

การออกแบบตัวควบคุมพีไอดีสำหรับชุดแทนเลื่อนในการสอบเทียบ  
สายเทปวัดระยะ  
Design PID Controller for Carriage in Measuring Tape Calibration  
System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบตัวควบคุมพีไอดีสำหรับชุดแทนเลื่อนในการสอบเทียบ  
สายเทปวัดระยะ

Design PID Controller for Carriage in Measuring Tape Calibration  
System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Design PID Controller for Carriage in Measuring Tape Calibration System



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING KING MONGKUT'S INSTITUTE OF TECHNOLOGY  
LARDKRABANG  
ACADEMIC YEAR 2020

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2554  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การออกแบบตัวควบคุมพีไอดีสำหรับชุดแทนเลียนในการสอบเทียบ  
สายเทปวัดระยะ

นักศึกษาผู้จัดทำ นายชินรอส มุขโรจน์ รหัสนักศึกษา 60010236  
นายพชรพล อัครพงศ์ไพศาล รหัสนักศึกษา 60010654

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2563

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ดร. อาจินต์ น่วมสำราญ	
รศ.ดร. วิทยา ทิพย์สุวรรณพร	

หัวข้อปริญญานิพนธ์	การออกแบบตัวควบคุมพีไอดีสำหรับชุดแทนเลื่อนในการสอบเทียบสายเทปวัดระยะ		
นักศึกษาผู้จัดทำ	นายชินรอส	मुखโรจน์	รหัสนักศึกษา 60010236
	นายพชรพล	อัครพงศ์ไพศาล	รหัสนักศึกษา 60010654
อาจารย์ที่ปรึกษา	รศ.ดร. อาจินต์	น่วมสำราญ	
	รศ.ดร. วิทยา	ทิพย์สุวรรณพร	
ปีการศึกษา	2563		

## บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการออกแบบตัวควบคุมพีไอดีสำหรับการเคลื่อนที่ของชุดแทนเลื่อนในอุปกรณ์สอบเทียบสายเทปวัดระยะที่ใช้งานในห้องปฏิบัติการสอบเทียบความยาว และ มิติ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณลักษณะของกระบวนการให้เป็นแบบอัตโนมัติ ลดการสิ้นของโมดูลกล้องถ่ายภาพและลดความผิดพลาดในการวัดสายเทปวัดระยะจากมุมพิทช์ (pitch) และมุมยอร์ว (yaw) ระบบที่ออกแบบใช้ STM32 เป็นไมโครคอนโทรลเลอร์สำหรับควบคุมการเคลื่อนที่ของชุดแทนเลื่อนด้วยอัลกอริทึมพีไอดี และ Raspberry Pi ที่มีระบบปฏิบัติการพื้นฐานเป็น Raspbian สำหรับควบคุมการสิ้นของโมดูลกล้องถ่ายภาพด้วยไมโครเซอร์โวมอเตอร์ 2 แกนพิทช์แกนยอร์วด้วยอัลกอริทึมพีไอดี ทำการทดลองควบคุมการเคลื่อนที่ของแทนเลื่อนไปที่ระยะ 10 20 และ 30 เซนติเมตร ด้วยข้อกำหนดทางสมรรถนะค่าพุ่งเกินน้อยกว่า 3% ค่าเวลาคงตัวน้อยกว่า 25 วินาที จากผลการทดลองพบว่าอัลกอริทึมพีไอดีสำหรับชุดแทนเลื่อนที่ออกแบบสามารถลดค่าความผิดพลาดจากการสิ้นจากการเคลื่อนที่ของชุดแทนเลื่อนทั้งในแนวการเคลื่อนที่ มุมพิทช์และมุมยอร์วด้วยค่าพุ่งเกิน 0% ค่าความผิดพลาดที่สถานะคงตัวเป็นศูนย์ และค่าเวลาคงตัวที่ 20.2 14.4 และ 10.8 วินาทีสำหรับการเคลื่อนที่ในระยะ 10 20 และ 30 เซนติเมตรตามลำดับ

