A)	(Ohm)	(mH)	(N.m)	(g.cm²)	(NO.)	(g)
11HS28P100	$1.8^\circ$	28	1.0	2.7	2.0	0.08	7	4	90
11HS34P100	$1.8^\circ$	34	1.0	4.0	4.2	0.1	10	4	120
11HS40P100	$1.8^\circ$	40	1.0	4.0	3.8	0.12	12	4	150
11HS50P100	$1.8^\circ$	50	1.0	5.5	5.6	0.17	18	4	200

## 6. หลักการทำงานของ Incremental Encoder



Incremental Encoder เป็นอุปกรณ์เซนเซอร์ที่ใช้สำหรับการวัดการหมุนของเพลา โดยจะทำการแปลงการเคลื่อนที่เชิงมุมของเพลามอเตอร์ให้เป็นสัญญาณ pulse ซึ่งเป็นการแสดงการหมุนของเพลาแบบสัมพัทธ์เมื่อเทียบกับตำแหน่งเริ่มต้น โครงสร้างของ Incremental Encoder ประกอบด้วยajanหมุน (rotary disk) ที่มีการเจาะช่องขนาดเล็กที่ระยะห่างคงที่ตลอดรอบของajan และอุปกรณ์ตรวจจับสัญญาณ ซึ่งมักเป็นเซนเซอร์แสงหรือเซนเซอร์แม่เหล็ก โดยเมื่อเพลาหมุนไปajanหมุนก็จะตัดผ่านเซนเซอร์ตรวจจับทำให้เกิดการสับเปลี่ยนระหว่างสัญญาณเปิด - ปิด (On - Off) ในรูปแบบของ pulse ตามตำแหน่งการหมุนของเพลา โดยจำนวน pulse ที่เกิดขึ้นในแต่ละรอบของการหมุน (Pulses Per Revolution) จะแปรผันตามตำแหน่งเชิงมุมและทิศทางการหมุน

## 7. Data sheet ของ AMT103-V

### FEATURES

- patented capacitive ASIC technology
- low power consumption
- CMOS outputs
- 16 DIP switch selectable resolutions
- index pulse
- modular package design
- straight (radial) and right-angle (axial) versions
- 9 mounting hole options for radial version
- 8 mounting hole options for axial version
- -40-100°C operating temperature



### ELECTRICAL

parameter	conditions/description	min	typ	max	units
power supply	VDD	3.6	5	5.5	V
current consumption	with unloaded output		6		mA
output high level			VDD-0.8		V
output low level				0.4	V
output current	CMOS sink/source per channel			2	mA
rise/fall time			30		ns

### INCREMENTAL CHARACTERISTICS

parameter	conditions/description	min	typ	max	units
channels	quadrature A, B, and X index				
waveform	CMOS voltage square wave				
phase difference	A leads B for CCW rotation (viewed from front)		90		degrees
quadrature resolutions <sup>1</sup>	48, 96, 100, 125, 192, 200, 250, 256, 384, 400, 500, 512, 800, 1000, 1024, 2048				PPR
index <sup>2</sup>	one pulse per 360 degree rotation				
accuracy			0.25		degrees
quadrature duty cycle (at each resolution)	256, 512, 1024, 2048 48, 96, 100, 125, 192, 200, 250, 384, 400, 500 800, 1000	49 47 43	50 50 50	51 53 56	%

Notes:  
1. Resolution selected via adjustable DIP switch, pre-set to 2048 PPR. All resolutions are listed as pre-quadrature, meaning the final number of counts is PPR x 4.  
2. Some stepper motors may leak a magnetic field causing the AMT index pulse to not function properly (non-magnetic version available with 8 pulses per revolution).

### MECHANICAL

parameter	conditions/description	min	typ	max	units
motor shaft length		9			mm
motor shaft tolerance			NOM +0/-0.015		mm
weight	AMT102 AMT103		20.5 14.0		g
axial play				±0.3	mm
rotational speed (at each resolution)	192, 384, 400, 500, 800, 1000, 1024, 2048 48, 96, 100, 125, 200, 250, 256, 512			7500 15000	RPM

### ENVIRONMENTAL

parameter	conditions/description	min	typ	max	units
operating temperature <sup>1</sup>		-40		100	°C
humidity	non-condensing			95	%
vibration	20-500 Hz, 1 hour on each XYZ			10	G
shock	11 ms, ±XYZ direction			50	G
RoHS	yes				

Note: 1. Encoders with operating temperature of -40-125°C are available as a custom order

## วิธีดำเนินการทดลอง

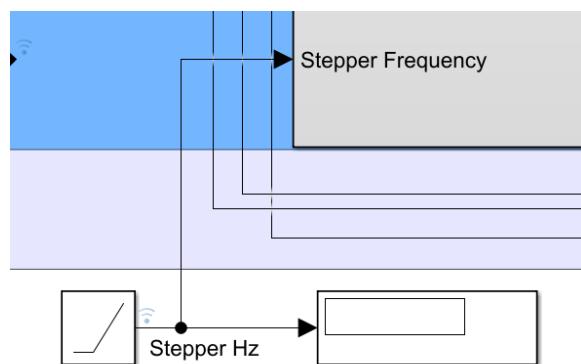
สามารถทำตาม ขั้นตอนการทดลองในการทดลองที่ 1 ข้อที่ 1-8 ดังนี้ เชื่อมต่อบอร์ด Nucleo STM32G474RE เข้ากับคอมพิวเตอร์ เชื่อมต่อสายไฟต่างๆของ Incremental Encoder, DRV 8255, Stepper Motor และส่วนงานผ่านโปรแกรม Simulink โดยการสั่งความถี่ของสัญญาณ (Frequency) ที่จ่ายเข้า Stepper Motor (Hz) เพิ่มขึ้นไปตั้งแต่ความถี่ 0 Hz ขึ้นไปจนถึงจุดที่เกิดการ Loss Step ในทุก Drive Mode ของ Stepper Motor

### อุปกรณ์

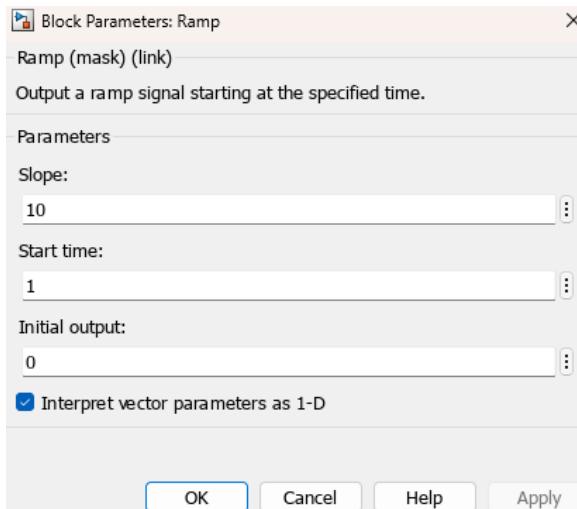
4. RS PRO Hybrid, Permanent Magnet Stepper Motor, 0.22Nm Torque, 2.8 V, 1.8°, 42.3 x 42.3mm Frame, 5mm Shaft จำนวน 1 อัน
5. Incremental Encoder AMT103-V จำนวน 1 อัน
6. Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
7. DRV8825 Stepper Motor Controller IC จำนวน 1 อัน
8. MotorXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ Stepper Motor

### ขั้นตอนการทดลอง

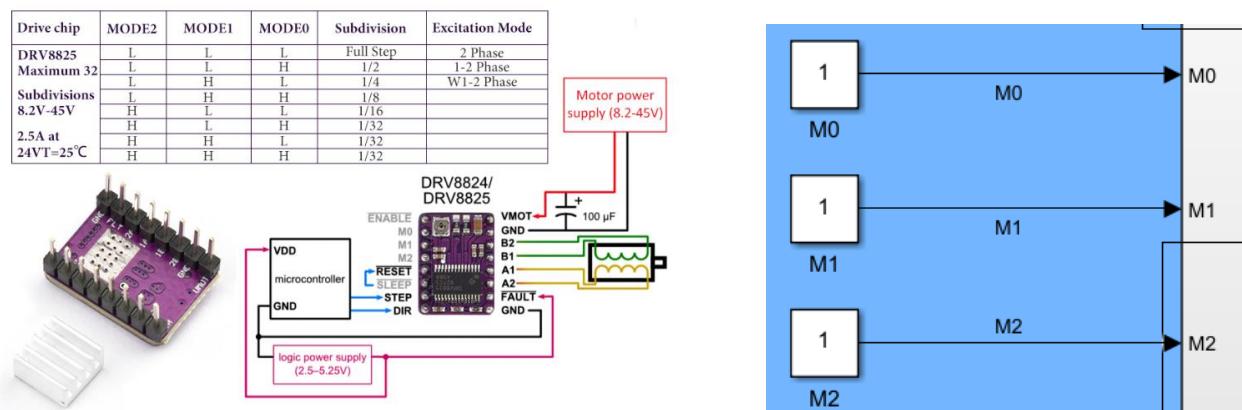
1. เชื่อมต่อ Nucleo STM32G474RE เข้ากับบอร์ด MotorXplorer และทำการจ่ายไฟ
2. ตรวจสอบว่า Stepper Motor เชื่อมต่อกับบอร์ด MotorXplorer อย่างถูกต้อง
3. เข้า Simulink ใน MATLAB เพื่อทำการเปิดไฟล์สำหรับควบคุมบอร์ด MotorXplorer
4. ทำการอัปโหลดไฟล์ bin ลงใน Nucleo STM32G474RE จากนั้นเลือกพอร์ตการส่งข้อมูลให้ถูกต้อง แล้วทำการรับพารามิเตอร์ของ Stepper Frequency ให้เปลี่ยนไปตามเวลาที่กำหนดด้วยบล็อก Ramp ดังรูป



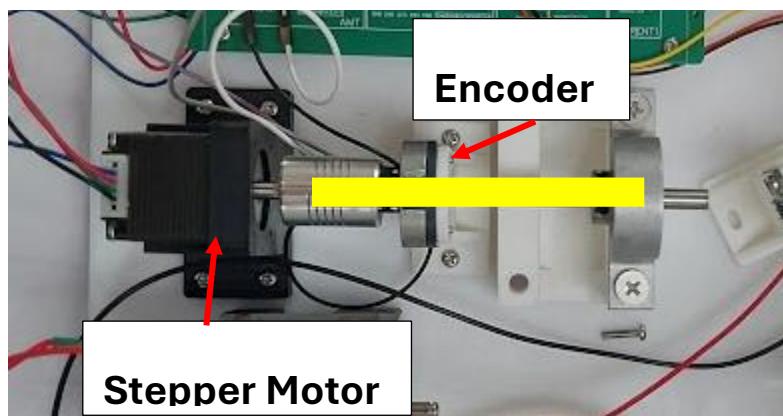
5. ทำการปรับค่าพารามิเตอร์ Slope ภายในล็อก Ramp ให้เปลี่ยน Stepper Frequency 10 Hz ทุก ๆ 1 วินาที



6. ทำการปรับค่า Mode 0, Mode 1 และ Mode 2 ให้สอดคล้องกับ Stepping Mode ที่ต้องการในการวัดผล โดยอ้างอิงวิธีปรับค่าจากรูป

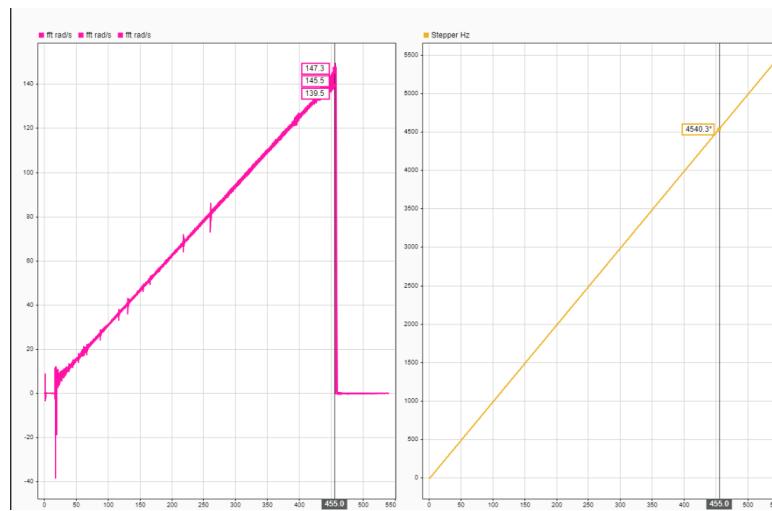


7. ทำการขันยึดตัวหนอนให้แน่นเพียง Coupling ส่วนที่ยึดรหัสระหว่าง Stepper Motor กับ Bearing, Bearing ที่อยู่ระหว่าง Encoder และ Stepper Motor และคลาย Pulley ผ่าน Stepper Motor ออกไป

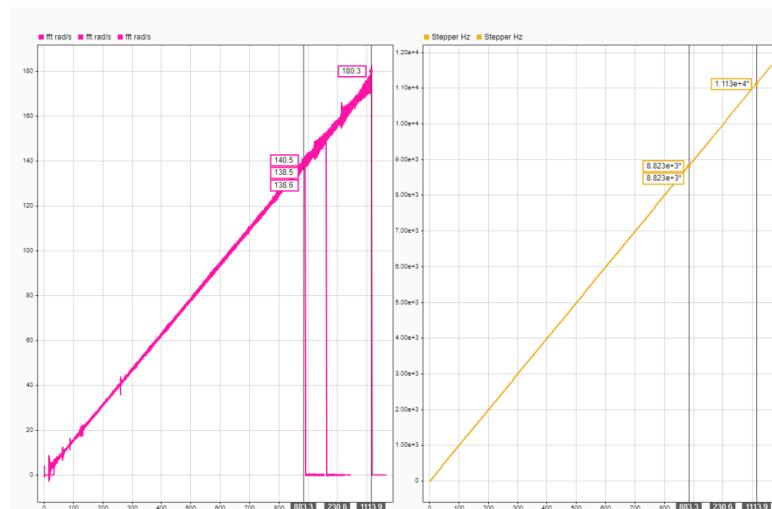


8. ทำการสั่งงานโปรแกรมแล้วเก็บค่าความเร็วที่วัดได้จาก Encoder เทียบกับค่าความถี่การสั่งงานที่ลงทะเบียนด้วยโหมด Step Drive อื่น ๆ

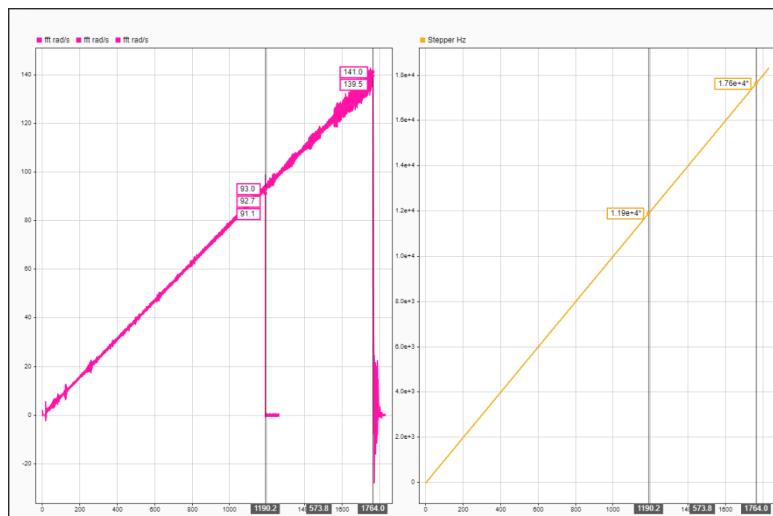
#### ผลการทดลอง



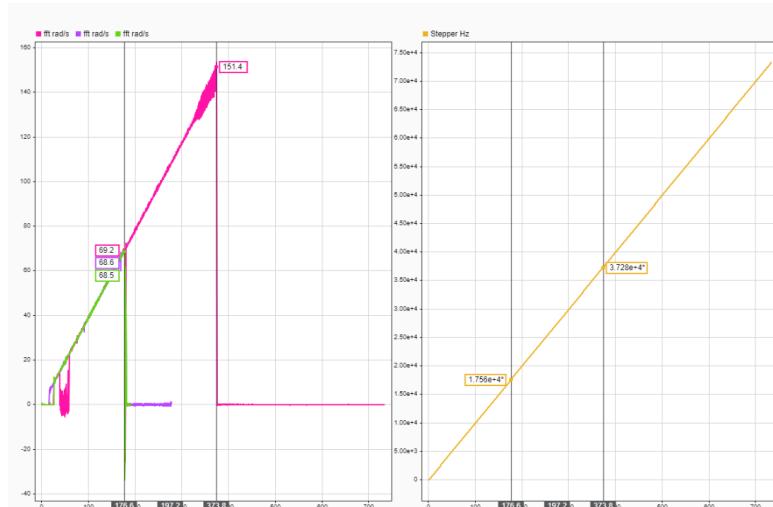
รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่สั่งงานในโหมด Full Step



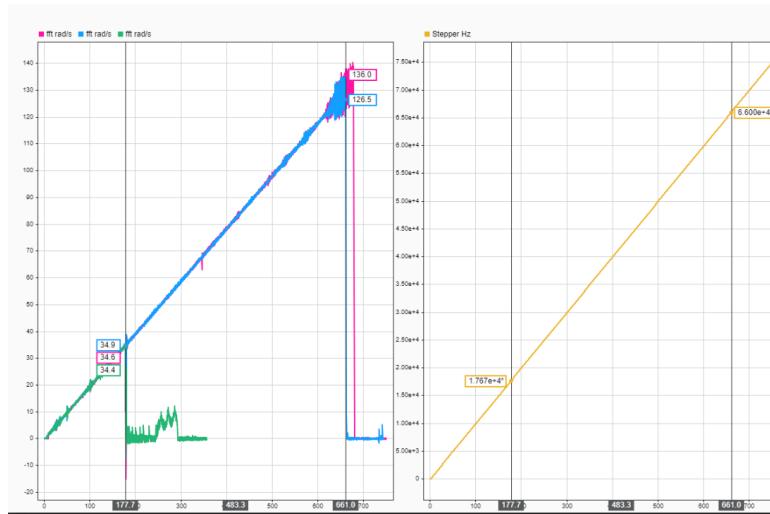
รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่สั่งงานในโหมด Half Step



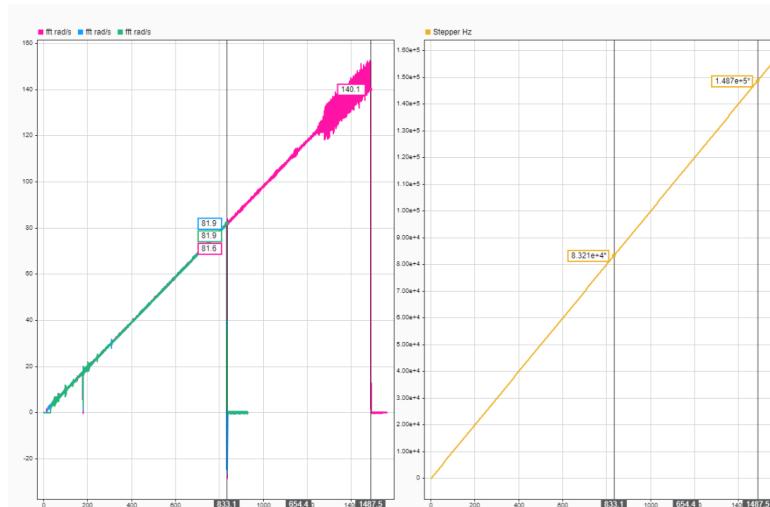
รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่ส่งงานในโหมด 1/4 Micro Step



รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่ส่งงานในโหมด 1/8 Micro Step



รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่สั่งงานในโหมด 1/16 Micro Step



รูปแสดงกราฟความเร็วที่หาได้จาก Encoder เทียบกับความถี่ที่สั่งงานในโหมด 1/32 Micro Step

Drive Mode	Speed (rad/s)	Frequency (Hz)
Full-Step	144.1	4540
Half-Step	139.2	8823
"1/4	140.3	17600
"1/8	68.8	17560
"1/16	34.6	17670
"1/32	81.9	83210

ตารางแสดงความเร็วสุดท้ายเทียบกับความถี่สุดท้ายที่สั่งก่อนเกิดการ Loss Step ในโหมด Drive ต่าง ๆ

	Ramp Slope	Frequency of loss Step (Hz)	speed (rad/s)
Full Step	50	3059.7	-95.6
	100	3103	-94.3
	150	3097	-91.4

รูปภาพแสดงความสัมพันธ์ของความเร็วและความถี่สุดท้ายก่อนเกิดการ Loss Step เมื่อมีการเปลี่ยนแปลง อัตราเร่ง 50, 100 และ 150

## สรุป

1. ความสัมพันธ์ระหว่าง Speed และ Frequency ของสัญญาณที่จ่ายให้ Stepper Motor

- ความเร็ว (Speed) ของ Stepper Motor มีความสัมพันธ์โดยตรงกับความถี่ (Frequency) ของสัญญาณที่จ่ายเข้า โดยความเร็วเพิ่มขึ้นเมื่อความถี่เพิ่มขึ้น
- ในโหมด Full-Step ความเร็วที่ได้จะต่ำกว่าโหมด Half-Step และ Micro-Step เนื่องจาก Step Angle ที่ใหญ่กว่า
- โหมด Micro-Step จึงแสดงให้เห็นราบรื่นในการเพิ่มความเร็วรอบ แม้ว่าจะต้องใช้ความถี่ที่สูงกว่าก็ตาม
- ข้อสังเกต การเปลี่ยนแปลงของ ความเร็วรอบ มีความสม่ำเสมอในโหมด Micro-Step มากที่สุด และเกิดการเคลื่อนที่ราบรื่นมากกว่าเมื่อเทียบกับ Full-Step และ Half-Step

2. ความสัมพันธ์ระหว่างความถี่และการเกิด Loss Step ในโหมดการควบคุมที่แตกต่างกัน

- Full-Step Mode เกิด Loss Step ที่ความถี่ต่ำสุดเมื่อเทียบกับ Half-Step และ Micro-Step เนื่องจาก การเคลื่อนที่มีความละเอียดต่ำ ทำให้ Stepper Motor ไม่สามารถตอบสนองต่อความถี่สูงได้
- Half-Step Mode สามารถรับความถี่ได้สูงกว่า Full-Step แต่ยังมีโอกาสเกิด Loss Step มากกว่า Micro-Step Mode หากความถี่เพิ่มขึ้นอย่างรวดเร็ว
- Micro-Step Mode รองรับความถี่สูงสุดก่อนเกิด Loss Step เนื่องจากการเคลื่อนที่มีความละเอียดสูง ช่วยลดแรงเสียดทานของโรเตอร์และเพิ่มความแม่นยำ
- ข้อสังเกต โน้มดการควบคุมที่ละเอียดกว่าจะช่วยลดความเสียงในการเกิด Loss Step แต่แรงบิดที่ลดลงใน โน้มด Micro-Step อาจเป็นข้อจำกัดในกรณีที่มีโหลดสูง

### 3. ผลกระทบของ Ramp Acceleration ต่อ Stepper Motor

การใช้ Ramp Acceleration ที่ มีอัตราเร่งต่ำ จะช่วยลดการเกิด Loss Step โดยเฉพาะในช่วงการเปลี่ยน ความเร็วที่สูงการเร่งความเร็วด้วย Ramp มีอัตราเร่งสูง(Acceleration สูง) จะทำให้เกิด Loss Step ได้ยากกว่า Ramp แบบค่อยเป็นค่อยไป ในโน้มด Micro-Step การใช้ Ramp Acceleration แบบช้าแสดงให้เห็นถึงความราบรื่น และลดการกระตุกได้ดีที่สุด ข้อสังเกต Ramp Acceleration ที่เหมาะสมช่วยให้ Stepper Motor เคลื่อนที่ด้วย กำลัง และลดการเปลี่ยนแปลงความเร็วโดยลดปัญหา Loss Step

#### อภิปรายผล

Full-Step Mode เหมาะสำหรับงานที่ต้องการแรงบิดสูง แต่ไม่เหมาะสมกับการทำงานที่ต้องการความเร็วสูงหรือ การเปลี่ยนแปลงความเร็วที่รวดเร็ว Half-Step Mode เป็นตัวกลางระหว่าง Full-Step และ Micro-Step เมามะ กับงานที่ต้องการความละเอียดและความเร็วปานกลาง Micro-Step Mode เหมาะสำหรับงานที่ต้องการความ แม่นยำสูง แต่ควรคำนึงถึงแรงบิดที่ลดลงและความถี่ที่ใช้ Acceleration มีความสำคัญในการปรับปรุงเสถียรภาพ ของการทำงาน และลดปัญหาการเกิด Loss Step โดยเฉพาะในโน้มด Micro-Step

#### ข้อเสนอแนะ

ควรจัดระเบียบผลการทดลองให้ชัดเจนและอ่านง่ายขึ้น เนื่องจากในรูปแบบปัจจุบันอาจทำให้ผู้อ่านเข้าใจ ข้อมูลได้ยาก โดยสามารถแบ่งข้อมูลออกเป็นหัวข้อหรือหมวดหมู่ที่ชัดเจน

#### อ้างอิง

1. <https://mall.factomart.com/principle-of-stepping-motor/>
2. <https://greensky-power.com/th/understanding-stepper-motor-basics/#:~:text=หลักการทำงานพื้นฐาน>  
ของสเต็ปเปอร์มอเตอร์คือ%3A%20เฟสของสเตเตอร์ตั้งแต่หนึ่งตัวขึ้นไปถูกรวมพลัง%2C%20กระแสไฟฟ้าผ่าน ชุดวงจรทำให้เกิดสนามแม่เหล็ก%2C และโรเตอร์จะอยู่ในแนวเดียวกับสนามแม่เหล็ก%3B%20การส่งกระแสไฟ ไฟฟ้าผ่านเฟสต่างๆ%20ตามลำดับ%2C%20โรเตอร์จะหมุนตามมุ่งที่กำหนดและไปถึงตำแหน่งที่ต้องการในที่สุด.
3. <https://blog.poscope.com/stepper-motor-driver/>

4. [https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html?utm\\_source=chatgpt.com](https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html?utm_source=chatgpt.com)
5. [https://www.zonemaker.com/product/2749/hanpose-stepper-motor-nema11-%E0%B9%81%E0%B8%A3%E0%B8%87%E0%B8%9A%E0%B8%B4%E0%B8%94-10-n-cm-1-0a-11hs3410?utm\\_source=chatgpt.com](https://www.zonemaker.com/product/2749/hanpose-stepper-motor-nema11-%E0%B9%81%E0%B8%A3%E0%B8%87%E0%B8%9A%E0%B8%B4%E0%B8%94-10-n-cm-1-0a-11hs3410?utm_source=chatgpt.com)
6. [https://www.faulhaber.com/en/know-how/tutorials/stepper-motor-tutorial-how-to-prevent-step-losses-with-stepper-motors/?utm\\_source=chatgpt.com](https://www.faulhaber.com/en/know-how/tutorials/stepper-motor-tutorial-how-to-prevent-step-losses-with-stepper-motors/?utm_source=chatgpt.com)
7. <https://www.alldatasheet.com/datasheet-pdf/pdf/1504991/CUID/AMT103-V.html>
8. <https://assunmotor.com/blog/incremental-encoder/>



LAB3 : Brushless DC Motor

#### Name

- นายณัฐพุทธ พุทธิวรดม 66340500018
- นางสาวนันท์นภัส คำสอน 66340500023
- นางสาวบุญวิลัย ปานทอง 66340500030

#### Objectives

- ออกแบบการทดลองโดยใช้ทักษะกระบวนการทางวิทยาศาสตร์ ในการ สืบเสาะพัฒนาระบบ pragmatics ทดลอง บันทึกผลการทดลอง สรุปผล และ อภิปรายผลการทดลอง เข้าใจหลักการทำงานของเซ็นเซอร์ และอุปกรณ์ทั้งหมด ที่ เกี่ยวข้องได้ ตลอดจนใช้หลักการทางวิทยาศาสตร์ คณิตศาสตร์ สถิติ และศาสตร์อื่นๆ ที่ เกี่ยวข้อง ประยุกต์ใช้ร่วมกับโปรแกรม MATLAB เพื่อเก็บผลการทดลอง วิเคราะห์ผล การทดลอง วิเคราะห์ความเที่ยงตรง ความแม่นยำ ได้อย่างถูกต้อง และมีเหตุผลรองรับ ตรวจสอบความถูกต้องเทียบกับทฤษฎีที่น่าเชื่อถือ
- อธิบายความสามารถในการรับรู้ปริมาณทางพิสิกส์ของเซ็นเซอร์ ทั้งหมดตั้งแต่ต้นจนจบกระบวนการได้ เช่น การอธิบายการวัดความเร็วของ Brushless DC Motor จาก Hall Sensor (จากคลื่นสัญญาณไฟฟ้าแปลงเป็นความเร็ว ได้อย่างไร)
- กำหนดตัวแปรในการทดลองได้อย่างถูกต้องและสมเหตุสมผล ไม่ ว่าจะเป็น ตัวแปรต้น ตัวแปรตาม ตัวแปรควบคุม อธิบายจุดประสงค์การทดลอง อธิบายสมมติฐานให้สอดคล้องกับตัวแปรที่กำหนด นิยามเชิงปฏิบัติการ และมีทฤษฎีที่ น่าเชื่อถือรองรับ เช่น ทฤษฎีทางพิสิกส์ หรือข้อมูลจาก Datasheet
- ออกแบบวิธีการทดลองเพื่อหาคำตอบ ให้สอดคล้องกับผลการ เรียนรู้ย่อทั้งหมดได้ด้วยตนเองอย่างถูกต้องตามหลักวิทยาศาสตร์ รวมทั้ง บันทึกผล สรุปผล อภิปรายผล ตามข้อมูลที่บันทึกได้จริง มีกระบวนการทำซ้ำ อธิบาย ที่มาของผล การทดลองนั้นได้ โดยใช้อุปกรณ์ เครื่องมือ ไฟล์ Simulink, mlx ฯลฯ และชุดการ ทดลองพร้อมบอร์ด Microcontroller ที่ TA จัดเตรียมให้เบื้องต้น
- เขียนรายงาน สัญลักษณ์ และสมการทางวิทยาศาสตร์ ได้อย่าง ถูกต้อง ทั้งขนาด และรูปแบบอักษร การเว้นช่องไฟ การเว้นขอบกระดาษ การเว้นระยะ พิมพ์ ให้ได้ระยะที่เหมาะสมตามหลักสากล

## Experiment 1:

### จุดประสงค์

1. เพื่อศึกษาหลักการทำงานของ Brushless DC Motor ,Sensorless Control ,Sensor-based Control รวมถึง BLDC Motor Control Techniques โดยมีเทคนิคที่ใช้ได้แก่ Trapezoidal ,Sinusoidal และ Field Oriented Control (FOC)
2. เพื่อศึกษาความแตกต่างของการควบคุมระหว่างแบบ 6-Step Control และแบบ Field Oriented Control (FOC) โดยเน้นในด้านของวิธีการเปลี่ยนและจัดการกับ Vector ของกระแสไฟฟ้า ผ่านการอธิบายด้วย Space Vector Modulation (SVM)
3. เพื่อศึกษาความแตกต่างของ BLDC Motor Control Techniques ที่ใช้เทคนิค Trapezoidal และ Sinusoidal
4. เพื่อศึกษาความสัมพันธ์ระหว่าง 6-Step Control และ Pulse Width Modulation (PWM)
5. เพื่อศึกษาความสัมพันธ์ระหว่าง Pulse Width Modulation (PWM) และ Frequency Output (Back EMF และ Hall Sensor)
6. เพื่อศึกษาความแตกต่างระหว่าง Feedback จาก Back EMF และ Hall Sensor
7. เพื่อหารือในการจัดการกับ Feedback ของ Sensorless Control เพื่อหาตำแหน่งในการนำมาเทียบกับ Feedback ของ Sensor-based Control บน ESC และ MCU (Op-Amp, Zero Crossing)
8. เพื่อศึกษาวิธีการหาความเร็วจาก Feedback ของ Trapezoidal Control

### สมมติฐาน

- ความถี่ของสัญญาณ Back-EMF จะเพิ่มขึ้น แบบแปรผันตรงกับความเร็วของ BLDC Motor
- สัญญาณ Back-EMF ในทั้ง 3 Phase จะมีแรงดันและความถี่เท่ากัน โดยแต่ละ Phase จะเลื่อนกัน 120 องศาทางไฟฟ้า
- จุด Zero-Crossing ของสัญญาณ Back-EMF สามารถใช้ตรวจสอบตำแหน่งของ Rotor ได้

## ตัวแปร

- ตัวแปรต้น : ค่าความเร็วรอบของ BLDC Motor ที่ปรับในโปรแกรม Motor Pilot
- ตัวแปรตาม : ความถี่ของสัญญาณ Back-EMF ทั้ง 3 Phase จาก BLDC Motor
- ตัวแปรควบคุม : อุณหภูมิ, ชนิดของ BLDC Motor, อุปกรณ์ที่ใช้ในการทดลอง, แหล่งจ่ายไฟ (Supply Voltage), Load บน BLDC Motor

## นิยามศัพท์เฉพาะ

1. BLDC Motor คือ มอเตอร์กระแสสลับแบบไร้แปรงถ่านที่ใช้การควบคุมด้วยระบบอิเล็กทรอนิกส์และสร้างแรงเคลื่อนไฟฟ้าจาก Back EMF เพื่อขับเคลื่อนโรเตอร์
2. สัญญาณ Back-EMF คือ แรงดันไฟฟ้าย้อนกลับที่เกิดขึ้นในชุดลวดของมอเตอร์ขณะที่โรเตอร์หมุนภายในสนามแม่เหล็ก ซึ่งแรงดันนี้ใช้เป็น Feedback ในการควบคุมมอเตอร์
3. Phase คือ ชุดลวดแต่ละชุดในมอเตอร์ BLDC ซึ่งสร้างแรงแม่เหล็กเพื่อขับเคลื่อนโรเตอร์ โดย BLDC Motor ทั่วไปมี 3 เฟส ได้แก่ U, V และ W

## นิยามเชิงปฏิบัติการ

1. U

สัญญาณใน phase 1 ของ Oscilloscope (สีเหลือง)

2. V

สัญญาณใน Phase 2 ของ Oscilloscope (สีฟ้า)

3. W

สัญญาณใน Phase 3 ของ Oscilloscope (สีเขียว)

4. Max App Speed

ความเร็วสูงสุดที่ตั้งค่าให้ในโปรแกรม Motor Pilot โดยในการทดลองนี้มีค่าเท่ากับ 9,864 RPM

5. Target Speed

ความเร็วที่กำหนดให้มอเตอร์หมุนในแต่ละครั้ง โดยสามารถตั้งค่าได้ทั้งในเชิงบวกและเชิงลบ เป็นเปอร์เซ็นต์ของ Max App Speed

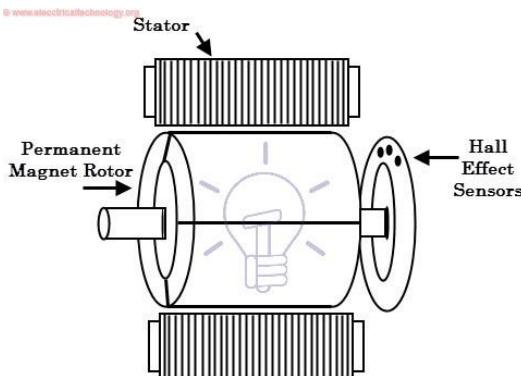
## เอกสารและงานวิจัยที่เกี่ยวข้อง

### 1. หลักการทำงานของ Brushless DC Motor (BLDC)

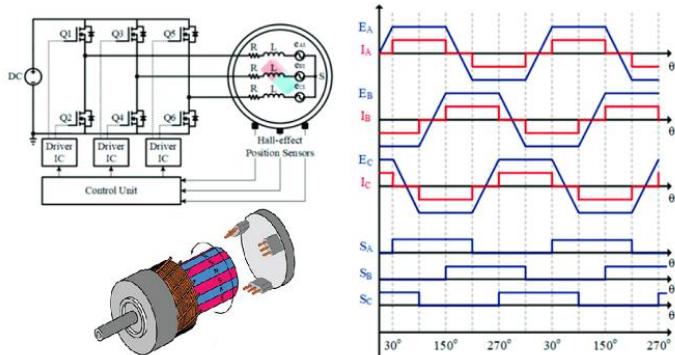
มอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor: BLDC) เป็นมอเตอร์ Synchronous ชนิดแม่เหล็กถาวรที่ขับเคลื่อนด้วยกระแสตรงและใช้ระบบ Commutation แบบควบคุมด้วยอิเล็กทรอนิกส์ แทนการใช้ระบบ Commutation แบบกลไก (Commutation คือกระบวนการเปลี่ยนกระแสในชุดลวดของมอเตอร์ในเวลาที่เหมาะสมเพื่อให้เกิดแรงบิดในการหมุน) BLDC ใช้ Rotor แบบแม่เหล็กถาวรและ Stator แบบชุดลวดที่มีกระแสไฟฟ้าผ่านตามลำดับโดยไม่มีการสัมผัสทางกลไก ตำแหน่งของ Rotor จะถูกตรวจจับโดยใช้ Hall Sensor หรือ Rotary Encoder ซึ่งช่วยกำหนดเวลาที่เหมาะสมในการสลับกระแสในชุดลวด

BLDC มีโครงสร้างที่สำคัญสองส่วน ได้แก่

- Rotor (ส่วนหมุน) : เป็นส่วนที่หมุนได้ ทำจากแม่เหล็กถาวร มีตั้งแต่ 2 ถึง 8 ชั้ว การใช้แม่เหล็กถาวรจะช่วยลดการสูญเสียพลังงานและเพิ่มแรงบิด โดย Rotor จะถูกออกแบบให้มีลักษณะเป็นแม่เหล็กแบบแรงดูดหรือแบบถาวร
- Stator (ส่วนอยู่กับที่) : เป็นส่วนที่อยู่กับที่ และประกอบด้วยชุดลวดทองแดงหลายชุด ชุดลวดเหล่านี้มักเรียกว่า ใจรูปแบบ 3 Phase ซึ่งได้รับพลังงานไฟฟ้าผ่านระบบอิเล็กทรอนิกส์กำลัง (Power Electronics) เช่น อินเวอร์เตอร์ เพื่อสร้างสนามแม่เหล็กหมุน

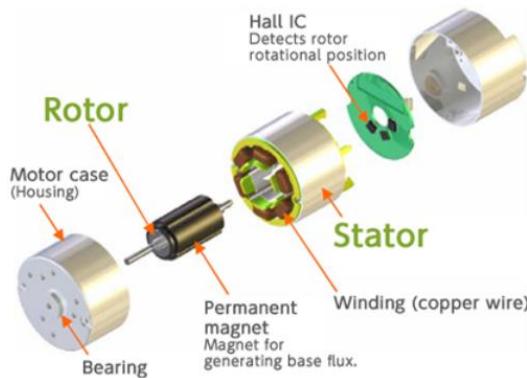


การทำงานของ BLDC จะเริ่มจากการที่กระแสไฟฟ้าถูกส่งไปยังชุดลวดภายใน Stator เพื่อสร้างสนามแม่เหล็กหมุน ซึ่งจะผลักหรือดึง Rotor ให้หมุน ตำแหน่งของ Rotor จะถูกตรวจจับโดย Hall Sensor หรือ Rotary Encoder เพื่อให้คอนโทรลเลอร์สามารถสลับกระแสไฟฟ้าในชุดลวด Stator ได้อย่างแม่นยำ ส่งผลให้มอเตอร์หมุนอย่างต่อเนื่อง โดยการสลับทิศทางของกระแสไฟฟ้าในชุดลวดจะถูกควบคุมด้วยวงจรอิเล็กทรอนิกส์

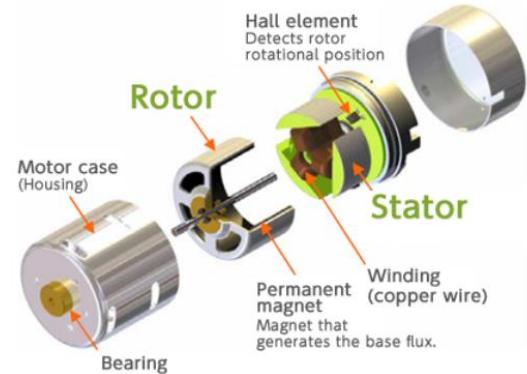


BLDC Motor สามารถแบ่งออกได้เป็น 2 ประเภทหลักตามโครงสร้างและวิธีการทำงาน

- Inner Rotor Design : Rotor จะอยู่ที่แกนกลางของมอเตอร์ล้อมรอบด้วยชุดลวด ทำให้ได้ขนาดเส้นผ่าศูนย์กลางของ Rotor ที่ใหญ่ขึ้น ส่งผลให้มอเตอร์มีแรงบิดสูงที่ความเร็วต่ำ และระบบความร้อนได้ดี

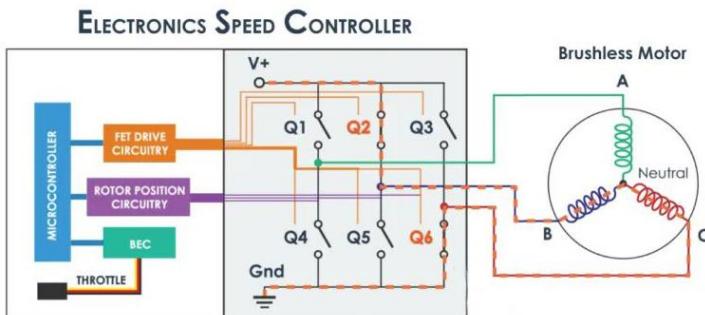


- Outer Rotor Design: Rotor จะถูกล้อมรอบด้วยชุดลวดที่อยู่ภายนอก ซึ่งจะช่วยลดการรบกวนเชิงกล ทำให้มอเตอร์มีความเร็วสูง แต่จะระบบความร้อนได้ยาก



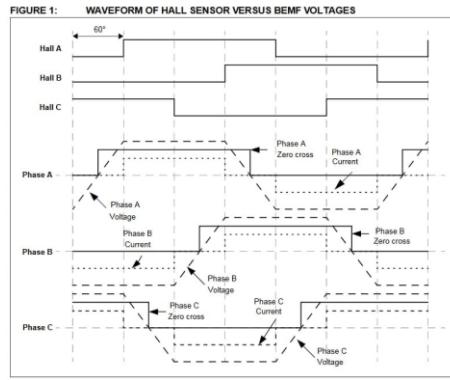
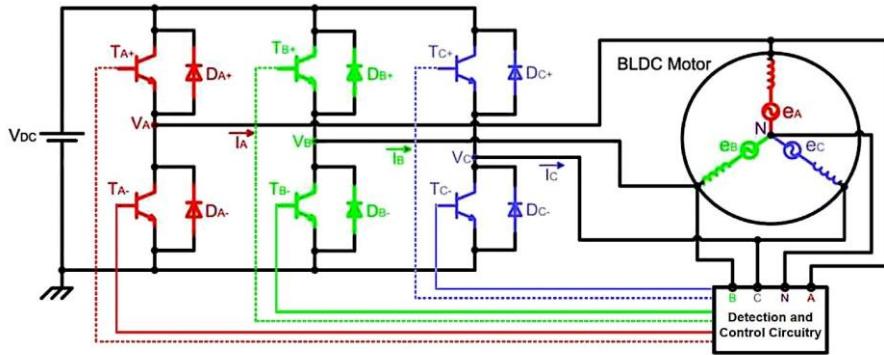
โดยการควบคุมความเร็วของมอเตอร์จะอาศัยการทำงานของ Electronic Speed Controller (ESC) หน้าที่เป็นตัวกลางควบคุมระหว่างแบตเตอรี่และมอเตอร์ BLDC โดยควบคุมการหมุนของมอเตอร์ผ่านการส่งสัญญาณไฟฟ้าแบบ 3 Phase ที่ปรับจังหวะการเปิด – ปิดชุดลวดของ Stator ตามคำสั่งจากคันเร่ง (Throttle) ปลายด้านหนึ่งของ ESC จะมีสายไฟสองเส้นเขื่อมต่อกับแบตเตอรี่ สายสีแดง (บวก) และสายสีดำ (ลบ) ส่วนปลาย

อีกด้านหนึ่งจะมีสายไฟสามเส้นที่เชื่อมต่อ ESC กับมอเตอร์ และส่วนขยายสุดท้ายจะเชื่อมต่อกับตัวรับคันเร่งที่ขับเคลื่อนด้วย BEC เมื่อกระแสไฟฟ้าถูกส่งไปยังชุดลวดของ stator ชุดลวดจะกล้ายเป็นแม่เหล็กไฟฟ้า สร้างขึ้นเหนือและใต้ที่ทำปฏิกิริยากับแม่เหล็กภายใน Rotor โดยขั้วที่เหมือนกันจะผลักกัน ทำให้ Rotor หมุน และ ESC จะปรับเปลี่ยนขั้วสนามแม่เหล็กไฟฟ้าอย่างต่อเนื่องเพื่อให้มอเตอร์หมุนอย่างต่อเนื่อง



## 2. Sensorless Control และ Sensor - Based Control

- Sensor - Based Control : การควบคุมจะอาศัยการใช้ Hall - Effect Sensors ที่ผึ้งอยู่ใน Stator เพื่อวัดตำแหน่งของ Rotor ที่หมุน โดยเซนเซอร์เหล่านี้จัดเรียงในตำแหน่งที่เท่า ๆ กัน เช่น  $60^\circ$  หรือ  $120^\circ$  รอบวงของ Stator เมื่อ Rotor หมุน สนามแม่เหล็กจากแม่เหล็กภายใน Rotor จะถูกตรวจจับโดย Sensor แล้วสร้างสัญญาณ Logic สูง หรือ ต่ำ ตามขั้วแม่เหล็กที่ผ่าน Sensor ในขณะนั้น การเปลี่ยนสถานะของสัญญาณจะทำให้ระบบสามารถระบุได้ว่า Rotor อยู่ในตำแหน่งใด และจะทำให้ระบบสามารถกำหนดลำดับการกระตุนชุดลวด (Commutation Sequence) ได้อย่างแม่นยำ เช่น ชุดลวดใดควรได้รับกระแสสูง (High) หรือกระแสต่ำ (Low) ในแต่ละช่วงเวลา และชุดลวดใดควรถูกปล่อย (Floating) ทำให้เกิดแรงแม่เหล็กที่ดึงหรือผลัก Rotor ให้หมุนไปยังตำแหน่งถัดไปในทิศทางที่ต้องการ
- Sensorless Control (Sensorless Trapezoidal): ใช้หลักการของแรงเคลื่อนไฟฟ้าย้อนกลับ (Back EMF) เพื่อกำหนดตำแหน่งของ Rotor เทียบกับ Stator ของมอเตอร์ โดย Back EMF ซึ่งเกิดขึ้นในชุดลวดของมอเตอร์ตามกฎของเลนซ์ (Lenz's Law) โดยเมื่อสนามแม่เหล็กเปลี่ยนแปลง EMF ที่เกิดขึ้นจะต้านการหมุนของมอเตอร์ และในมอเตอร์ที่มีฟลักซ์แม่เหล็กและจำนวนชุดลวดคงที่ ค่า EMF จะเป็นสัดส่วนกับความเร็วเชิงมุมของ Rotor โดยการติดตาม Back EMF ผ่านไมโครคอนโทรลเลอร์ที่ตั้งโปรแกรมไว้จะทำให้สามารถตรวจจับตำแหน่งสัมพัทธ์ระหว่าง stator และ rotor ได้โดยไม่ต้องใช้ Hall-effect Sensors ในช่วงเริ่มการทำงานมอเตอร์จะหยุดนิ่งและไม่สร้าง Back EMF ได้ทำให้คอนโทรลเลอร์ไม่สามารถระบุตำแหน่ง Rotor ได้ในช่วงเริ่มต้นการทำงาน ซึ่งสามารถแก้ได้ด้วยการเริ่มการทำงานในโหมดเปิด (Open Loop) เพื่อสร้าง EMF ให้เพียงพอ ก่อนที่คอนโทรลเลอร์จะเข้าควบคุมกระบวนการอย่างสมบูรณ์