<b>Thesis Tittle</b>	Design PID Controller for carriage in measuring tape calibration system	
<b>Authors</b>	Mr.Chinoros	Mookrote
	Mr.Patcharapon	Akarapongpaisan
<b>Thesis advisor</b>	Assoc.Prof. Dr. Arjin	Numsomran
	Assoc.Prof. Dr. Vittaya	Tipsuwanporn
<b>Year</b>	2020	

## Abstract

This thesis presents the design of a PID controller for carriage motion in measuring tape calibration equipment used in the length and dimensional calibration laboratory. The objective is to improve the process characteristics to be automated, reduce camera module shake and reduce the tape measuring errors from pitch and yaw angles. The designed system used an STM32 microcontroller for controlling the movement of carriage by PID algorithm. As well as applying a Raspberry Pi based on the Raspbian operating system to control camera module shake by 2 axes micro servo motor, yaw and pitch axis, with PID algorithm. The experiments were performed to control the carriage motion at 10, 20 and 30 centimeters with the performance specification, a percent overshoot less than 3%, a settling time less than 25 seconds. The proposed PID controller for carriage motion in measuring tape calibration equipment can reduce the error due to vibration caused by the movement of the carriage in both directions, pitch angle and yaw angle with 0% overshoot, error steady state of zero, and the settling time at 20.2, 14.4 and 10.8 seconds for movements in a distance of 10, 20 and 30 centimeters, respectively.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเนื่องด้วยคำแนะนำ คำปรึกษา รวมถึงอุปกรณ์เครื่องมือต่าง ๆ ที่ใช้สำหรับการทำการวิจัยจาก รศ.ดร. อาจินต์ น่วมสำราญ และ รศ.ดร. วิทยา ทิพย์สุวรรณพร ซึ่งเป็นอาจารย์ผู้ให้คำปรึกษา ปรับทัศนคติและความคิดเกี่ยวกับการทำโครงการให้แก่ผู้จัดทำด้วยดีเสมอมาทางผู้จัดทำรู้สึกซาบซึ้ง และขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ สาขาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้อบรม สั่งสอน ประสิทธิ์ประสาทวิชาความรู้ รวมถึงให้คำแนะนำต่าง ๆ ขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนทุกคนที่ช่วยกันทำงาน อดทน มานะบากบั่นร่วมกันมาจนโครงการนี้สำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ขอขอบพระคุณบิดา มารดา และครอบครัว ที่เป็นกำลังใจให้การช่วยเหลือต่าง ๆ ในการทำโครงการนี้ สำเร็จลุล่วงไปด้วยดี

คณะผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อ .....	I
ABSTRACT .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญตาราง .....	IX
สารบัญรูป .....	X
บทที่ 1 บทนำ .....	1
1.1 ความสำคัญของปริภูมิอนุพันธ์ .....	1
1.2 วัตถุประสงค์ของปริภูมิอนุพันธ์ .....	2
1.3 ขอบเขตของปริภูมิอนุพันธ์ .....	2
1.4 ขั้นตอนการศึกษา .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง .....	4
2.1 หลักการวัดความตรง .....	4
2.1.1 มุมออยเลอร์ (Euler Angles) .....	4
2.1.2 สมการที่เกี่ยวข้องกับหลักการวัดความตรง .....	5
2.2 หลักการประเมินค่าความไม่แน่นอน .....	6
2.2.1 ค่าเฉลี่ยของการวัด .....	6
2.2.2 ค่าเบี่ยงเบนมาตรฐาน .....	7
2.2.3 ค่าความไม่แน่นอนชนิดเอ .....	7
2.2.4 การแจกแจงความน่าจะเป็น .....	7
2.2.5 ค่าความไม่แน่นอนชนิดบี .....	9
2.2.6 ค่าความไม่แน่นอนมาตรฐาน .....	9