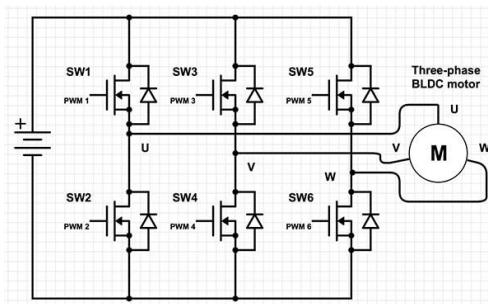


### การระบุตำแหน่งของ Rotor ในมอเตอร์

- Sensor - Based Control : ใช้ Hall - Effect Sensors เพื่อหาตำแหน่งของ Rotor โดยตรง ด้วยการตรวจจับการเปลี่ยนแปลงของสนามแม่เหล็กจาก Rotor ผ่านเซนเซอร์ที่ติดตั้งใน Stator
- Sensorless Control : ใช้ Back EMF ซึ่งสัมพันธ์กับความเร็วและตำแหน่งของ Rotor ในการหาตำแหน่ง Rotor แต่การใช้ Back EMF จะไม่สามารถระบุตำแหน่งได้ในช่วงเริ่มต้นการทำงาน และจำเป็นต้องเริ่มต้นในโหมดเปิด (Open Loop) ก่อนที่จะสามารถตรวจจับตำแหน่งได้ ทำให้มีความแม่นยำน้อยกว่า Sensor - Based Control โดยเฉพาะในช่วงเริ่มต้นการทำงาน

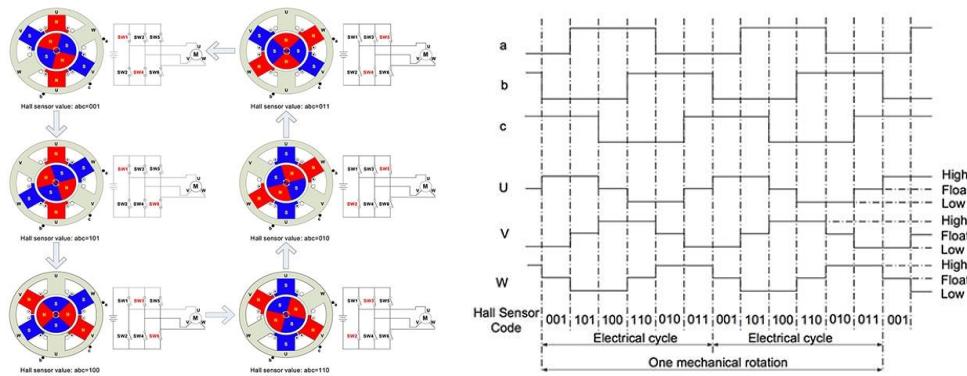
### 3. BLDC Motor Control Techniques

การควบคุมมอเตอร์ BLDC 3 Phases มักใช้ MOSFET Bridge ที่ประกอบด้วยคู่ MOSFET สามคู่ สำหรับควบคุมแต่ละ Phase ของมอเตอร์ โดยใช้ Pulse - Width Modulation (PWM) เพื่อแปลงแรงดัน DC เป็นแรงดันขับมอเตอร์



## การควบคุม Commutation ของมอเตอร์ BLDC มี 3 เทคนิค ได้แก่

- Trapezoidal Control (6-Step Block Commutation) เป็นวิธีใช้งานที่ง่ายที่สุด โดยจะกระตุ้นขดลวด 2 Phase (ขดลวดหนึ่งเส้นสูง และอีกเส้นหนึ่งต่ำ) ในแต่ละขั้นตอน ในขณะที่ขดลวด Phase ที่เหลือจะลอยตัว โดยขั้นตอนการกระตุ้นจะมี 6 ขั้นตอนในรอบหนึ่งของการหมุน ซึ่งแต่ละขั้นตอนจะสร้างแรงดันและกระแสในแต่ละขดลวด
  - ข้อดี: ควบคุมง่าย
  - ข้อเสีย: มี Torque Ripple โดยเฉพาะในช่วงความเร็วต่ำ
  - ลักษณะกราฟ: เป็นรูปทรงสี่เหลี่ยม (Square Wave) ซึ่งมีการกระตุ้นที่สม่ำเสมอระหว่าง Phase ที่ทำงาน

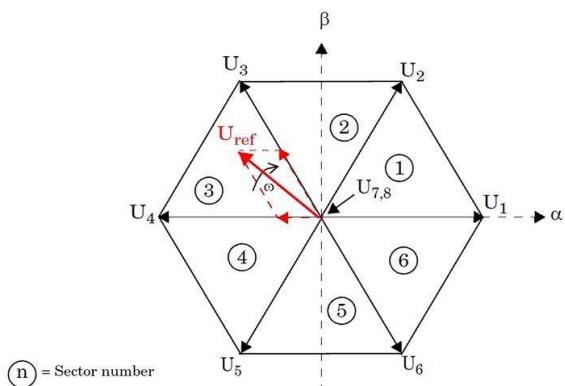


- Sinusoidal Control จะมีความซับซ้อนมากกว่า Trapezoidal Control แต่จะช่วยลด Torque Ripple ได้ โดยกระตุ้นกระแสในขดลวดทั้งสามด้วยมุ่ง Phase ต่างกัน  $120^\circ$  ทำให้การจ่ายพลังงานที่ราบรื่นกว่ามาก Trapezoidal Control
  - ข้อดี: มี Torque Ripple น้อยกว่า trapezoidal Control
  - ข้อเสีย: การควบคุมซับซ้อนและต้องการการควบคุมแรงบิดที่เปลี่ยน永久
  - ลักษณะกราฟ: มีลักษณะเป็นรูปไซน์ (Sine Wave) ซึ่งแสดงถึงการกระตุ้นที่ราบรื่นและสม่ำเสมอในแต่ละ Phase
- Field - Oriented Control (FOC) คือเทคนิคที่จะทำการควบคุมกระแสในขดลวดของ stator เพื่อให้ พลักด้วย Rotor และ พลักด้วย Stator ทำมุ่ง  $90^\circ$  กันตลอดเวลา ซึ่งช่วยให้มอเตอร์ทำงานได้ที่ประสิทธิภาพสูงสุด โดย FOC จะใช้การแปลงกระแสจากแบบ 3 Phase ไปเป็นแบบ 2 Phase (d-axis และ q-axis) และควบคุมกระแสในแกนเหล่านี้เพื่อให้มุ่งพลักด้วย Rotor และ Stator สมพันธ์กัน
  - ข้อดี: ลด Torque Ripple อย่างมากเมื่อเทียบกับ Trapezoidal Control และ Sinusoidal Control

- ข้อเสีย : มีความซับซ้อนและมีการคำนวณที่ซับซ้อนกว่าเทคนิคอื่น อีกทั้งยังต้องการการควบคุมที่มีความแม่นยำสูง
  - ลักษณะกราฟ: กราฟของกระแสจะมีลักษณะเป็นรูปวงกลมที่มีมุ่ง 90° ระหว่างแกน d-axis และ q-axis ซึ่งแสดงถึงการควบคุมที่มีความสมดุลและราบรื่น ส่วนกราฟแรงดันก็จะมีลักษณะเป็น Sinusoidal Wave แต่มีการควบคุมที่ดีกว่า Sinusoidal Control เนื่องจากมีการควบคุมมุมฟลักช์แบบสมบูรณ์

#### 4. Space Vector Modulation (SVM) และ 6-Step Control

Space Vector Modulation (SVM) เป็นเทคนิคที่ใช้ใน FOC สำหรับมอเตอร์恒流永磁同步馬達 (Permanent Magnet Synchronous Motor (PMSM)) โดย SVM จะสร้างสัญญาณ PWM ที่ควบคุมสวิตช์ของอินเวอร์เตอร์ เพื่อให้แรงดันที่ไม่ดูลสามารถขับมอเตอร์ให้หมุนที่ความเร็วหรือแรงบิดที่ต้องการ เทคนิคนี้ช่วยให้สามารถควบคุมแรงดันไฟฟ้าที่สูงไปยังมอเตอร์ได้อย่างแม่นยำ ด้วยการใช้ลำดับการสลับที่สอดคล้องกับเวลาของอินเวอร์เตอร์แรงดันอ้างอิงในแต่ละรอบ PWM



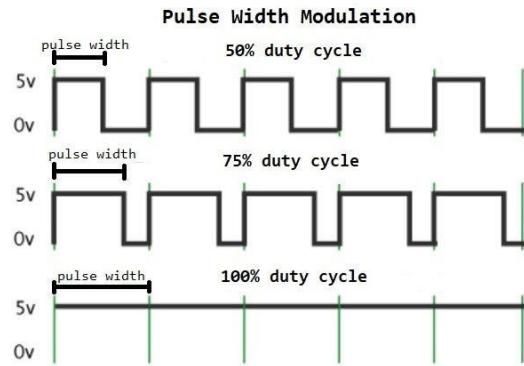
Space Vector	S1	S3	S5
U1	1	0	0
U2	1	1	0
U3	0	1	0
U4	0	1	1
U5	0	0	1
U6	1	0	1
U7	0	0	0
U8	1	1	1

6-Step Control และ FOC มีความแตกต่างกันในวิธีการจัดการกับเวกเตอร์กระแสไฟฟ้าในมอเตอร์ โดย 6-Step Control จะใช้การกระตุ้น Phase Stator ตามลำดับที่คงที่ทุก ๆ 60 องศา ซึ่งสร้างคลื่นกระแสไฟฟ้ารูปสี่เหลี่ยมคงที่ ทำให้มี Torque Ripple และประสิทธิภาพต่ำ โดยเฉพาะที่ความเร็วต่ำ แต่ FOC จะใช้ SVM ใน การปรับเวกเตอร์กระแสไฟฟ้าอย่างต่อเนื่อง โดยควบคุมมุ่งหวัง Rotor และฟลักซ์ของ Stator ให้เป็น 90 องศา ช่วยลด Torque Ripple และเพิ่มประสิทธิภาพ โดยเฉพาะที่ความเร็วสูง ดังนั้นการใช้เทคนิค FOC ด้วย SVM จะให้การควบคุมที่ละเอียดกว่า ทำให้มอเตอร์มีประสิทธิภาพสูงและแรงบิดเสถียรกว่า

## 5. Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) គឺ មេនីកថានៃការគុរចុមទំប័ននៅផ្លូវដែលសំរាប់ប្រើប្រាស់នៅក្នុងការប្រព័ន្ធដែលត្រូវបានបង្កើតឡើង។ នៅក្នុងការប្រព័ន្ធ នឹងបានបង្កើតឡើងជាប្រព័ន្ធទិន្នន័យ ដែលមានការគុរចុមទំប័នដែលត្រូវបានបង្កើតឡើង។ នៅក្នុងការប្រព័ន្ធទិន្នន័យ នឹងបានបង្កើតឡើងជាប្រព័ន្ធទិន្នន័យ ដែលមានការគុរចុមទំប័នដែលត្រូវបានបង្កើតឡើង។

แบบอนาล็อกด้วยสัญญาณดิจิตอลได้ โดยการควบคุมความกว้างของพัลซ์ (PWM) จะกำหนดค่าแรงดันเฉลี่ยที่มอเตอร์ได้รับ โดยการปรับความกว้างของ Pulse หรือ Duty Cycle การปรับความกว้างของ Pulse มาขึ้นจะทำให้แรงดันเฉลี่ยสูงขึ้น ซึ่งจะทำให้มอเตอร์หมุนเร็วขึ้น ในขณะที่การลดความกว้างของ Pulse จะลดแรงดันเฉลี่ยและทำให้มอเตอร์หมุนช้าลง

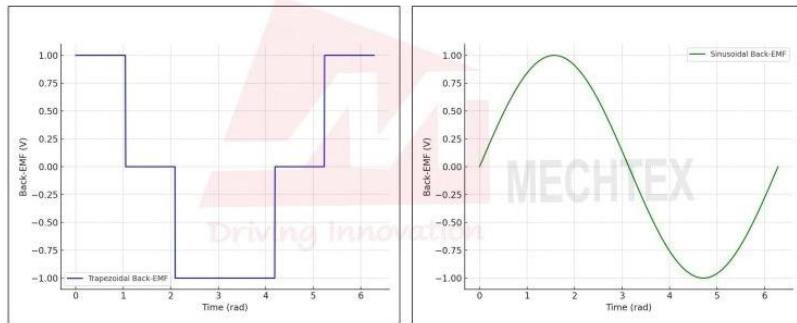


## 6. Feedback จาก Back EMF และ Hall Sensors

แรงเคลื่อนไฟฟ้าย้อนกลับ (Back EMF) : แรงดันไฟฟ้าที่เกิดขึ้นในมอเตอร์ไฟฟ้า โดยเฉพาะในมอเตอร์ BLDC เนื่องจากการหมุนของมอเตอร์ เมื่อ rotor ของมอเตอร์ BLDC หมุนภายในสนามแม่เหล็ก ก็จะเหนี่ยวนำให้เกิดแรงดันไฟฟ้าในขดลวดของ stator ที่ต้านกับแรงดันไฟฟ้าที่ใช้ ซึ่งเป็นผลจากกฎการเหนี่ยวนำแม่เหล็กไฟฟ้าของ法拉เดย์ ซึ่งระบุว่าการเปลี่ยนแปลงของฟลักซ์แม่เหล็กผ่านตัวนำจะเหนี่ยวนำให้เกิดแรงเคลื่อนไฟฟ้า (EMF) ซึ่งมีความสำคัญในการช่วยปรับความเร็วและเพิ่มประสิทธิภาพของมอเตอร์ ขณะเดียวกันยังจำกัดความเร็วสูงสุดของมอเตอร์เนื่องจาก Back-EMF จะสมดุลกับแรงดันที่จ่ายเข้า

Back EMF ในมอเตอร์ BLDC ที่พบบ่อยมี 2 ประเภท

- Trapezoidal Back EMF : พบมากที่สุดในมอเตอร์ BLDC ที่มีการกระจายขดลวดแบบ Trapezoidal ซึ่งจะช่วยให้ Back EMF ตรงกับกระแสที่จ่ายเข้าสู่ขดลวด เพื่อให้การสลับกระแส (Commutation) ทำงานได้ราบรื่น
- Sinusoidal Back EMF: พบในมอเตอร์ BLDC ที่มีการกระจายขดลวดแบบไชน์ ซึ่งหมายความว่ากับการควบคุมด้วย Field-Oriented Control (FOC) เพื่อให้ได้แรงบิดที่นุ่มนวล



Trapezoidal Back-EMF

Sinusoidal Back-EMF

การเกิด Back EMF ในมอเตอร์ BLDC จะอิงตามกฎของพาราเดีย เมื่อ rotor หมุน ฟลักซ์แม่เหล็กที่เชื่อมโยง กับชด漉ดของ stator เปลี่ยนแปลง ส่งผลให้เกิดแรงดันไฟฟ้า (Back EMF) มีสูตรคำนวณดังนี้

$$E_b = K_e \cdot \omega$$

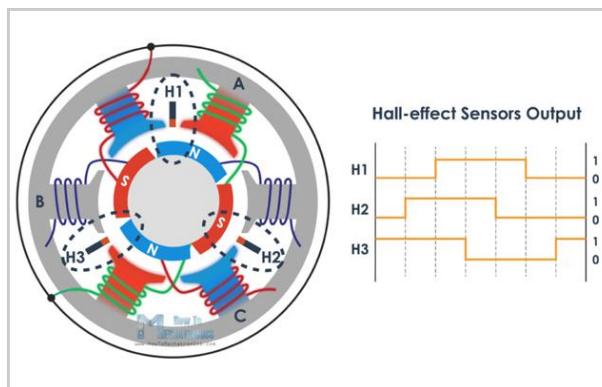
โดยที่  $E_b$  แทน Back EMF หน่วยเป็น V

$K_e$  แทน ค่าคงที่ Back EMF หน่วยเป็น  $V \cdot rad^{-1} \cdot s$

$\omega$  แทน ความเร็วเฉลี่ย หน่วยเป็น rad/s

Back EMF มีบทบาทสำคัญในการควบคุมมอเตอร์ BLDC โดยช่วยให้สามารถควบคุมความเร็วของมอเตอร์ได้อย่างแม่นยำ เนื่องจากมีความสัมพันธ์โดยตรงกับความเร็วของ Rotor ช่วยป้องกันการหมุนเร็วเกินไปโดยการลดกระแสที่เหลือเข้าสู่ชด漉ด และเพิ่มประสิทธิภาพการทำงานของมอเตอร์ด้วยการลดการสูญเสียพลังงาน อีกทั้งยังมีบทบาทในระบบเบรกแบบสร้างพลังงานกลับ (Regenerative Braking)

Hall Effect Sensors ช่วยให้ BLDC มอเตอร์ทำงานได้อย่างมีประสิทธิภาพโดยการระบุตำแหน่งของ Rotor ผ่านการตรวจจับการเปลี่ยนแปลงของสนามแม่เหล็กแล้วให้สัญญาณโลจิก High และ Low แทนการตรวจจับ โดยข้อมูลนี้จะถูกใช้ในการคำนวณจังหวะการจ่ายกระแสไฟฟ้าเข้าสู่ Stator เพื่อให้แรงบิดสอดคล้องกับตำแหน่งของ Rotor

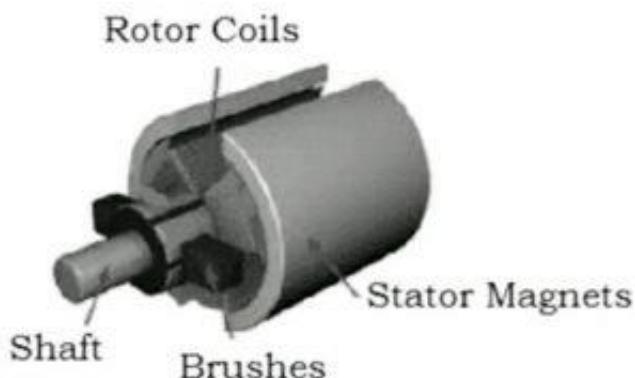


ในการประมวลผล Feedback ที่ได้รับ Op - Amp (Operational Amplifier) จะทำหน้าที่ขยายหรือเปรียบเทียบสัญญาณที่ได้จาก Back - EMF หรือ Hall Sensors เพื่อให้เหมาะสมกับการวิเคราะห์ของวงจร

ควบคุม โดย Op - Amp จะทำหน้าที่ตรวจจับช่วงที่ Back - EMF ข้ามผ่านแรงดันศูนย์ แล้วใช้ Comparator ภายใน Op - Amp เพื่อสร้างสัญญาณดิจิทัลที่แสดงจังหวะ Zero Crossing (ตรวจจับจุดที่แรงดันหรือกระแสของ สัญญาณเปลี่ยนผ่านศูนย์) จากนั้นสัญญาณนี้จะถูกส่งต่อให้ไมโครคอนโทรลเลอร์เพื่อประมวลผลการคำนวณ

## 7. Permanent Magnet Synchronous Motor (PMSM)

มอเตอร์ชนิด synchronous ที่ใช้แม่เหล็กในการสร้างสนามกระตุน ซึ่งทำให้มันหมุนโดยไม่มีการใช้เบรค ถ่านและไม่ต้องการกระแสไฟฟ้าผ่าน rotor มอเตอร์นี้มีลักษณะการกระจายฟลักซ์ในรูปแบบ sine wave ทั้งใน ส่วนของฟลักซ์และกระแสไฟฟ้า ซึ่งช่วยให้มอเตอร์มีขนาดเล็กและมีประสิทธิภาพสูง โดยการควบคุมมักใช้กับ ไมโครคอนโทรลเลอร์ขั้นสูงเช่น PIC32MK หรือ dsPIC33 เพื่อให้สามารถควบคุมได้อย่างแม่นยำ มอเตอร์นี้จะ ประกอบด้วยส่วนที่เป็น stator (ไม่หมุน) และ rotor (ส่วนที่หมุนประกอบด้วยแม่เหล็กการที่มีแรงเหนี่ยวนำสูง



มอเตอร์ synchronous จะแบ่งประเภทเป็นมอเตอร์เป็น 2 ประเภท

- salient pole: มีค่าเหนี่ยวนำในทิศทางตรงและเชิงมุมไม่เท่ากัน ( $L_d \neq L_q$ )
- non-salient pole: มีค่าเหนี่ยวนำในทิศทางตรงและเชิงมุมเท่ากัน ( $L_d = L_q$ )

PMSM นั้นมีความคล้ายคลึงกับ BLDC หากแต่ยังมีข้อแตกต่างกันดังตารางด้านล่าง

PMSM	BLDC
มอเตอร์ AC แบบไร้เบรคถ่าน	มอเตอร์ DC แบบไร้เบรคถ่าน
ไม่มีแรงบิดริบเปิล	มีแรงบิดริบเปิล
ประสิทธิภาพสูงกว่า	ประสิทธิภาพต่ำกว่า
มีเสียงรบกวนต่ำ	มีเสียงรบกวนสูงกว่า
ใช้ในงานอุตสาหกรรม รถยนต์ หุ่นยนต์ ระบบเชอร์โว มอเตอร์ และรถไฟ	ใช้ในระบบพวงมาลัยไฟฟ้า HVAC และรถไฟไฮบริด

การควบคุมมอเตอร์ PMSM แบบ Field-Oriented Control (FOC) เป็นเทคนิคที่ควบคุมกระแสสเตเตอเริ่มให้ สัมพันธ์กับกรอบอ้างอิงหมุน (Rotating Reference Frame - RRF) เพื่อปรับแรงบิดและฟลักซ์ได้อย่างมี ประสิทธิภาพ โดยกรอบอ้างอิงจะถูกตั้งให้แกน d-axis อยู่ในแนวเดียวกับฟลักซ์ของมอเตอร์ การควบคุมนี้ใช้

เซ็นเซอร์วัดตำแหน่งโรเตอร์หรือคำนวณแบบไร้เซ็นเซอร์ (Sensorless) เพื่อลดความซับซ้อนและเพิ่มประสิทธิภาพ  
เหมาะสมสำหรับระบบที่ต้องการการตอบสนองรวดเร็วและแรงบิดที่ราบรื่น เช่น หุ่นยนต์และยานยนต์ไฟฟ้า มี  
วิธีการควบคุมดังนี้

- การเปลี่ยนกรอบอ้างอิง: เปลี่ยนสัญญาณกระแสและแรงดันของมอเตอร์จากการกรอบอ้างอิงแบบคงที่  
(Stationary Reference Frame) เป็นกรอบอ้างอิงหมุน (Rotating Reference Frame - RRF) โดยใช้ Park's Transformation เพื่อแยกส่วนควบคุมแรงบิด (q-axis) และฟลักซ์ (d-axis)
- การคำนวณแรงบิดและฟลักซ์: กระแส d-axis ใช้ควบคุมฟลักซ์แม่เหล็ก  
กระแส q-axis ใช้ควบคุมแรงบิด
- การควบคุมกระแสด้วย PID: ใช้ตัวควบคุม PID สำหรับกระแส d และ q เพื่อรักษาระดับแรงดันไฟฟ้าที่เหมาะสมในกรอบ RRF
- การกลับกรอบอ้างอิง: ผลลัพธ์จากการควบคุมจะถูกแปลงกลับเป็นกรอบอ้างอิงแบบคงที่เพื่อควบคุม อินเวอร์เตอร์
- การใช้เซ็นเซอร์หรือเทคนิคไร้เซ็นเซอร์: ตำแหน่งโรเตอร์สามารถวัดได้ด้วย encoder หรือประมาณค่าด้วย เทคนิคไร้เซ็นเซอร์ (Sensorless)

#### 8. A2212/13T TECHNICAL DATA

No. of Cells:	2 - 3 Li-Poly 6 - 10 NiCd/NiMH
Kv:	1000 RPM/V
Max Efficiency:	80%
Max Efficiency Current:	4 - 10A (>75%)
No Load Current:	0.5A @10V
Resistance:	0.090 ohms
Max Current:	13A for 60S
Max Watts:	150W
Weight:	52.7 g / 1.86 oz
Size:	28 mm dia x 28 mm bell length

Shaft Diameter:	3.2 mm
Poles:	14
Model Weight:	300 - 800g / 10.5 - 28.2 oz

## 9. สูตรคำนวณความเร็ว

โดยกำหนดให้

- Speed** คือ ความเร็วรอบของ Brushless DC Motor ในหน่วย RPM
- f** คือ ความถี่ของสัญญาณ Back-EMF ในหน่วย Hz
- P** คือ จำนวนขั้วแม่เหล็ก (Pole) ของ Brushless DC Motor ซึ่งใน Datasheet ของ Brushless DC Motor รุ่น A2212/13T มีจำนวน Poles คือ 14
- 60** คือ ค่าการแปลงจากวินาทีเป็นนาที

สมการความเร็วซิงโครนัส synchronous speed (Ns)

$$Speed = \frac{f \times 120}{P}$$

### วิธีดำเนินการทดลอง

ควบคุมการทำงานของ มอเตอร์ BLDC Motor ผ่านการใช้โปรแกรม ST Motor Pilot v1.2.12 และอ่านค่า Back EMF จาก Oscilloscope เพื่อนำมาคำนวณความเร็วของมอเตอร์ จากนั้นจึงเปรียบเทียบความเร็วที่ตั้งค่าในโปรแกรมกับความเร็วที่คำนวณได้

### วัสดุอุปกรณ์

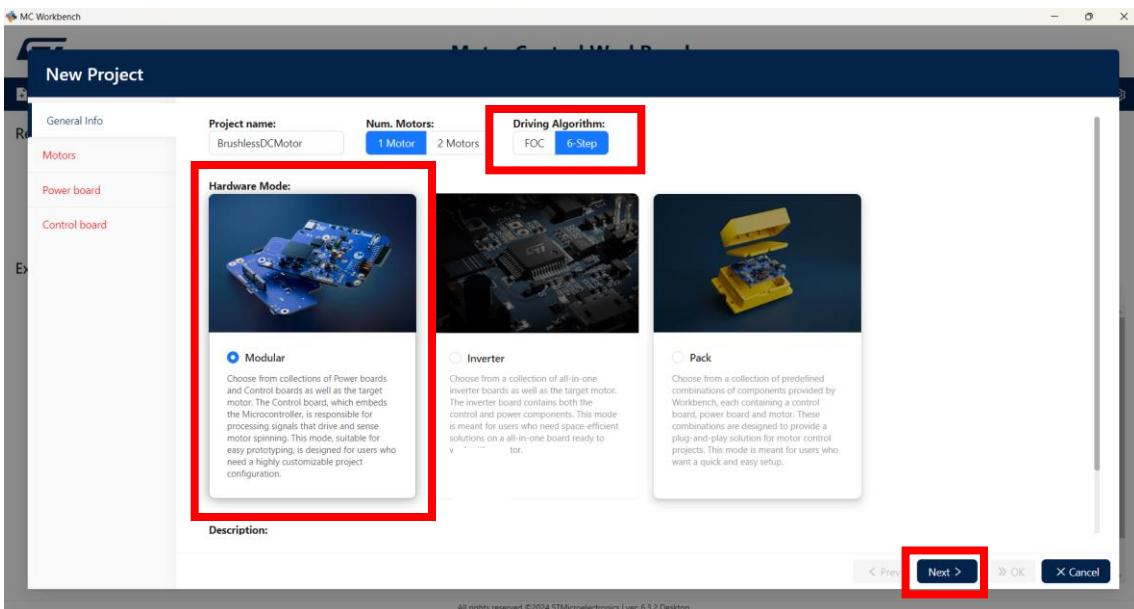
1. Brushless DC Motor จำนวน 1 อัน
2. STMICROELECTRONICS X-NUCLEO-IHM08M1 จำนวน 1 อัน
3. Nucleo STM32G474RE และสายอัปโหลด จำนวน 1 ชุด
4. BLDCXplorer จำนวน 1 ชุด ประกอบไปด้วย ฐานสามารถจับต่อวงจร Breadboard, 3D-Print ใช้สำหรับการประกอบกับ BLDC Motor
5. Oscilloscope (Tablet Oscilloscope Smart Series STO1004 100MHz Digital Handheld)
6. Adapter 24 V

### ขั้นตอนการทดลอง

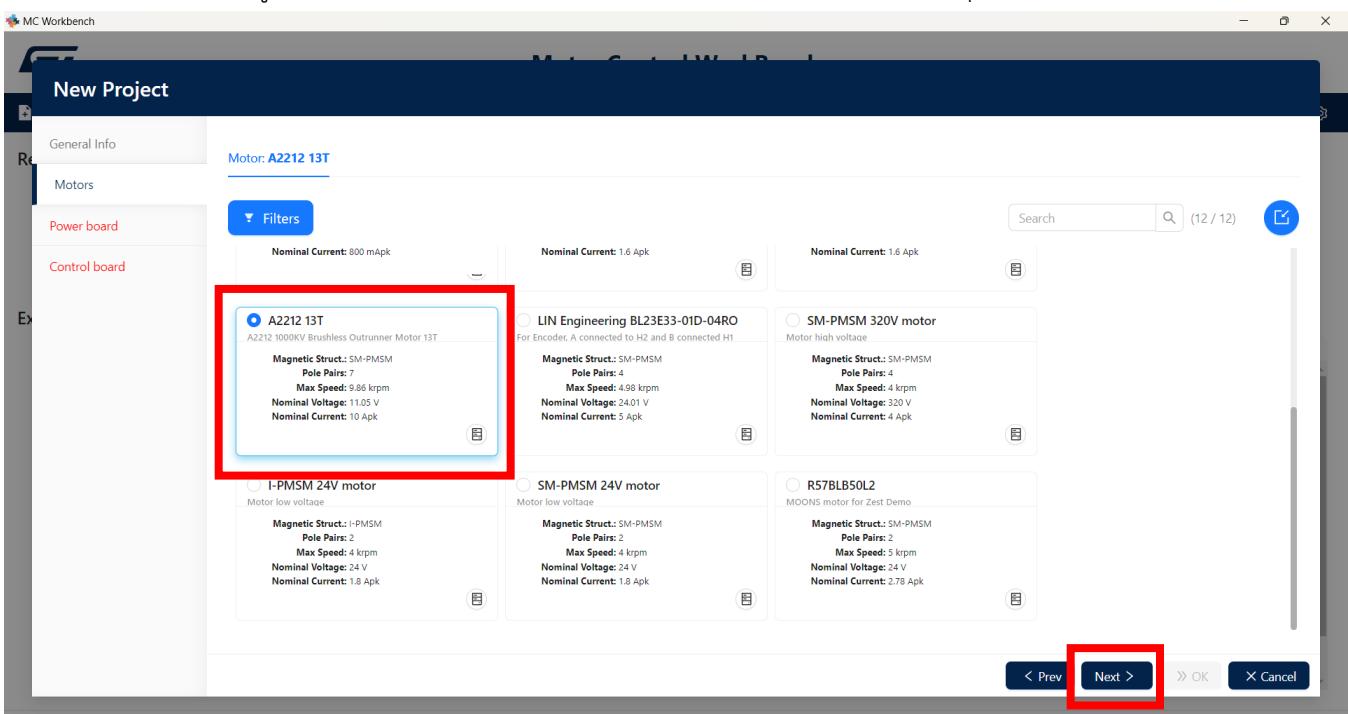
1. การควบคุม BLDC Motor ด้วยโปรแกรม ST Motor Pilot v1.2.12 จะต้องใช้โปรแกรม Motor Control (MC) Workbench เพื่อสร้างไฟล์ในการกำหนดค่าที่เหมาะสม จากนั้นนำไฟล์ไปใช้งานใน ST Motor Pilot ต่อไป เริ่มต้นด้วยการตั้งค่าพารามิเตอร์ของมอเตอร์และบอร์ดควบคุมในโปรแกรม

โดยเมื่อเปิดโปรแกรม Motor Control (MC) Workbench ขึ้นมาครั้งแรก โปรแกรมจะขึ้นหน้า New Project ให้ทำการตั้งค่า Driving Algorithm เป็น 6-Step และ Hardware Mode: เป็นรูปแบบ Modular จากนั้น กดปุ่ม

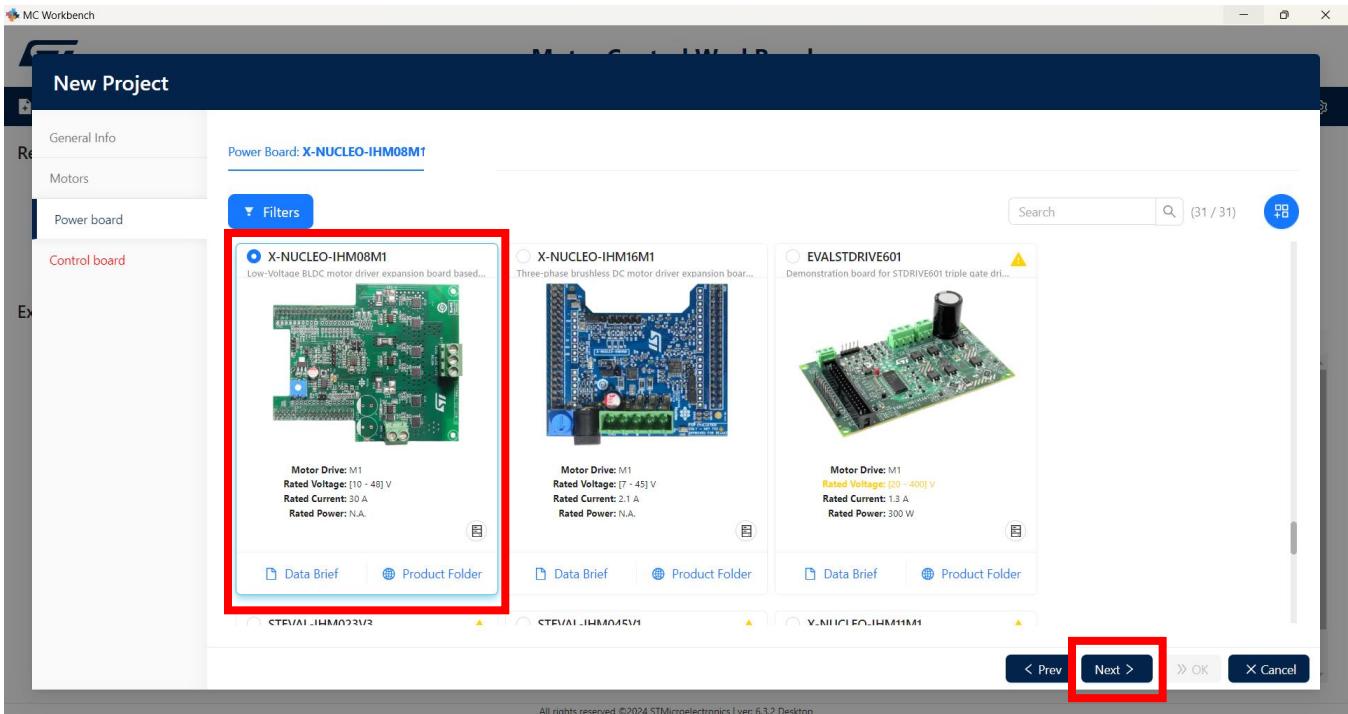
Next



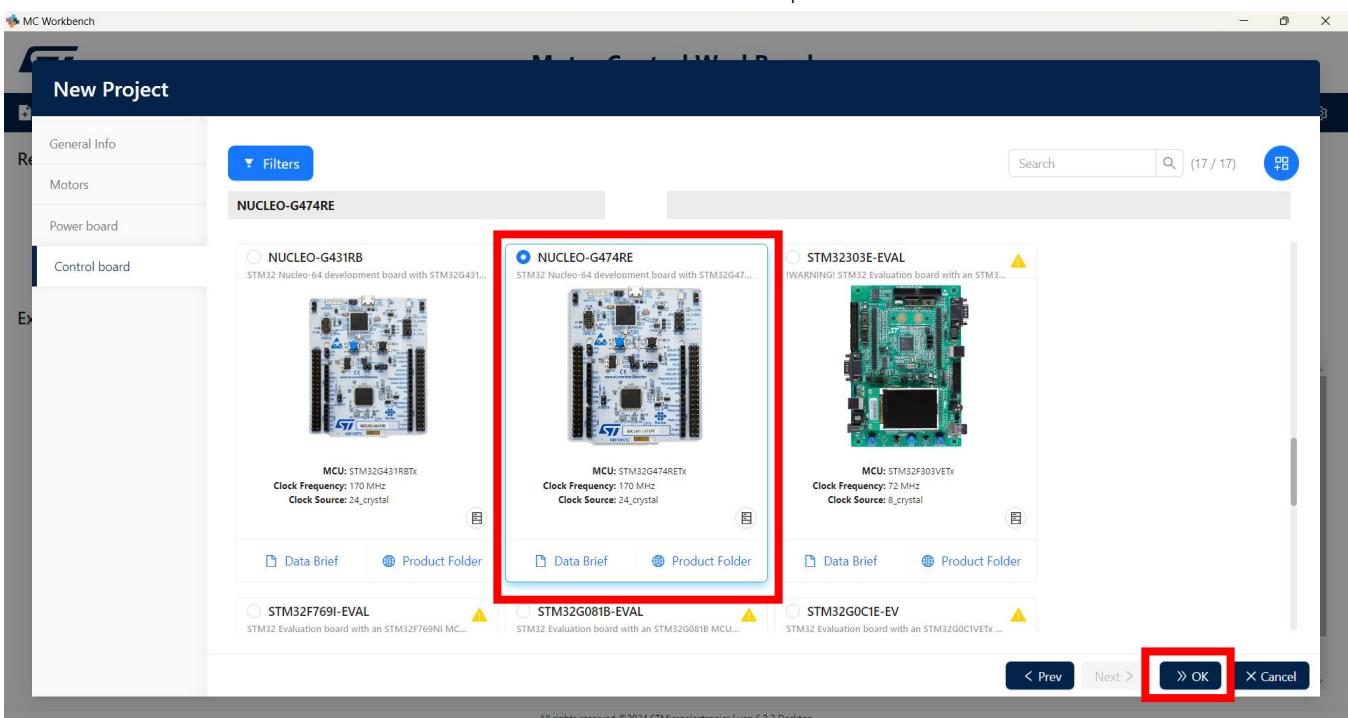
ເນື່ອງຍູ້ໃນທີ່ Motors ໄທເລືອກ motor ຮ້າສ A2212 13T ຈາກນັ້ນດັບປຸ່ມ Next



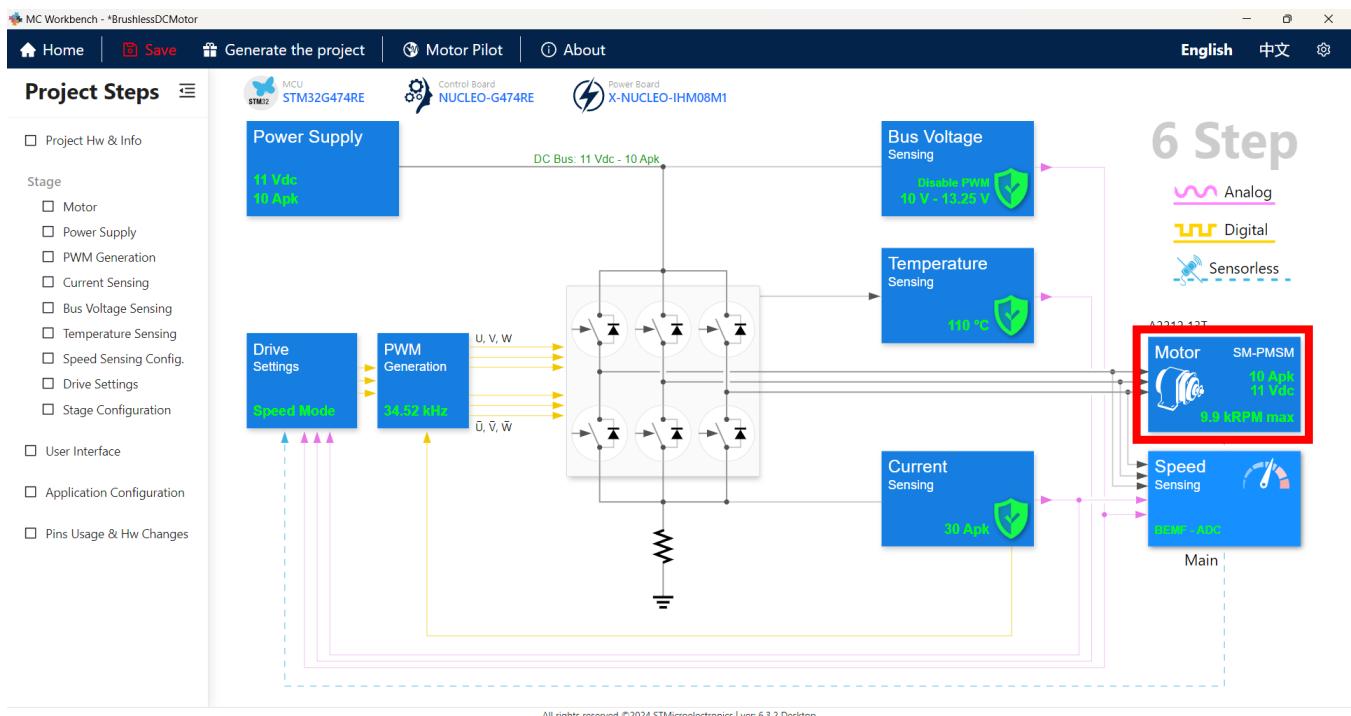
ໃນທີ່ Power board ເລືອກ X-NUCLEO-IHM08M1 ຈາກນັ້ນດັບປຸ່ມ Next



ในหน้า Control board เลือก NUCLEO-G474RE จากนั้นกดปุ่ม Next

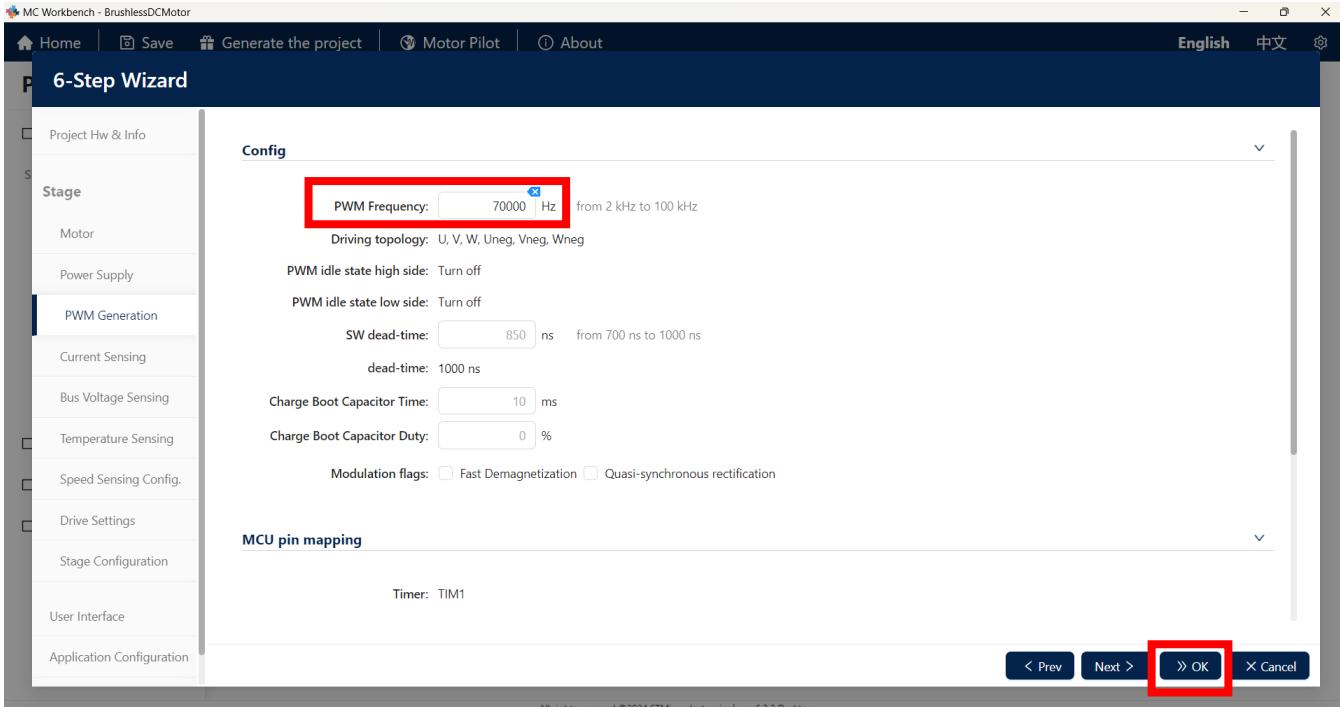


กดเลือกที่ Motor เพื่อแก้ไขค่าต่างๆ



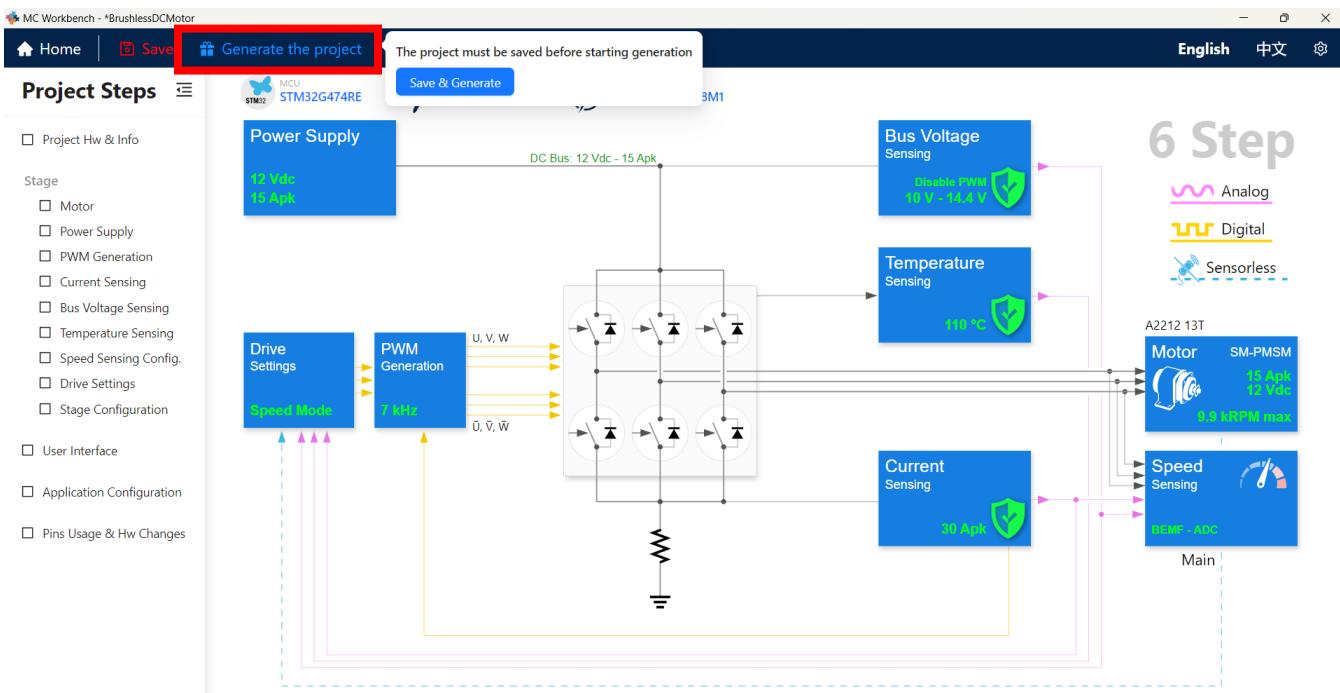
ในหน้า Motor ตั้งค่า Max current เป็น 15Apk และ ตั้งค่า Max DC Voltage เป็น 12V

จากนั้นไปที่หน้า PWM Generation และทำการตั้งค่า PWM Frequency เป็น 70,000 Hz เมื่อตั้งค่าเสร็จทั้งหมดแล้วทำการบันทึกค่าที่แก้ไขแล้วด้วยการกดปุ่ม OK



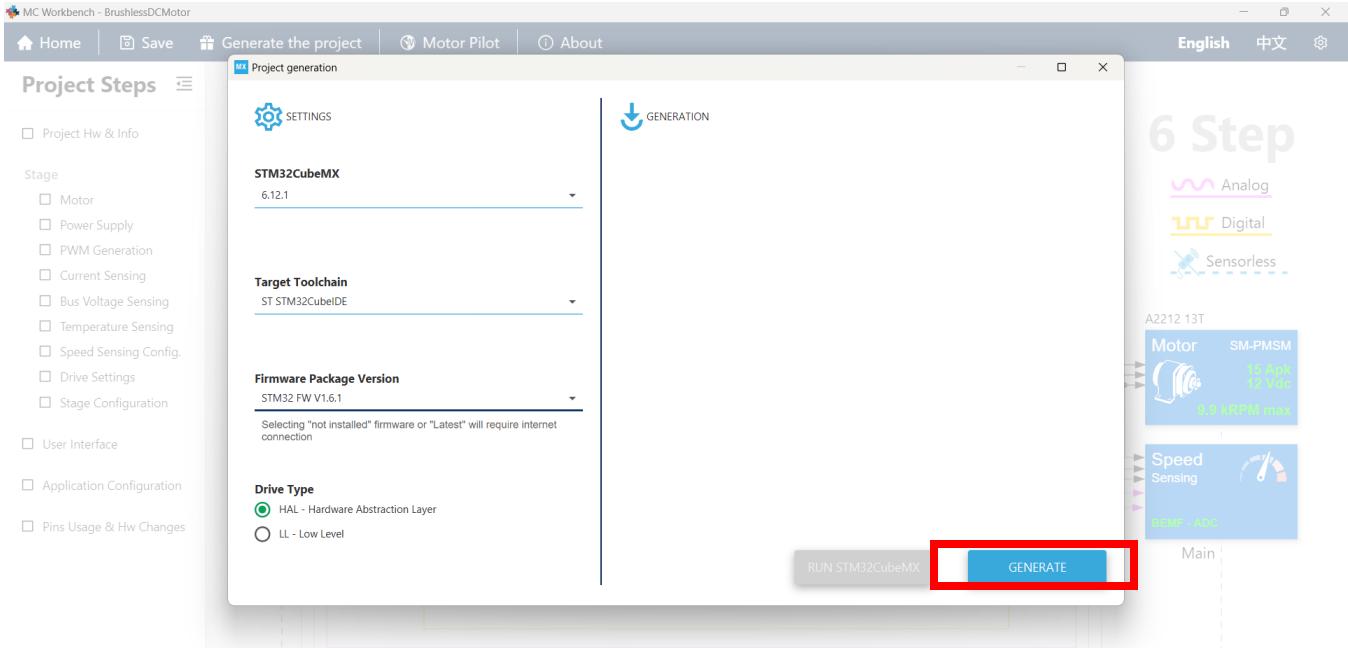
เมื่อตั้งค่า ทุกอย่างในโปรแกรม Motor Control (MC) Workbench เสร็จแล้ว ให้ทำการ save แล้ว เปิด

Project ใน STM32cudeIDE โดยการ กดปุ่ม Generate the project

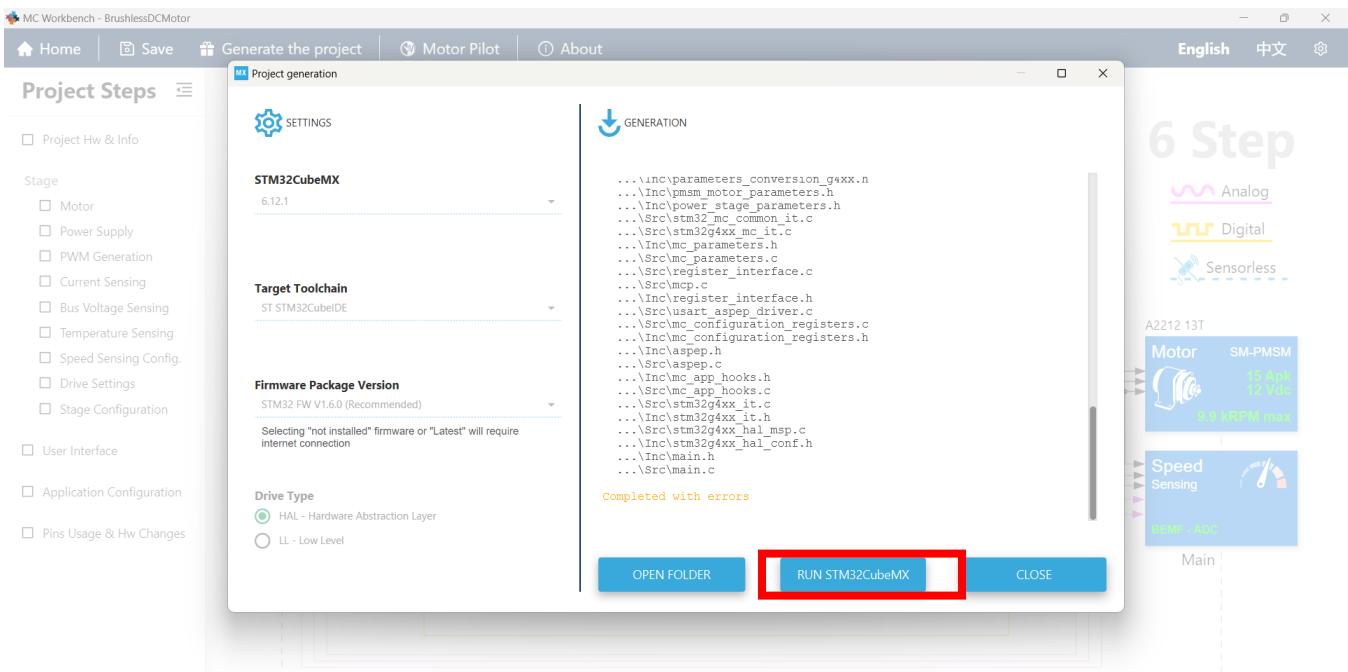


จากนั้นกดปุ่ม Generate โค้ด ตามขั้นตอนดังรูปด้านล่าง

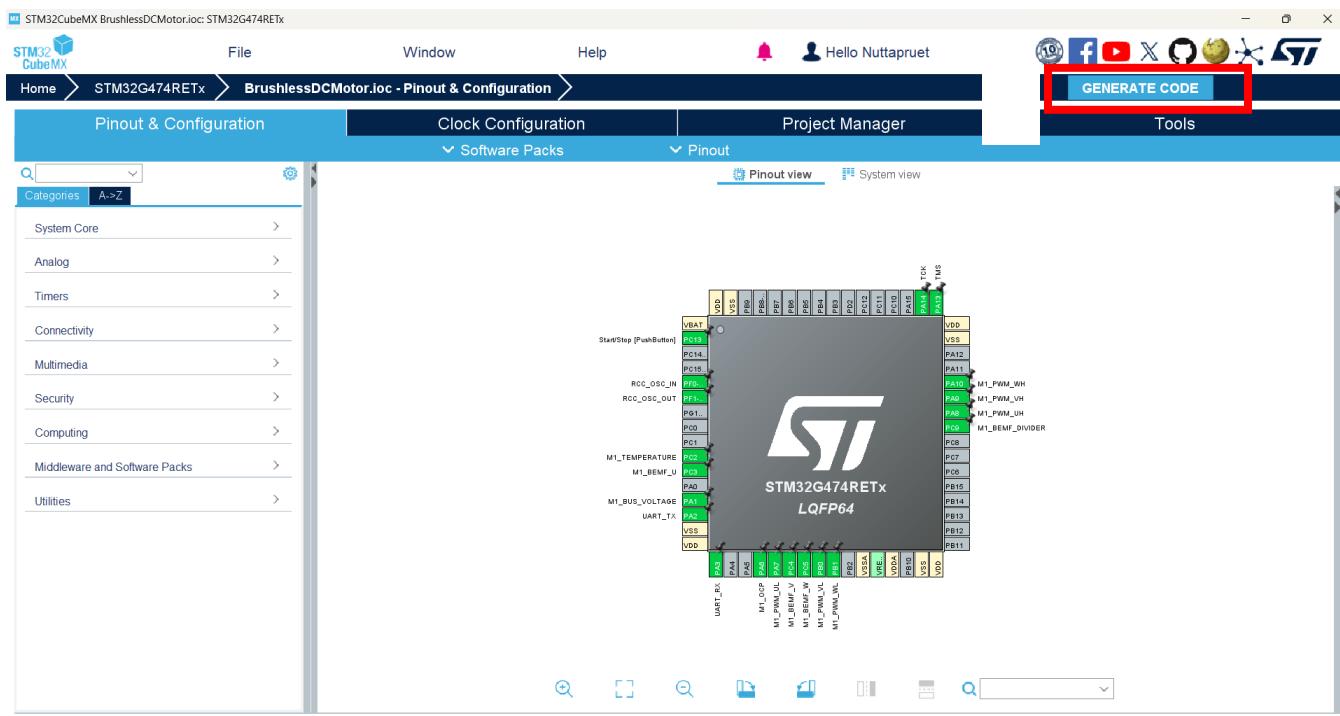
ขั้นตอนที่ 1 กดปุ่ม GENERATE



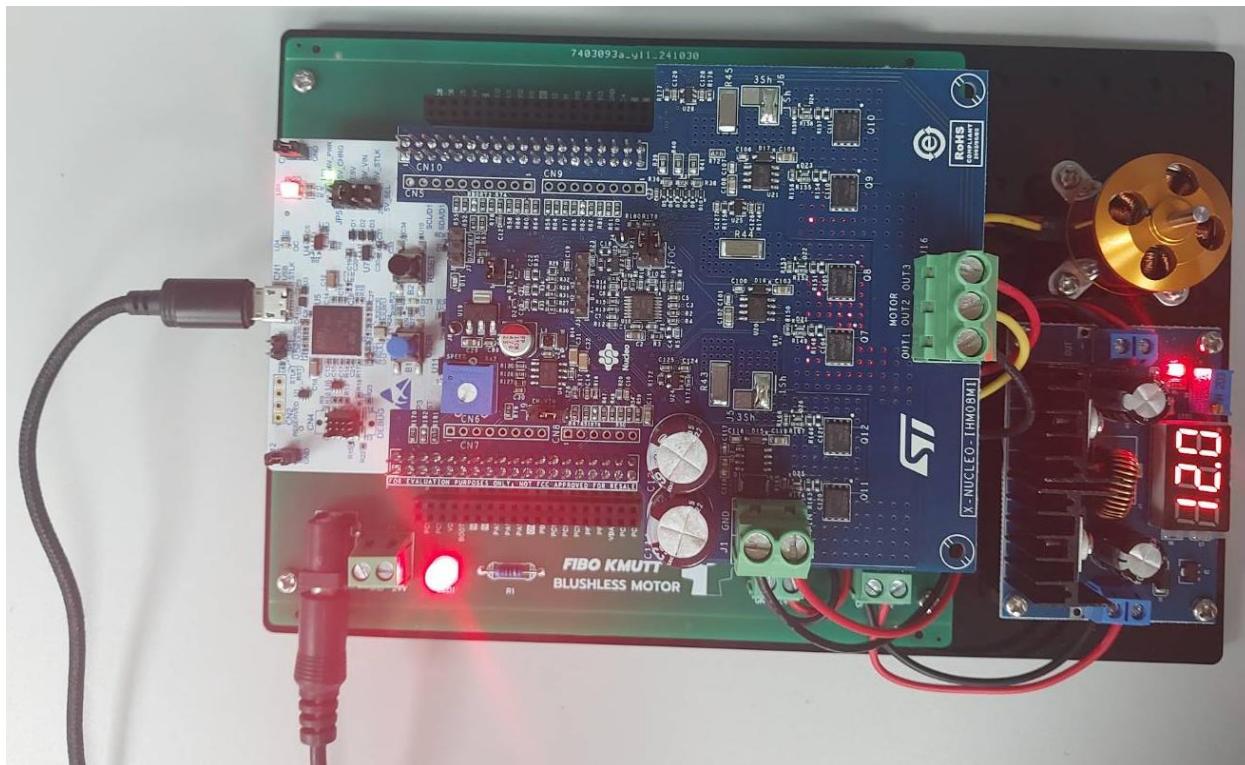
ขั้นตอนที่ 2 กดปุ่ม RUN STM32CubeMX



ขั้นตอนที่ 3 กดปุ่ม GENERATE CODE



เชื่อมต่อบอร์ด Nucleo STM32G474RE เข้ากับ คอมพิวเตอร์ โดยเลือกการจ่ายไฟ เป็นแบบ E5V และทำการจ่ายไฟเข้าบอร์ด BLDCXplorer โดยใช้ Adapter



เปิดไฟล์ main.c ที่เปิดได้จาก ชื่อโปรเจค > Application > User > main.c

Project Explorer X

- BrushlessDCMotor (in STM32CubeIDE)
- > Includes
- > Application
  - > Startup
  - > User
    - > aspep.c
    - > main.c
    - > mc\_api.c
    - > mc\_app\_hooks.c
    - > mc\_config\_common.c
    - > mc\_config.c
    - > mc\_configuration\_registers.c
    - > mc\_interface.c
    - > mc\_math.c
    - > mc\_parameters.c
    - > mc\_tasks\_sixstep.c
    - > mc\_tasksc.c
    - > mcpc\_config.c
    - > mcp.c
    - > motorcontrol.c
    - > pwm\_common.c
    - > register\_interface.c
    - > regular\_conversion\_manager.c
    - > stm32\_mc\_common\_it.c
    - > stm32g4xx\_hal\_msp.c
    - > stm32g4xx\_it.c
    - > stm32g4xx\_mc\_it.c
    - > syscalls.c
    - > sysmem.c
    - > usart\_aspep\_driver.c
  - > Drivers
  - > Middlewares
    - > BrushlessDCMotor.ioc
    - STM32G474RETX.FLASH.Id
    - STM32G474RETX.RAM.Id

main.c

```
/* USER CODE BEGIN Header */
=====
* @file           : main.c
* @brief          : Main program body
=====
* @attention
=====
* Copyright (c) 2024 STMicroelectronics.
* All rights reserved.
*
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
=====
*/
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes --*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef ---*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */

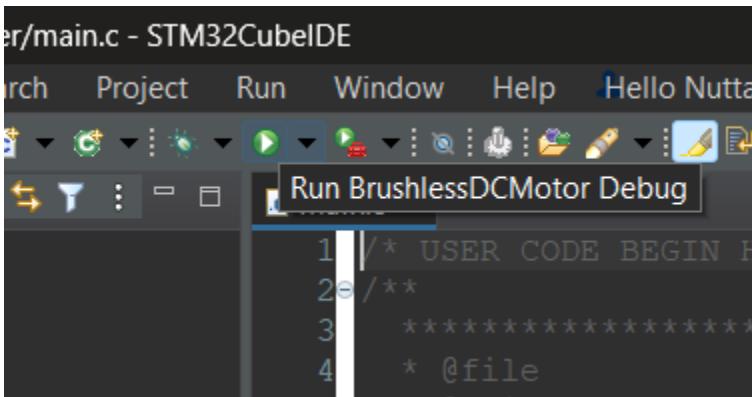
/* USER CODE BEGIN Header */
=====
* @file           : main.c
* @brief          : Main program body
=====
* @attention
=====
* Copyright (c) 2024 STMicroelectronics.
* All rights reserved.
*
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
=====
*/
/* USER CODE END Header */
/* Includes -----*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef ---*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */

/* Private defines */
=====
```

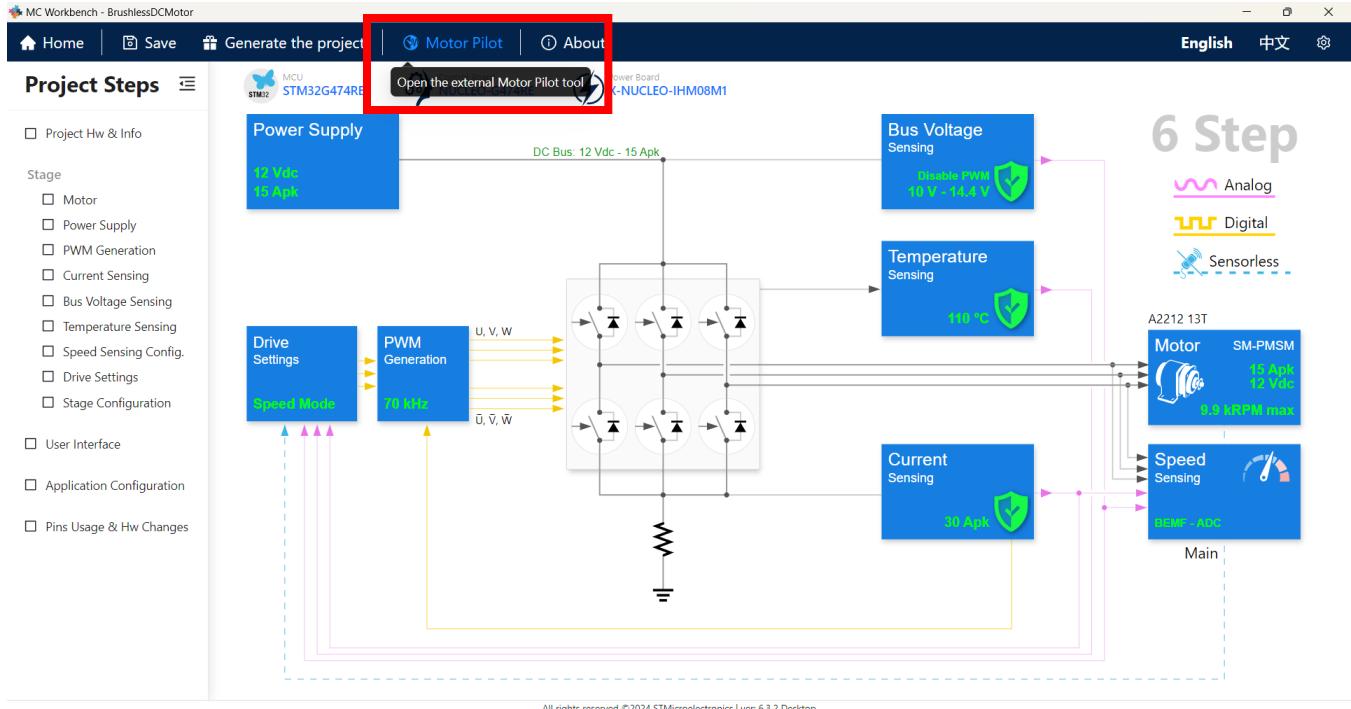
Outline X Build Targets

  - main.h
  - hadc1: ADC\_HandleTypeDefDef
  - hadc2: ADC\_HandleTypeDefDef
  - htim1: TIM\_HandleTypeDefDef
  - htim2: TIM\_HandleTypeDefDef
  - huart2: UART\_HandleTypeDefDef
  - hdma\_usart2\_rx: DMA\_HandleTypeDefDef
  - hdma\_usart2\_tx: DMA\_HandleTypeDefDef
  - SystemClock\_Config(void): void
  - MX\_GPIO\_Init(void): void
  - MX\_DMA\_Init(void): void
  - MX\_ADC1\_Init(void): void
  - MX\_ADC2\_Init(void): void
  - MX\_TIM1\_Init(void): void
  - MX\_TIM2\_Init(void): void
  - MX\_USART2\_UART\_Init(void): void
  - MX\_NVIC\_Init(void): void
  - main(void): int
  - SystemClock\_Config(void): void
  - MX\_NVIC\_Init(void): void
  - MX\_ADC1\_Init(void): void
  - MX\_ADC2\_Init(void): void
  - MX\_TIM1\_Init(void): void
  - MX\_TIM2\_Init(void): void
  - MX\_USART2\_UART\_Init(void): void
  - MX\_DMA\_Init(void): void
  - MX\_GPIO\_Init(void): void

แล้วทำการกด Run code



เปิดโปรแกรม Motor Pilot ผ่าน MC Workbench

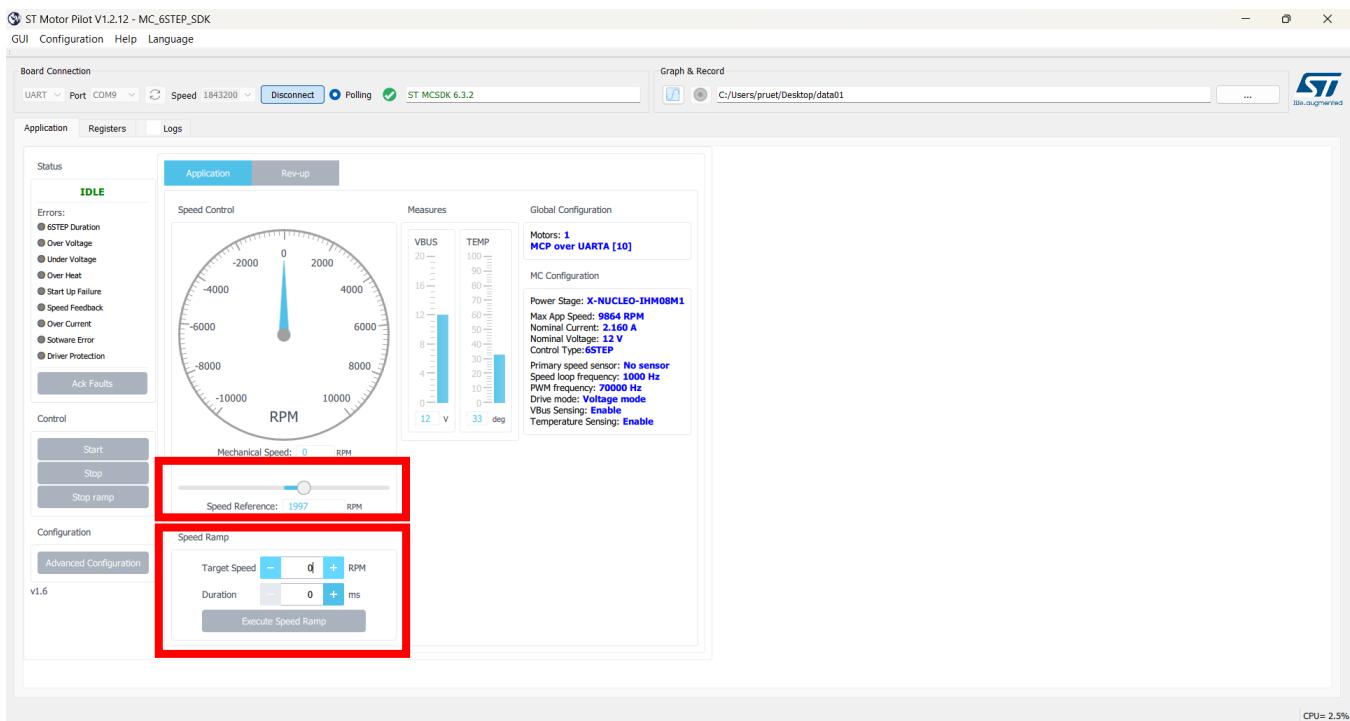


เมื่อเปิดโปรแกรม Motor Pilot ขึ้นมาครั้งแรก จะต้องทำการเลือกหน้าจอโดยการกดปุ่ม Discover Board  
โปรแกรมจะทำการสแกนพอร์ต USB ที่เชื่อมต่ออยู่ และทำการเชื่อมต่ออัตโนมัติ

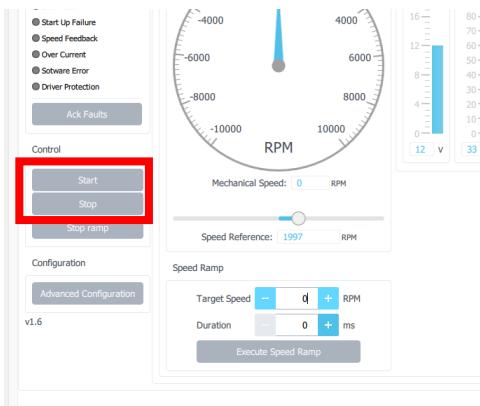


## 2. การควบคุมการทำงานของ Brushless DC Motor ผ่านโปรแกรม ST Motor Pilot

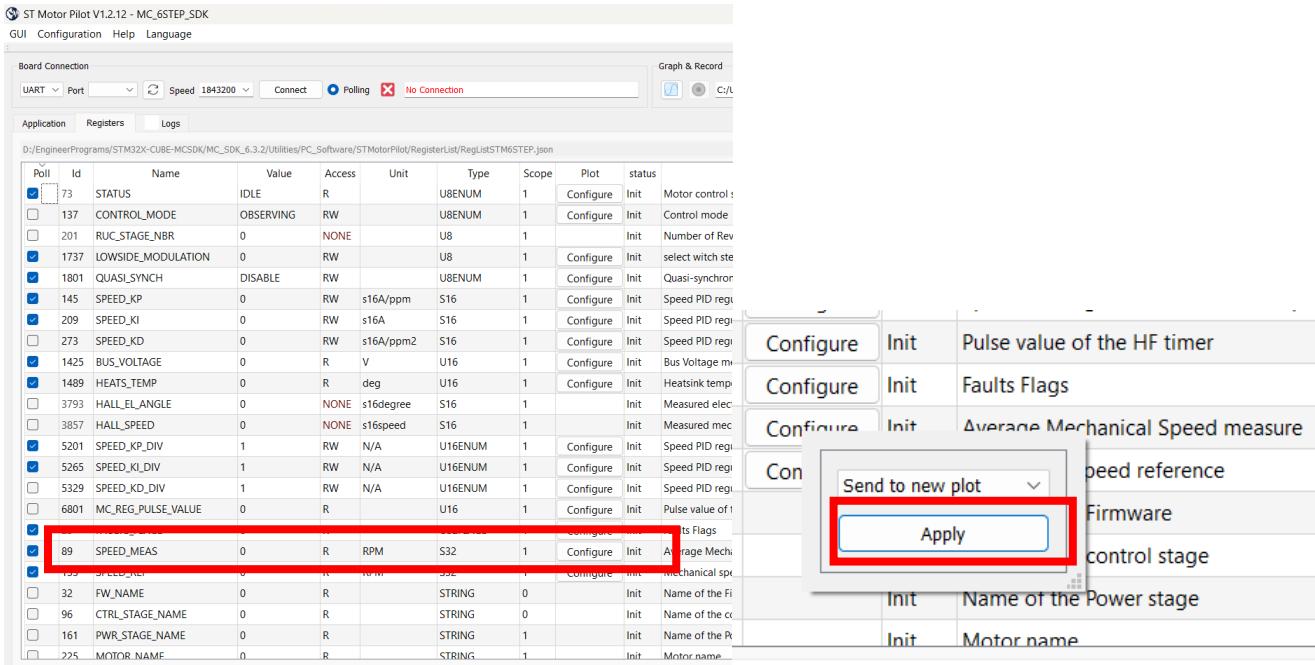
เมื่อเชื่อมต่อบอร์ด Nucleo STM32G474RE เข้ากับคอมพิวเตอร์ ภายในโปรแกรม Motor Pilot เราจะสามารถตั้งค่าความเร็วรอบของ Brushless DC Motor ที่ต้องการได้ผ่านการใช้ Speed Slider ที่เป็นการใช้ Slider เพื่อเลือกความเร็วรอบในหน่วย RPM (รอบต่อนาที) ที่ต้องการให้ Brushless DC Motor ทำงาน หรือ การตั้งค่าผ่าน Speed Ramp ที่เราสามารถกรอกความเร็วรอบในหน่วย RPM (รอบต่อนาที) ที่ต้องการและกดปุ่ม Execute Speed Ramp เพื่อให้ Brushless DC Motor ทำงานตามความเร็วรอบที่ตั้งค่า ซึ่งความเร็วรอบของ Brushless DC Motor จะค่อย ๆ เปลี่ยนแปลงจนถึงค่าความเร็วรอบที่กำหนดในช่วงเวลาหนึ่ง โดยเราสามารถเปลี่ยนค่าความเร็วรอบของ Brushless DC Motor ระหว่างที่ Brushless DC Motor หมุนอยู่ได้ แต่จะไม่สามารถเปลี่ยนทิศทางการหมุนได้



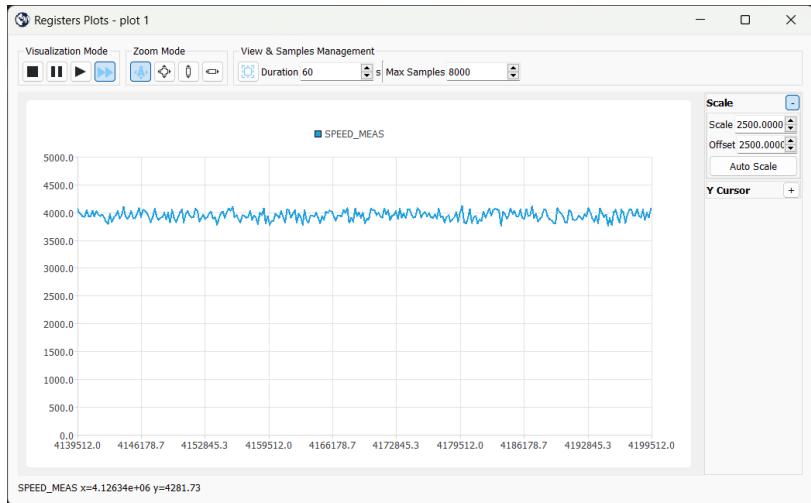
โดยการ กดปุ่ม Start ในโปรแกรม Motor Pilot เพื่อเริ่มการทำงานของ Brushless DC Motor หรือ กดปุ่ม Stop ในโปรแกรม Motor Pilot เพื่อหยุดการทำงานของ Brushless DC Motor



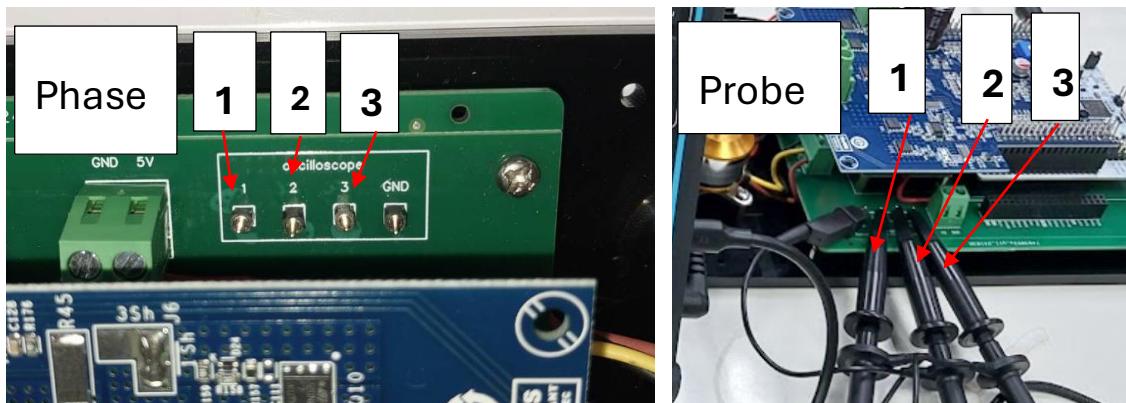
เราสามารถสังเกตค่าความเร็วของปัจจุบันของ Brushless DC Motor ได้ผ่านการเปิดหน้า Registers และกดปุ่ม Configure ในหัวข้อ SPEED\_MEAS จากนั้นกดปุ่ม Apply เพื่อติดกราฟความเร็วของ Brushless DC Motor ที่โปรแกรม Motor Pilot อ่านได้



จะได้กราฟความเร็วดังรูปด้านล่าง



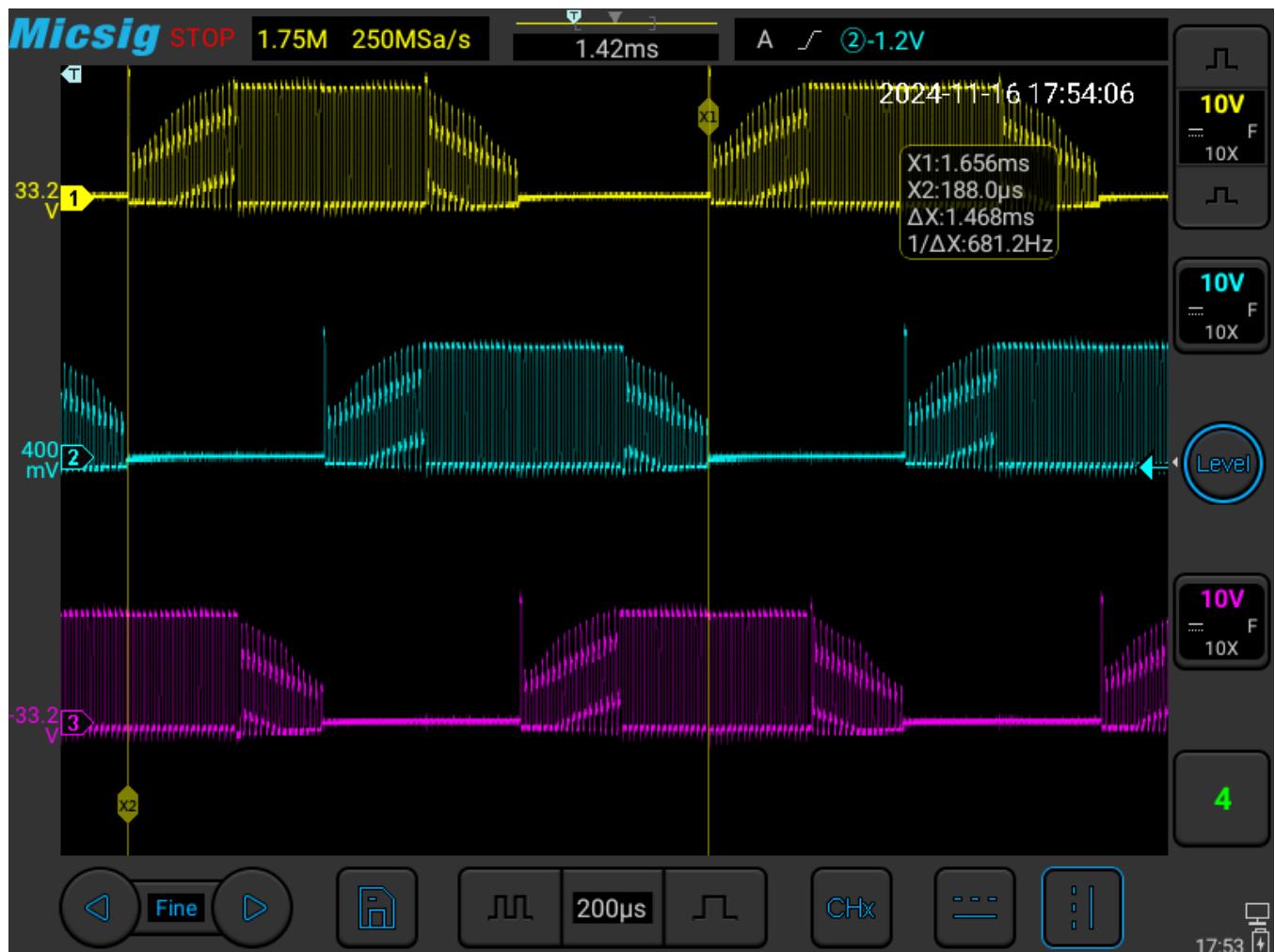
3. วิธีการใช้ Oscilloscope เพื่ออ่านค่าสัญญาณ Back-EMF ของ Brushless DC Motor ขณะที่กำลังหมุนอยู่ เริ่มด้วยการใช้ Probe ของ Oscilloscope เชื่อมต่อกับขั้วของขดลวดมอเตอร์ ที่มีขดลวด 3 Phase โดยนำ Probe ที่ 1 เชื่อมกับ ขดลวด Phase ที่ 1 , Probe ที่ 2 เชื่อมกับ ขดลวด Phase ที่ 2 และ Probe ที่ 3 เชื่อมกับ ขดลวด Phase ที่ 3 ดังรูป



ทำการตั้งค่าการวัดบน Oscilloscope เลือกโหมดการแสดงผลโดยตั้ง Oscilloscope ให้แสดงผลในโหมด AC Coupling เพื่อเน้นแรงดัน Back-EMF, ตั้งค่าช่วงแรงดัน Voltage Scale (V/div) ให้เหมาะสมกับสัญญาณที่คาดว่าจะได้ ( $\pm 10V$  เนื่องจาก แรงดันสูงสุดที่ Brushless DC Motor ได้รับคือ  $12V$ ), ปรับตั้งค่าทริกเกอร์ (Trigger) เพื่อจับสัญญาณให้มั่นคง (Trigger Mode: Edge, Rising หรือ Falling) และ การปรับตั้งค่าช่วงเวลา Time Scale (Time/Div) ให้เห็นเป็นรูปคลื่นชัดเจนซึ่งจะต้องทำการปรับตั้งค่าใหม่ทุกรอบที่ทำการเปลี่ยนความเร็วรอบของ Brushless DC Motor โดยกราฟสัญญาณ Back-EMF ควรเป็นสัญญาณคลื่นไอน์ (Sine Wave) หรือ คลื่นสี่เหลี่ยม คางหมู (Trapezoidal) ดังรูปด้านล่าง



4. การบันทึกค่าสัญญาณ Back-EMF จาก Oscilloscope ทำการสั่งงานให้ Brushless DC Motor ในโปรแกรม Motor Pilot ให้หมุนในความเร็วที่แตกต่างกัน เริ่มจากความเร็วสูงสุดที่โปรแกรม Motor Pilot สามารถสั่งงาน Brushless DC Motor ได้ จากนั้นลดความเร็วลง 20% จน Brushless DC Motor หยุดหมุน โดยในทุกความเร็วทำการบันทึกสัญญาณที่อ่านได้จาก Oscilloscope โดยใช้ฟังก์ชัน Measure ของ Oscilloscope เพื่อวัดความถี่ของสัญญาณ Back-EMF, วัดค่าแรงดันยอด (Peak Voltage) ของสัญญาณ Back-EMF, วัดสัญญาณทั้ง 3 Phase ของ Brushless DC Motor พร้อมกัน ตรวจสอบความสมดุลและ Phase-shift ของคลื่นแต่ละ Phase ตัวอย่างค่าที่ได้จากการวัดความถี่ของสัญญาณ Back-EMF โดยตั้งค่าความเร็วรอบเท่ากับ 5,918.4 RPM ในโปรแกรม Motor Pilot หรือคิดเป็น 40% ของความเร็วรอบของ Brushless DC Motor ที่สามารถปรับได้จากโปรแกรม Motor Pilot



5. การคำนวณหาความเร็วของ Brushless DC Motor จากการอ่านค่าความถี่ของกราฟ Back-EMF ใน Oscilloscope โดยใช้สมการความสัมพันธ์ระหว่าง ความถี่ของสัญญาณ Back-EMF และ ความเร็วของมอเตอร์ Brushless DC Motor หรือ สมการความเร็วซิงโครนัส synchronous speed ( $N_s$ ) โดยนำค่าความถี่ของสัญญาณ Back-EMF ไปคำนวณหาความเร็วของ Brushless DC Motor ในช่วงต่าง ๆ เพื่อดูความสัมพันธ์ระหว่างความเร็วของที่คำนวณจาก ความถี่ของสัญญาณ Back-EMF และความเร็วของที่วัดได้จากในโปรแกรม Motor Pilot ผ่านการใช้เครื่องมืออย่าง MATLAB

เนื่องจากข้อมูลค่าความถี่ของสัญญาณ Back-EMF ทั้ง 3 Phase ของ Brushless DC Motor ที่เก็บมาไม่เท่ากัน จึงต้องนำทั้ง 3 ค่า มาหาค่าเฉลี่ยก่อน เมื่อได้ความถี่เฉลี่ยแล้วนำไปแทนค่าในสมการความเร็วซิงโครนัส synchronous speed ( $N_s$ ) ซึ่งจะได้เป็นความเร็วของที่เกิดจากการคำนวณความถี่ของ Back-EMF นั้นเอง

6. การวิเคราะห์ ความเร็วของ Brushless DC Motor ในโปรแกรม กับ ความเร็วของที่เกิดจากการคำนวณ ความถี่ของสัญญาณ Back-EMF ของ Brushless DC Motor โดยใช้การหาค่า เปรอร์เซ็นต์ความคลาดเคลื่อน

(%Error) ที่แสดงความแตกต่างระหว่าง ค่าความเร็วrob จาก Back EMF กับ ความเร็วrob ที่อ่านได้จากในโปรแกรม Motor Pilot โดยแสดงผลในรูปของเปอร์เซ็นต์เพื่อให้เข้าใจง่ายขึ้น

$$\%Error = \frac{|\text{ความเร็วrob ที่จาก Motor Pilot} - \text{ค่าความเร็วrob จาก Back EMF}|}{\text{ค่าความเร็วrob จาก Back EMF}} \times 100$$

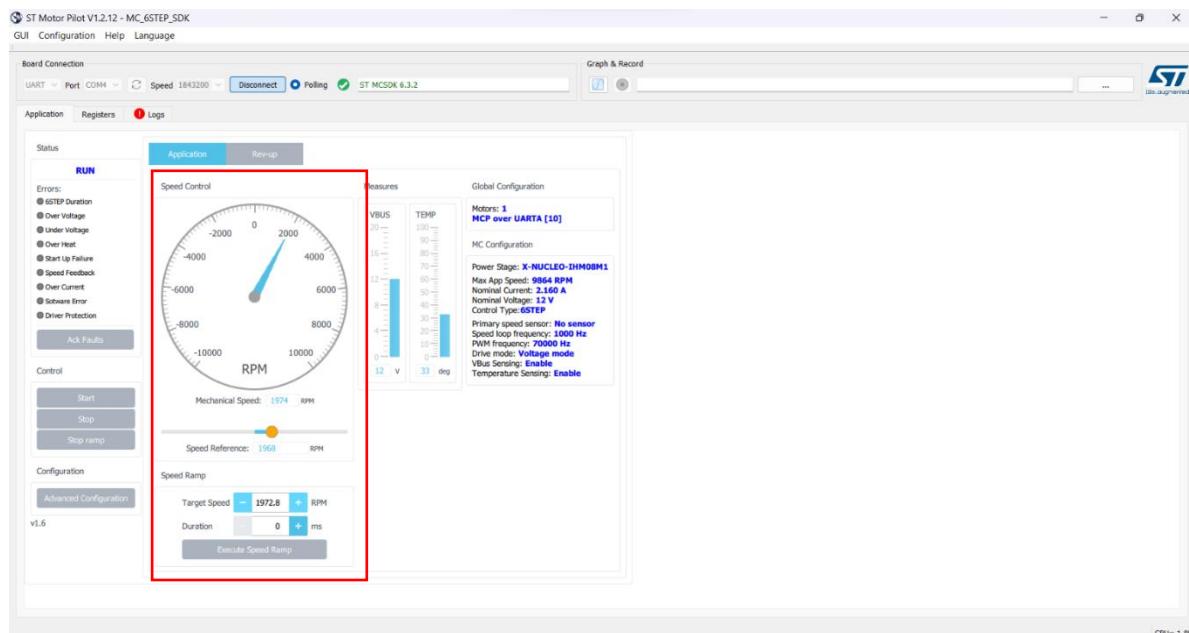
## ผลการทดลอง

ได้มีการแบ่งการเก็บผลการทดลองออกเป็นทั้งหมด 10 ช่วง โดยแบ่งออกเป็น

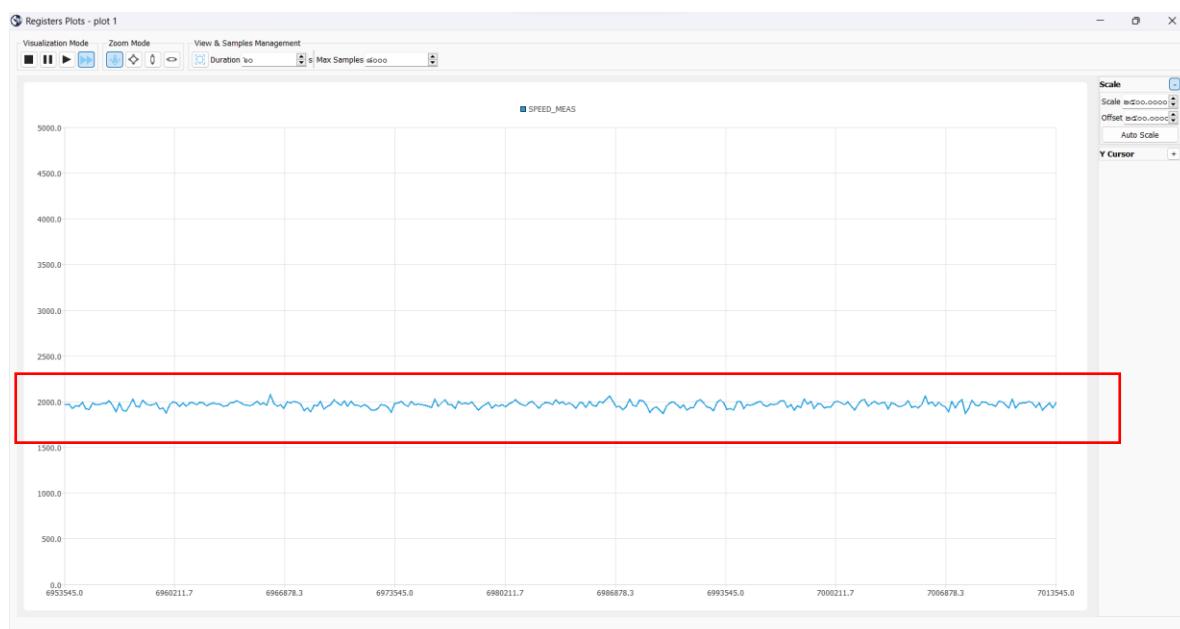
- ทางบวก ได้แก่ 20%, 40%, 60%, 80% และ 100% ของ Max App Speed
  - ทางลบ ได้แก่ -20%, -40%, -60%, -80% และ -100% ของ Max App Speed
- ซึ่ง Max App Speed มีค่า 9,864 RPM

โดยได้ผลการทดลอง ดังนี้

1. กำหนด Target Speed เป็น 1,972.8 RPM ซึ่งคิดเป็น 20% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนตั้งกราฟนี้



เมื่อสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

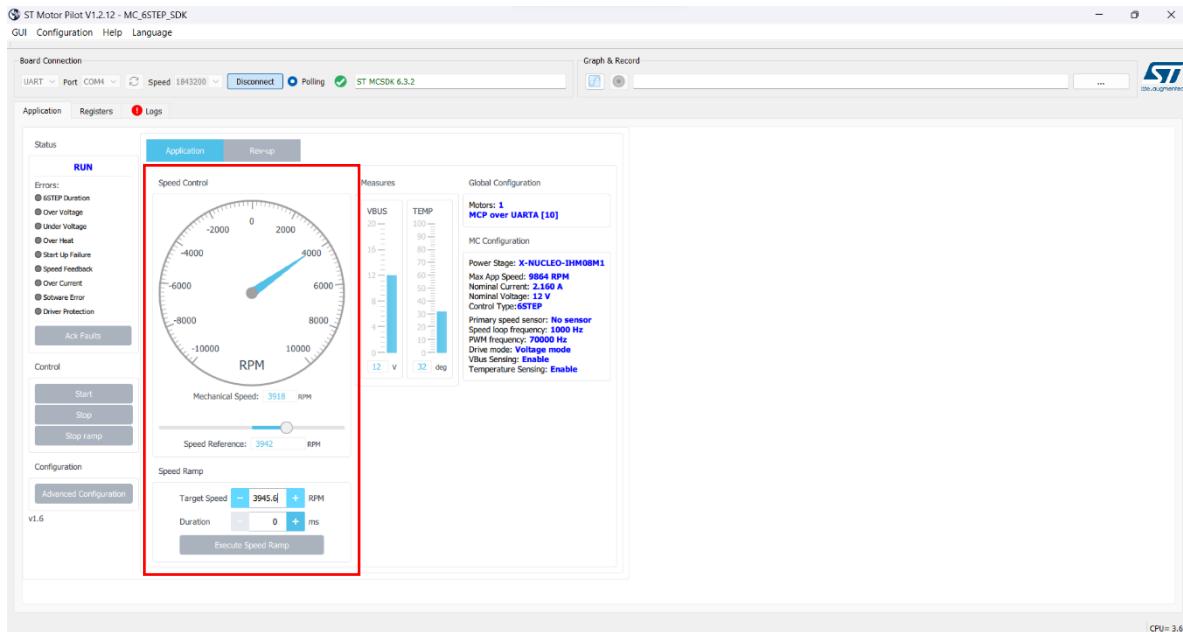


สัญญาณ	ความถี่ (Hz)
U	241.5
V	228.3
W	229.4
ค่าเฉลี่ย	233.1

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาค่านวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

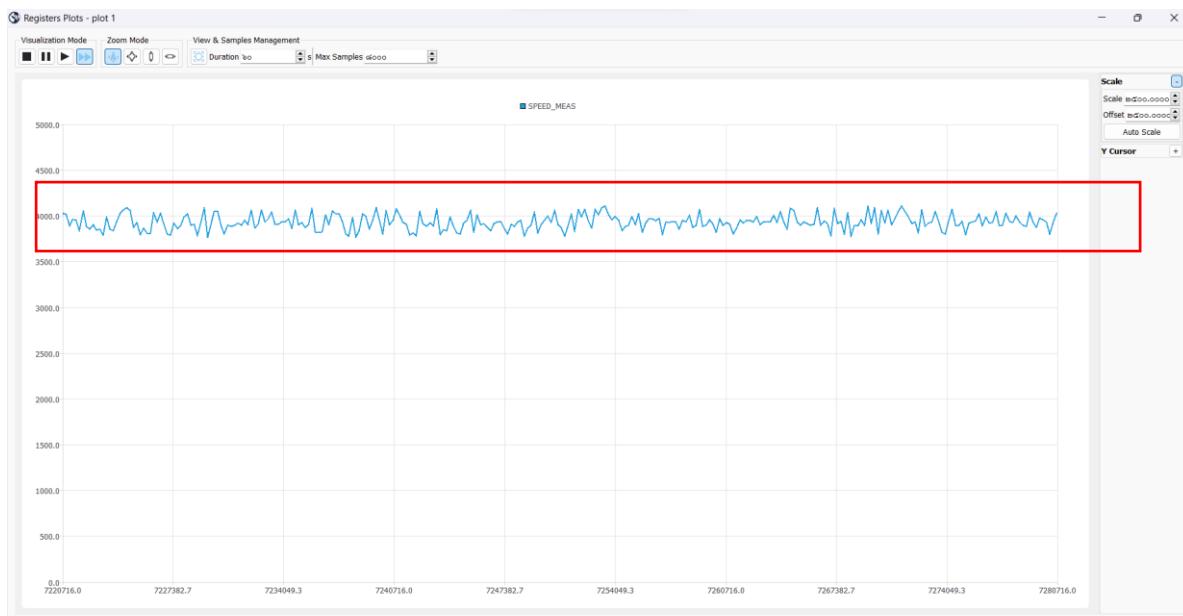
ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
1,957.1	1,972.8	0.8%

## 2. ตั้งค่า Target Speed เป็น 3,945.6 RPM ซึ่งคิดเป็น 40% ของ Max App Speed



หลังจากที่

เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อต่อสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

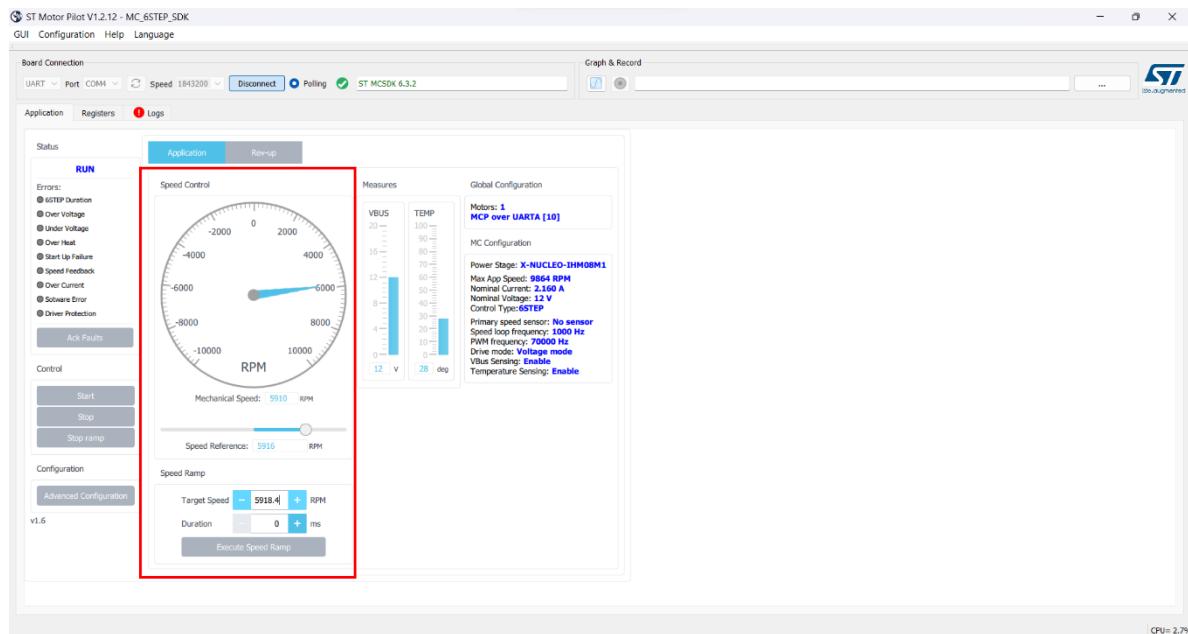


สัญญาณ	ความถี่ (Hz)
U	454.5
V	454.5
W	454.5
ค่าเฉลี่ย	454.5

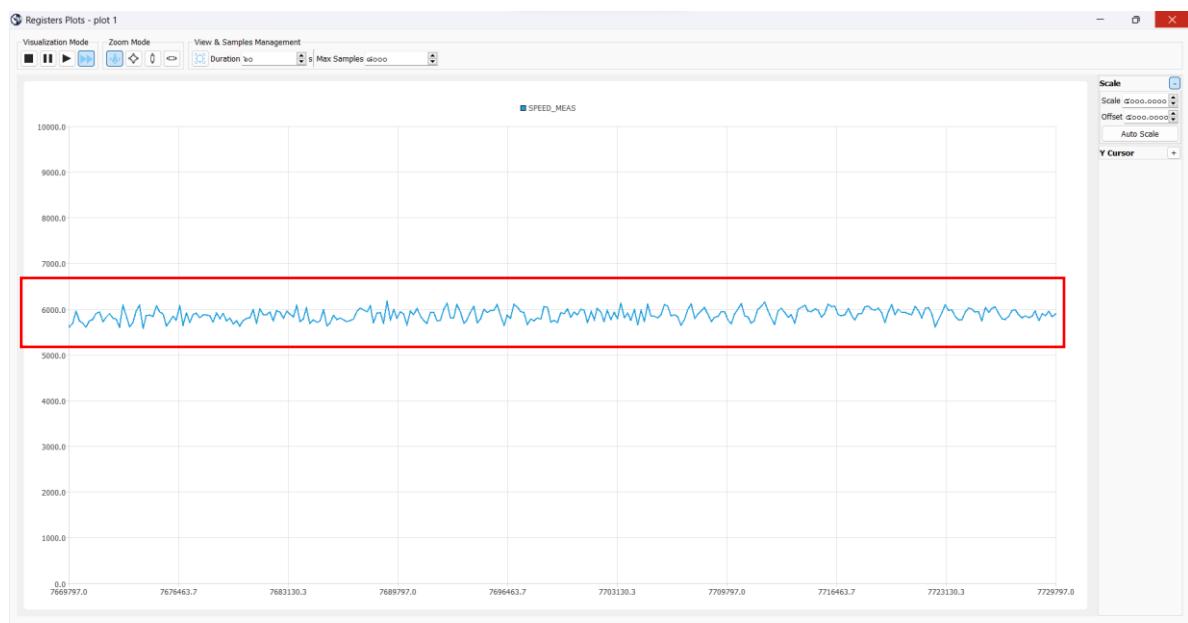
เมื่อได้ค่าเฉลี่ยของความถี่แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คาดเคลื่อน(% Error)
3,895.7	3,945.6	1.3%

### 3. ตั้งค่า Target Speed เป็น 5,918.4 RPM ซึ่งคิดเป็น 60% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อุดสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

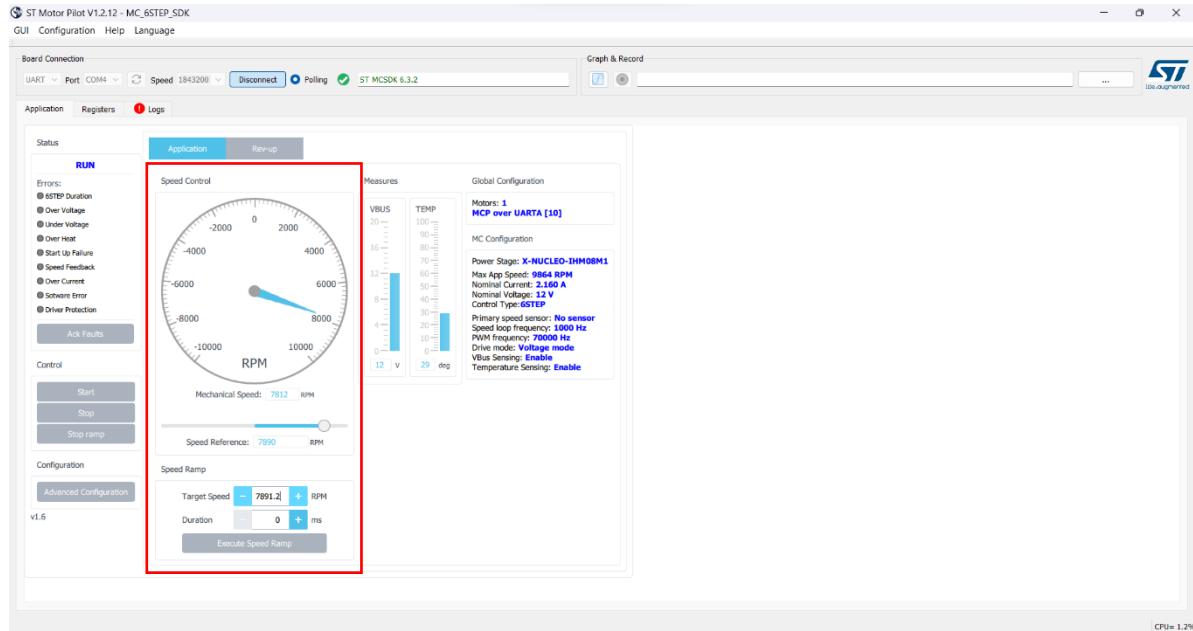


สัญญาณ	ความถี่ (Hz)
U	680.3
V	680.3
W	680.3
ค่าเฉลี่ย	680.3

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คาดเคลื่อน(% Error)
3,895.7	5,918.4	1.5%

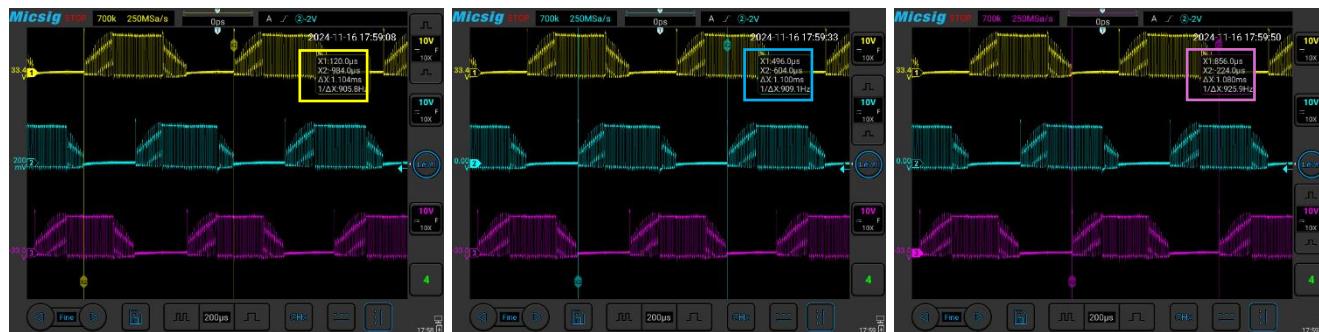
4. ตั้งค่า Target Speed เป็น 7,891.2 RPM ซึ่งคิดเป็น 80% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อดูสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณห้องหมุด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

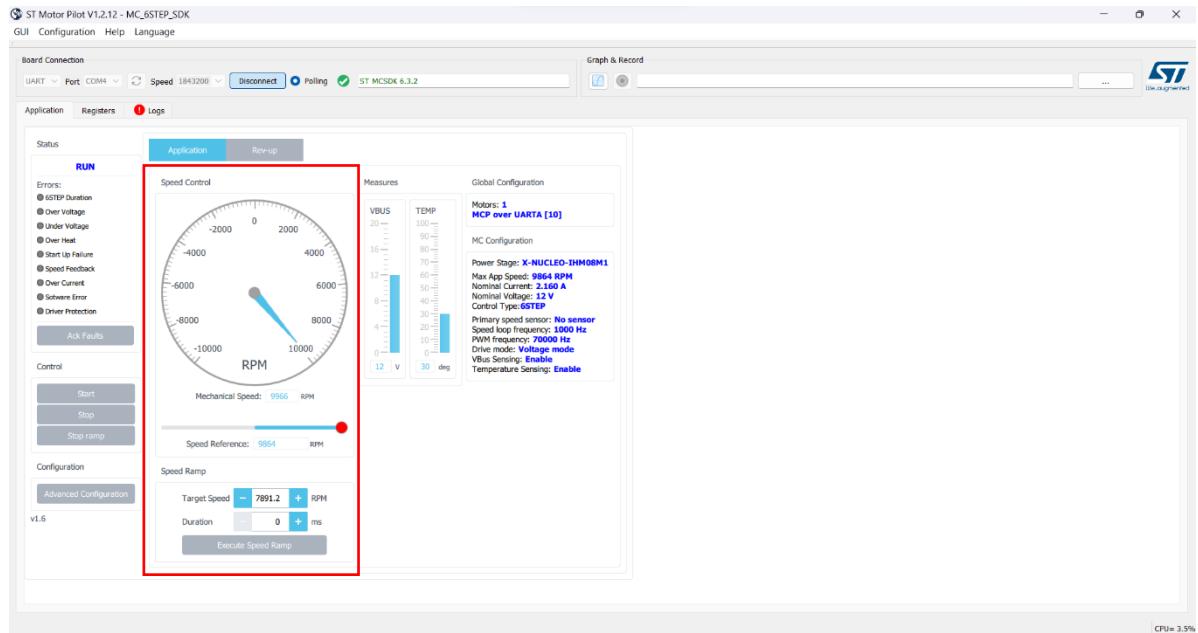


ลักษณะ	ความถี่ (Hz)
U	905.8
V	909.1
W	925.9
ค่าเฉลี่ย	913.6

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาค่านวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
7,830.9	7,891.2	0.8%

## 5. ตั้งค่า Speed Reference เป็น 9,864 RPM ซึ่งคิดเป็น 100% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อคุณสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

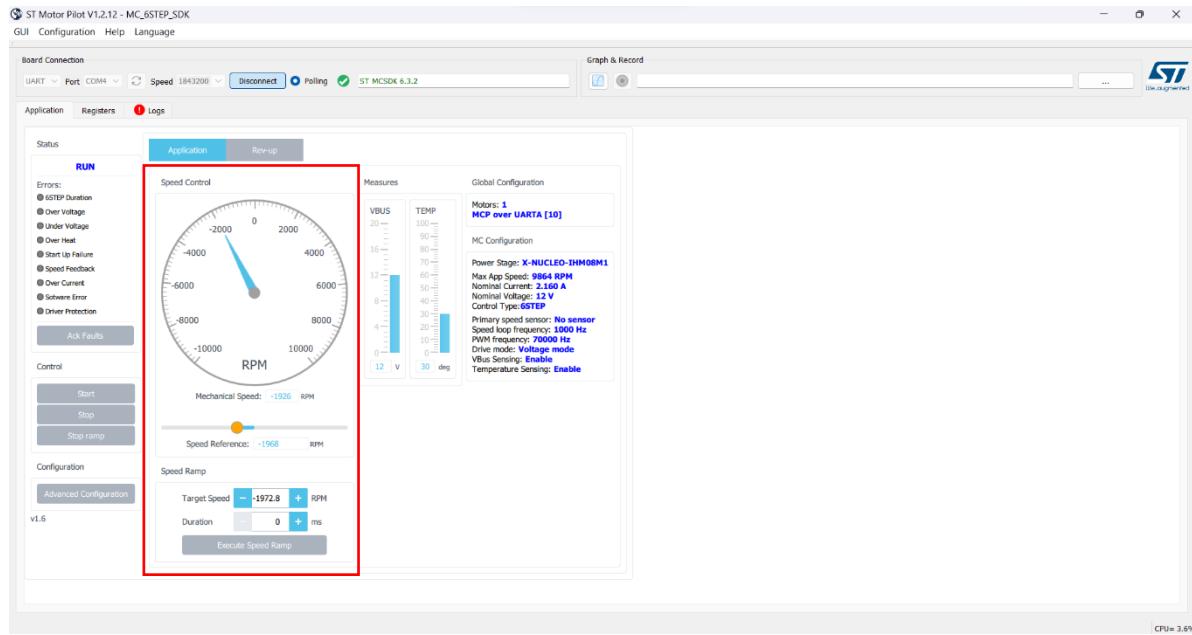


ลักษณะ	ความถี่ (kHz)
U	1.142
V	1.166
W	1.139
ค่าเฉลี่ย	1.149

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาค่านวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
9,848.6	9,864	0.2%

## 6. ตั้งค่า Target Speed เป็น -1,972.8 RPM ซึ่งคิดเป็น -20% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อต้องสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

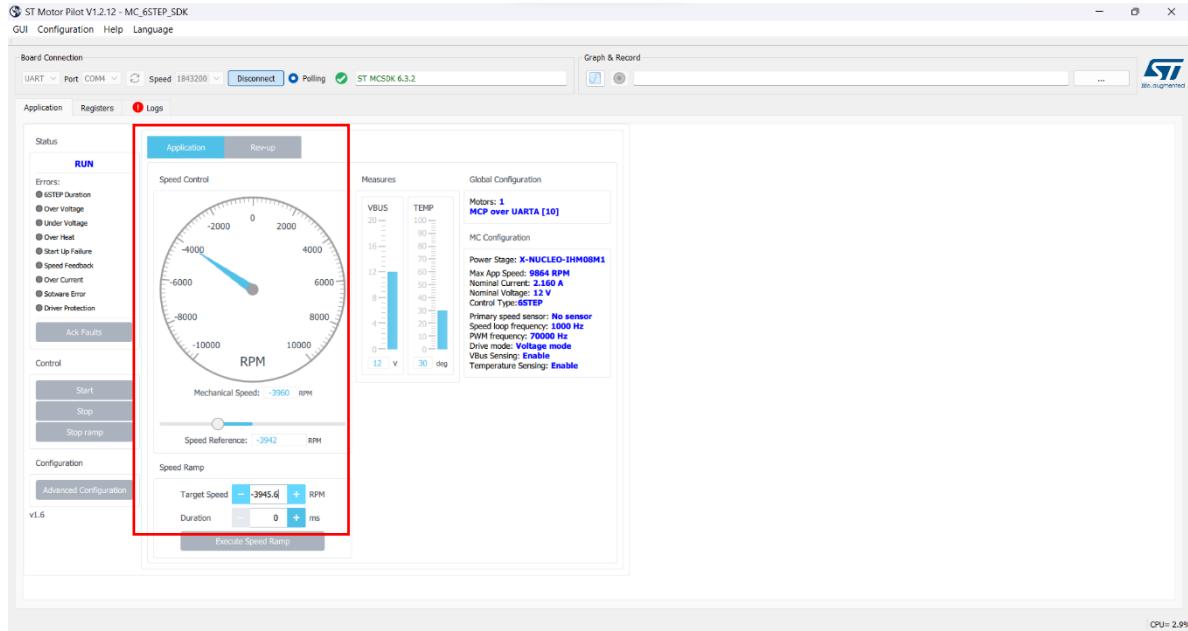


สัญญาณ	ความถี่ (Hz)
U	228.3
V	227.3
W	229.4
ค่าเฉลี่ย	233.1

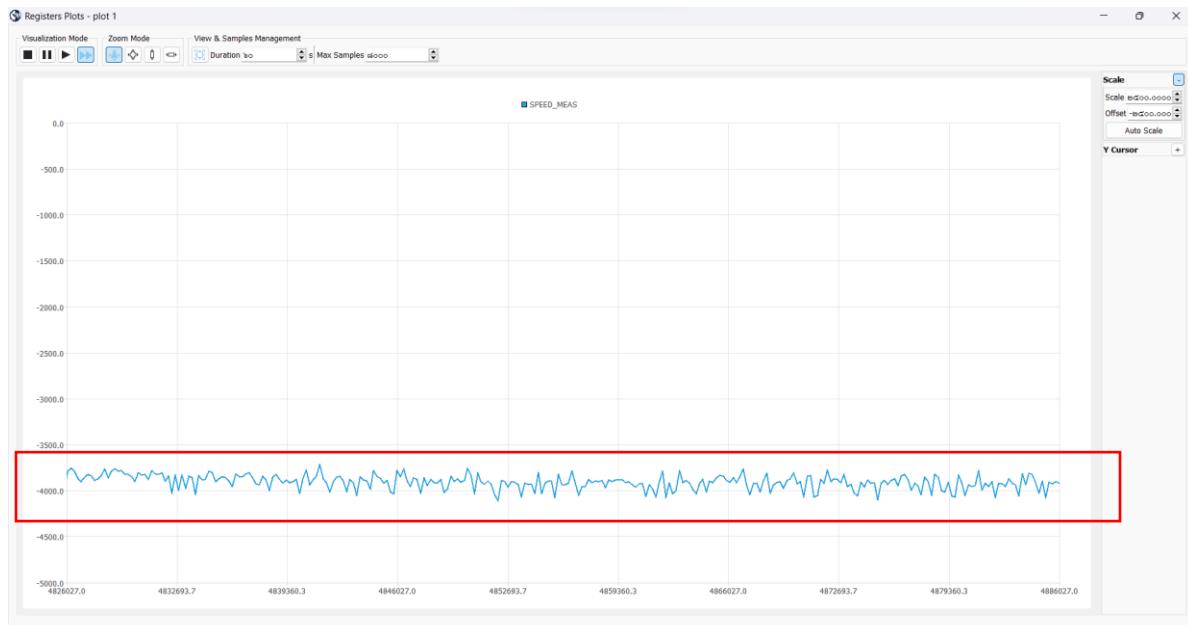
เมื่อได้ค่าเฉลี่ยของความถี่แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
1,997.7	1,972.8	1.2%

7. ตั้งค่า Target Speed เป็น -3,945.6 RPM ซึ่งคิดเป็น -40% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อคุณรันโปรแกรมที่แสดงบน Oscilloscope จะได้สัญญาณห้องหมุด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

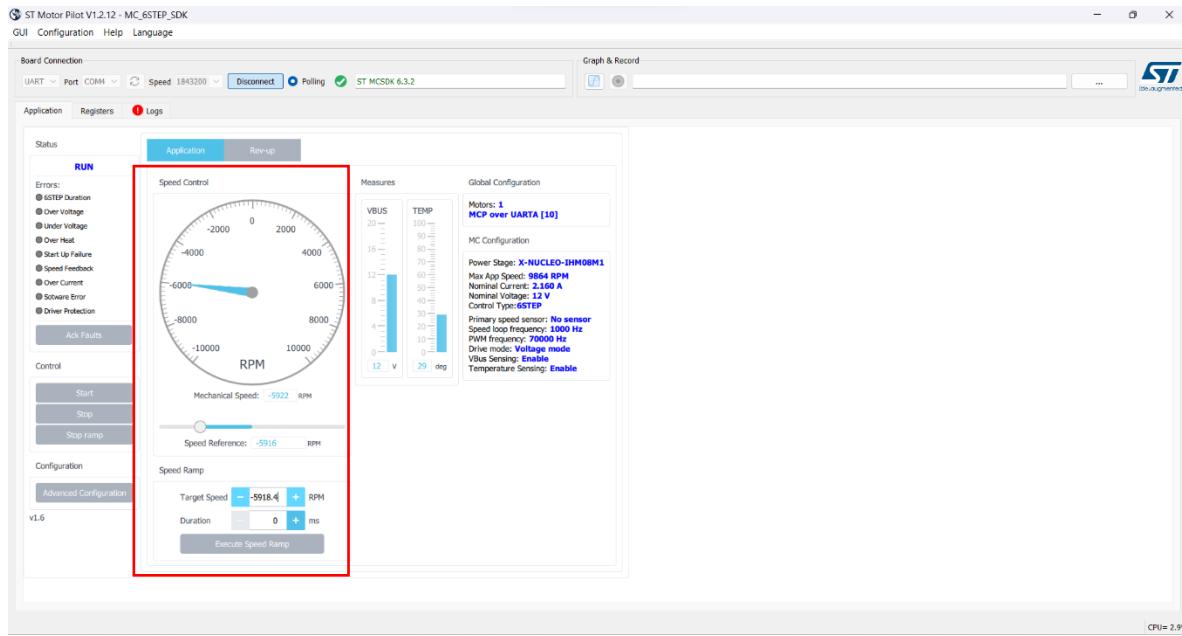


สัญญาณ	ความถี่ (Hz)
U	458.7
V	458.7
W	454.5
ค่าเฉลี่ย	457.3

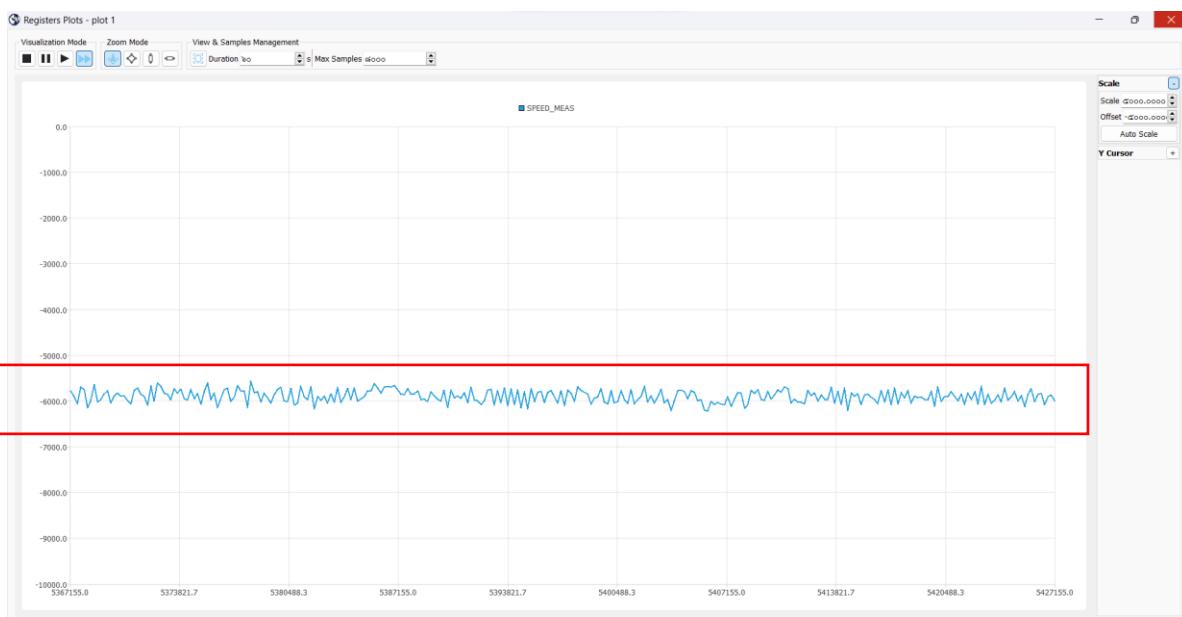
เมื่อได้ค่าเฉลี่ยของความถี่แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
3,919.7	3,945.6	0.7%

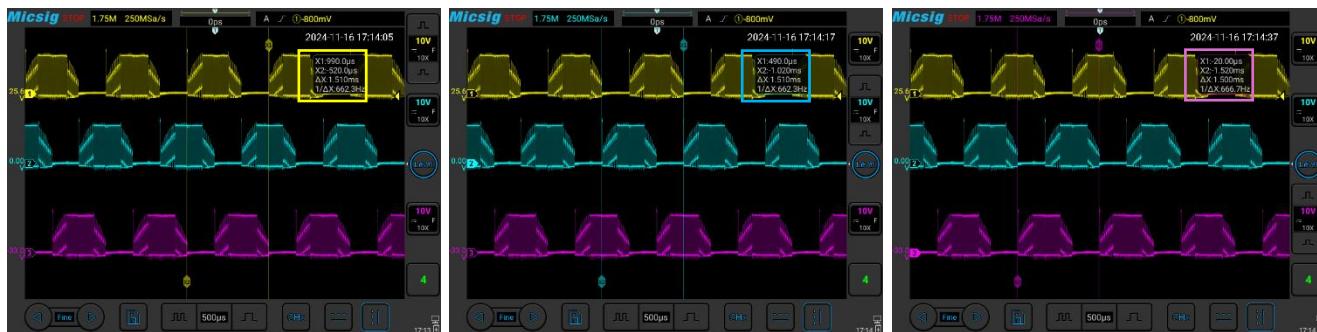
## 8. ตั้งค่า Target Speed เป็น -5,918.4 RPM ซึ่งคิดเป็น -60% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อุดสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

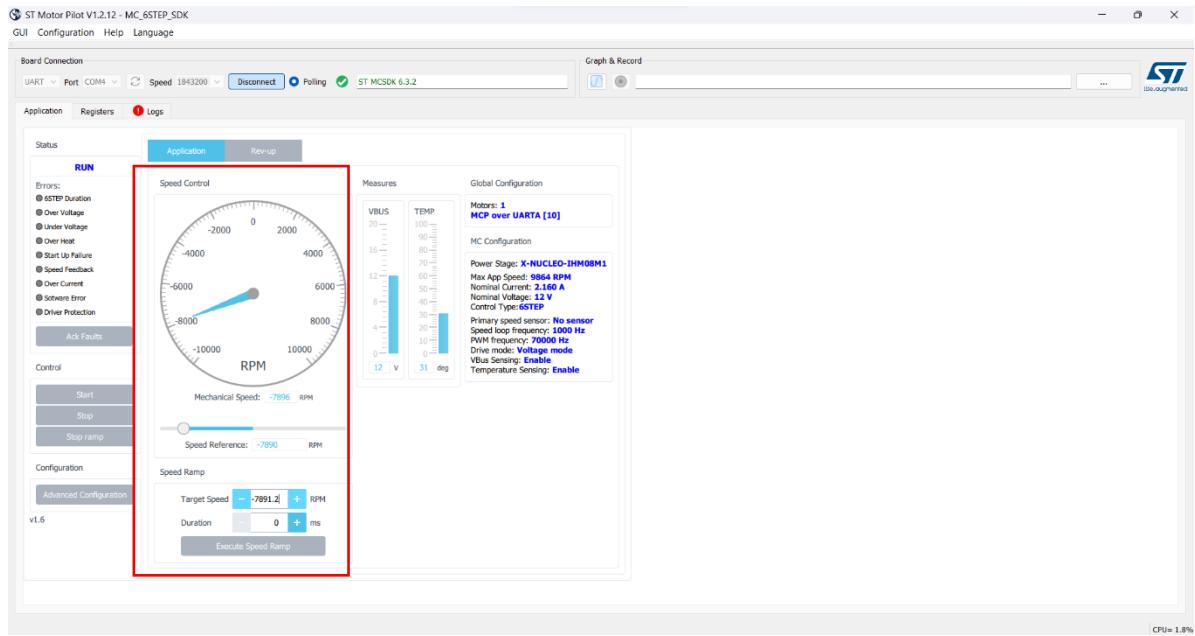


สัญญาณ	ความถี่ (Hz)
U	662.3
V	662.3
W	666.7
ค่าเฉลี่ย	663.8

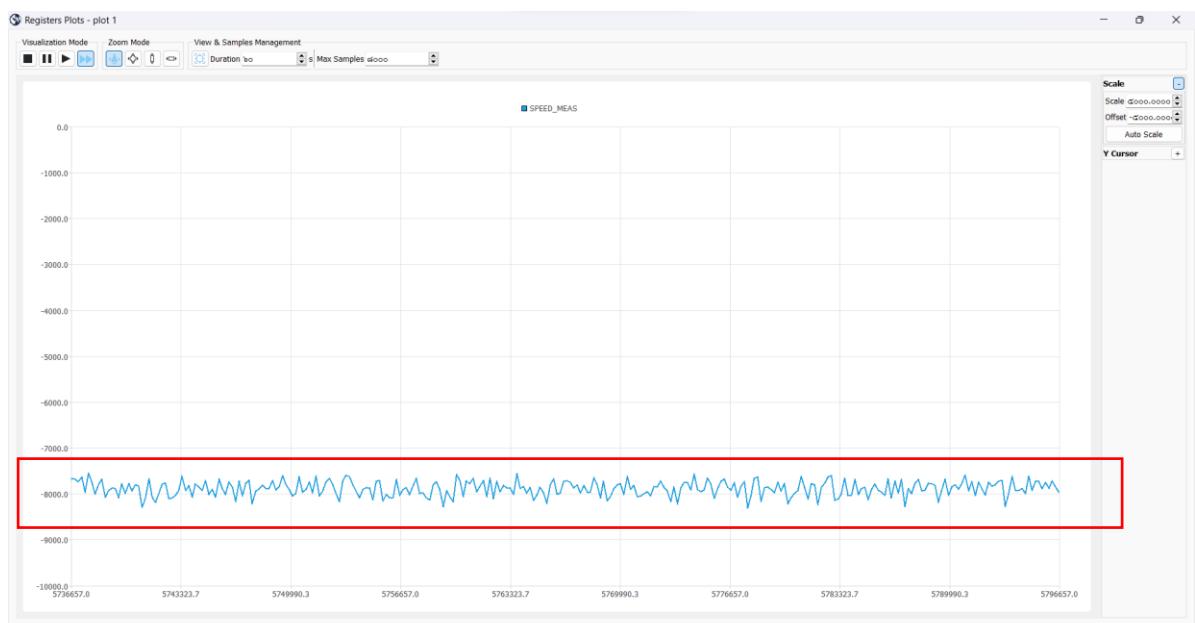
เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
5,689.4	5,918.4	4.0%

9. ตั้งค่า Target Speed เป็น -7,891.2 RPM ซึ่งคิดเป็น -80% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อุดสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

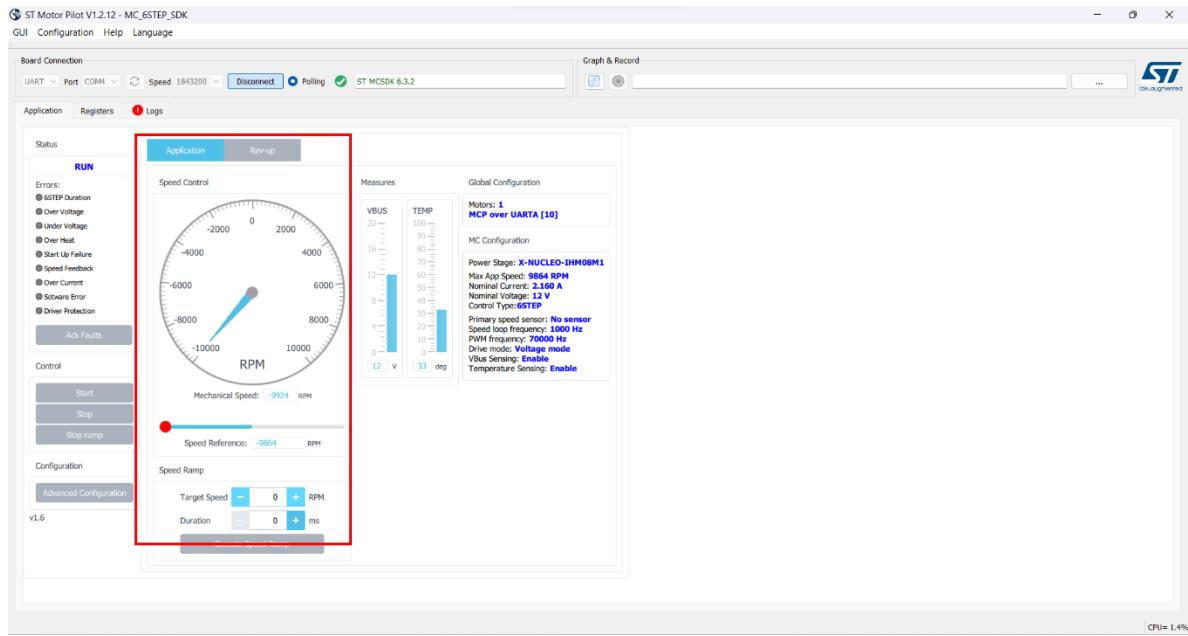


สัญญาณ	ความถี่ (Hz)
U	912.4
V	922.5
W	925.9
ค่าเฉลี่ย	920.3

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
7,888.0	7,891.2	0.04%

## 10. ตั้งค่า Speed Reference เป็น -9,864 RPM ซึ่งคิดเป็น -100% ของ Max App Speed



หลังจากที่เปลี่ยนค่าของความเร็วแล้ว จะได้ค่าของความเร็วรอบ (RPM) ที่เปลี่ยนดังกราฟนี้



เมื่อุดสัญญาณที่แสดงบน Oscilloscope จะได้สัญญาณทั้งหมด 3 Phase ได้แก่ สัญญาณ U, สัญญาณ V และสัญญาณ W และได้ค่าของความถี่ ดังนี้

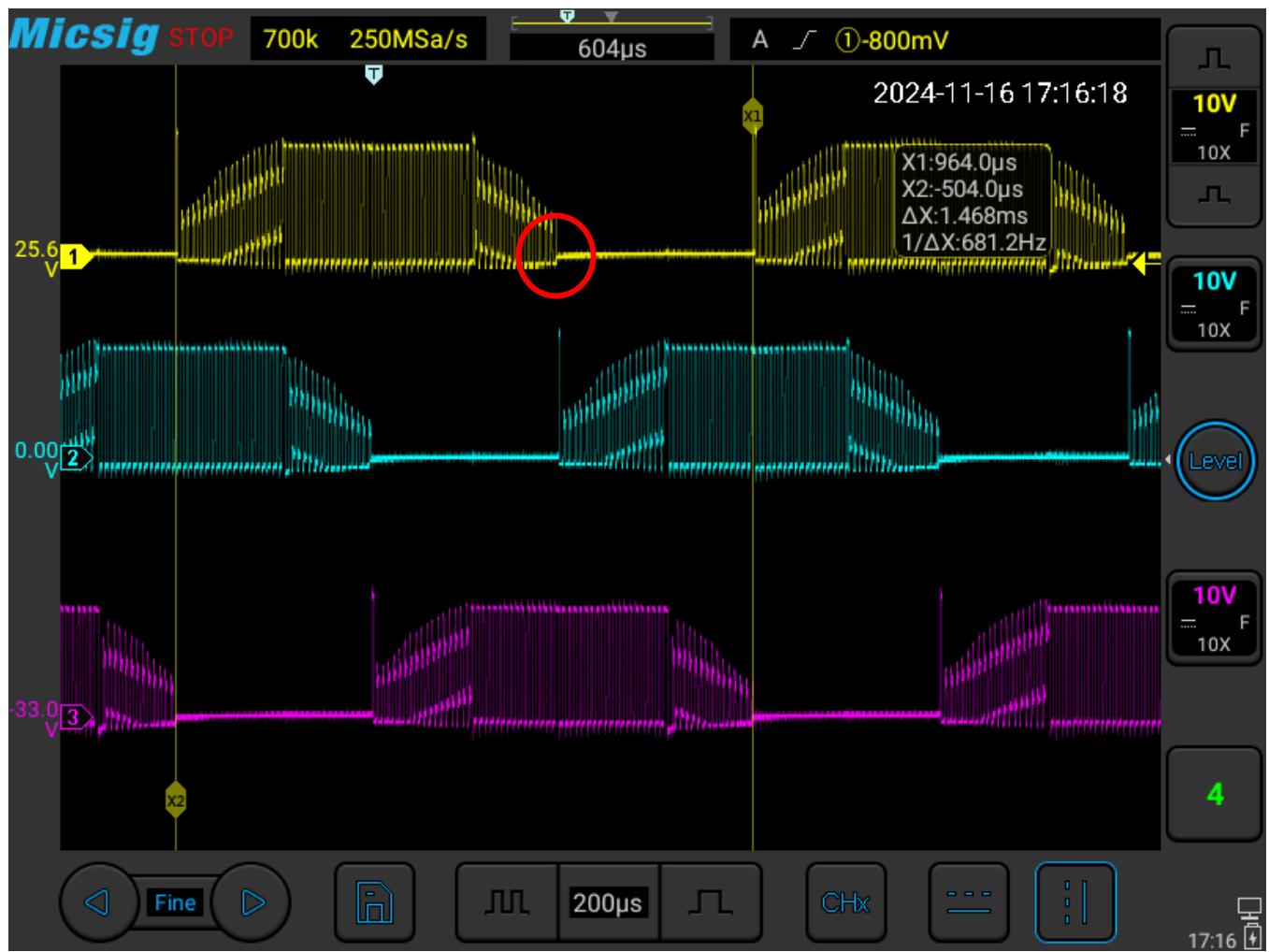


สัญญาณ	ความถี่ (kHz)
U	1.136
V	1.142
W	1.166
ค่าเฉลี่ย	1.148

เมื่อได้ ค่าเฉลี่ยของความถี่ แล้วนำไปหาคำนวนหา ความเร็วรอบในหน่วย RPM ได้ดังนี้

ความเร็วรอบ Back-EMF	ความเร็วรอบ ในโปรแกรม	ความเร็วรอบที่คลาดเคลื่อน(% Error)
9,840	9,864	0.2%

การดูส่วนอื่นๆของกราฟ เพื่อดู ลักษณะของสัญญาณ Back-EMF, การตรวจจับจุด Zero-Crossing (ZC), ความสมดุลของเฟสทั้ง 3 เฟส



โดยจาก การทดลอง จะเห็นได้จากรูปว่า ลักษณะของสัญญาณ Back-EMF เป็น สี่เหลี่ยมคงที่  
จะเห็น จุด Zero-Crossing (ZC) ที่สี่เหลี่ยม สีแดงในรูป

## สรุปผลการทดลอง

- ความเร็วรอบที่ได้จาก Back-EMF ใกล้เคียงกับค่าความเร็วเป้าหมาย โดยมีค่า % Error เฉลี่ยอยู่ในช่วง 0.04% ถึง 4.0% ซึ่งอยู่ในระดับที่ยอมรับได้
- สัญญาณที่วัดได้จาก Oscilloscope ในทั้ง 3 เฟส (U, V, W) มีความถี่ที่สมดุล โดยค่าเฉลี่ยของความถี่ถูกนำไปคำนวณเป็นความเร็วรอบในหน่วย RPM
- ลักษณะของสัญญาณ Back-EMF ที่วัดได้แสดงเป็นรูปทรงสี่เหลี่ยมคงที่มีจุด Zero-Crossing (ZC) ชัดเจน

## อภิปรายผล

กราฟสัญญาณของ BLDC Motor ทั้ง 3 Phase (U, V, W) มีลักษณะที่สะท้อนถึงเทคนิคการควบคุมที่ใช้ โดยกราฟแสดงความสมมาตรและ Phase Shift 120° ในช่วงความเร็วปานกลางถึงสูง เนื่องจาก Commutation Sequence ที่แม่นยำ สำหรับเทคนิค Trapezoidal Control กราฟ Back EMF แสดงเป็นรูปสี่เหลี่ยมคงที่ ซึ่งหมายความว่าการกระตุนแบบ 6-Step Commutation ในขณะที่ Sinusoidal Control และ Field-Oriented Control (FOC) ทำให้กราฟมีลักษณะที่ราบรื่นและสมดุลมากขึ้น โดยเฉพาะใน FOC ที่ใช้ Space Vector Modulation (SVM) ซึ่งช่วยลด Torque Ripple และปรับแรงดันในแต่ละเฟสอย่างต่อเนื่อง

ในช่วงความเร็วต่ำ (-20%, -40%) พบร่วมกับ Sensorless Control มีข้อจำกัดในการความสามารถในการตรวจจับ Zero Crossing ของ Back EMF เนื่องจากแรงดันย้อนกลับมีค่าต่ำ ส่งผลให้กราฟสัญญาณอาจมีความผิดเพี้ยนในขณะที่การใช้ Hall Effect Sensors ใน Sensor-based Control แสดงผลที่แม่นยำตั้งแต่เริ่มต้น โดยกราฟ Back EMF ยังคงความสมมาตรแม้ในช่วงความเร็วต่ำหรือการหมุนกลับทิศทาง

Space Vector Modulation (SVM) ช่วยเพิ่มประสิทธิภาพของ BLDC Motor ในช่วงความเร็วสูงและการหมุนกลับทิศทาง โดยการปรับแรงดันและกระแสในแต่ละ Phase ให้เหมาะสมผ่าน PWM ซึ่งช่วยลด Torque Ripple และปรับสมดุลของระบบ การใช้ Feedback จาก Back EMF และ Hall Sensors ผ่านการประมวลผลด้วย Op-Amp และ Zero Crossing ยิ่งช่วยเพิ่มความแม่นยำและความเสถียรของการควบคุม

จากการทดลองนี้ สรุปได้ว่าการเลือกใช้เทคนิคการควบคุมที่เหมาะสม เช่น FOC ที่ใช้ SVM หรือการใช้ Hall Sensors มีบทบาทสำคัญในการเพิ่มประสิทธิภาพของ BLDC Motor และลดข้อจำกัดของระบบ Sensorless Control โดยเฉพาะในช่วงความเร็วต่ำและการหมุนกลับทิศทาง

## ข้อเสนอแนะ

ควรหาโปรแกรมควบคุมที่สามารถปรับการส่งค่า Target Speed ที่แม่นยำกว่านี้

## อ้างอิง

1. [https://www.allaboutcircuits.com/technical-articles/sensorless-brushless-dc-bldc-motor-control/#:~:text=Sensorless%20BLDC%20motor%20control%E2%80%94sometimes%20called%20sensorless%20trapezoidal%20control,respect%20to%20the%20motor%27s%20stator%20%28t%20stationary%20part%29.](https://www.allaboutcircuits.com/technical-articles/sensorless-brushless-dc-bldc-motor-control/#:~:text=Sensorless%20BLDC%20motor%20control%E2%80%94sometimes%20called%20sensorless%20trapezoidal%20control,respect%20to%20the%20motor%27s%20stator%20%28t%20stationary%20part%29)
2. [https://mechtex.com/blog/back-emf-in-bldc-motors-a-complete-guide#:~:text=Back-EMF%20is%20a%20natural%20feedback%20mechanism%2C%20that%20helps,to%20allow%20precise%20control%20without%20any%20external%20sensors.](https://mechtex.com/blog/back-emf-in-bldc-motors-a-complete-guide#:~:text=Back-EMF%20is%20a%20natural%20feedback%20mechanism%2C%20that%20helps,to%20allow%20precise%20control%20without%20any%20external%20sensors)
3. [https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/#google\\_vignette](https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/#google_vignette)
4. <https://www.brushless.com/how-to-control-a-brushless-dc-motor>
5. <https://www.electrical4u.com/brushless-dc-motors/>
6. <https://www.electricaltechnology.org/2016/05/bldc-brushless-dc-motor-construction-working-principle.html#what-is-a-brushless-dc-motor-bldc>
7. [https://media.monolithicpower.com/mps\\_cms\\_document/b/r/brushed-dc-and-bldc-motors\\_parameter-operation-drivers\\_r1.0\\_1.pdf](https://media.monolithicpower.com/mps_cms_document/b/r/brushed-dc-and-bldc-motors_parameter-operation-drivers_r1.0_1.pdf)
8. <https://greensky-power.com/th/brushless-dc-motor-bldc-basics/>
9. <https://www.digikey.co.th/th/articles/how-to-power-and-control-brushless-dc-motors>
10. <https://onlinedocs.microchip.com/oxy/GUID-0A4BC4EE-29F5-4736-8125-17139B84E7B5-en-US-2/GUID-C1C929BC-0D12-40FB-B555-F28B23CB457E.html>
11. <https://www.fisherpaykeltechnologies.com/knowledge-hub/what-is-a-permanent-magnet-synchronous-motor>
12. [https://www.rhydolabz.com/documents/26/BLDC\\_A2212\\_13T.pdf](https://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf)
13. <https://www.elprocus.com/what-is-a-permanent-magnet-synchronous-motor-its-working/>
14. [https://www.st.com/resource/en/application\\_note/an4220-sensorless-sixstep-bldc-commutation-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4220-sensorless-sixstep-bldc-commutation-stmicroelectronics.pdf)