## สารบัญ(ต่อ)

2.2.7	ค่าความไม่แน่นอนรวม.....	9
2.2.8	ค่าความไม่แน่นอนขยาย .....	10
2.3	ตัวตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ .....	10
2.3.1	ไจโรสโคป .....	10
2.3.2	ตัววัดความเร่ง.....	11
2.3.3	Sensor fusion.....	11
2.4	ทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง (DC MOTOR) .....	12
2.5	ตัวเข้ารหัสแบบแม่เหล็ก .....	13
2.5.1	โครงสร้างของตัวเข้ารหัสแม่เหล็ก .....	13
2.5.2	หลักการของการเข้ารหัสแม่เหล็ก.....	14
2.6	ทฤษฎีการควบคุมแบบพีไอดี (PID CONTROLLER).....	16
2.6.1	การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี.....	16
2.6.2	Proportional Action.....	19
2.6.3	Integral Action.....	20
2.6.4	Derivative Action.....	21
2.6.5	ผลการตอบสนองของระบบ .....	22
2.7	ทฤษฎีการประมวลผลภาพ.....	23
2.7.1	Region-of-interest (ROI) .....	24
2.7.2	RGB Image .....	25
2.7.3	Grayscale Image .....	25
2.7.4	Threshold .....	27
2.8	ระบบปฏิบัติการ RASPBAN .....	28
<b>บทที่ 3 วิธีการดำเนินงาน.....</b>		<b>29</b>
3.1	ขั้นตอนการดำเนินงาน .....	29

## สารบัญ(ต่อ)

3.2	โครงสร้างและการออกแบบชุดแทนเลียนสำหรับการสอบเทียบสายเทปวัดระยะ.....	29
3.2.1	ชุดแทนเลียนสำหรับการสอบเทียบสายเทปวัดระยะโมเดลที่ 1 .....	29
3.2.2	ชุดแทนเลียนสำหรับการสอบเทียบสายเทปวัดระยะโมเดลที่ 2 .....	31
3.3	องค์ประกอบทางด้านฮาร์ดแวร์.....	34
3.3.1	บอร์ดไมโครคอนโทรลเลอร์ STM32 NUCLEO F401RE.....	34
3.3.2	บอร์ด Raspberry Pi 3 Model B.....	35
3.3.3	Adept RaspClaws Robot Hat.....	36
3.3.4	มอเตอร์ไฟฟ้ากระแสตรงพร้อม Encoder.....	37
3.3.5	บอร์ดสำหรับไดร์มอเตอร์ไฟฟ้ากระแสตรง Dual TB6612FNG.....	38
3.3.6	เซนเซอร์วัดความเอียง MPU 6050.....	39
3.3.7	เซอร์โวมอเตอร์ MG90S.....	40
3.3.8	อะแดปเตอร์ไร้สาย EW-7811Un .....	41
3.4	ซอฟต์แวร์ที่ใช้พัฒนาระบบ .....	41
3.4.1	STM32CubeIDE.....	41
3.4.2	MATLAB .....	42
3.4.3	VNC Viewer .....	43
3.4.4	Pycharm.....	44
3.5	การควบคุมการเคลื่อนที่ของชุดแทนเลียน.....	45
3.5.1	การเชื่อมต่อระหว่างบอร์ด STM32 F401RE กับบอร์ด Drive DC Motor.....	45
3.5.2	การตั้งค่า Pinout และ Clock บน STM32 CubeIDE.....	46
3.5.3	การเขียนโปรแกรมในการเคลื่อนที่สำหรับชุดแทนเลียน .....	48
3.6	การควบคุมการสั้นของชุดแทนเลียน .....	48
3.6.1	การส่งข้อมูลระหว่างอุปกรณ์.....	48
3.6.2	การติดตั้งระบบปฏิบัติการ Raspbian.....	49
3.6.3	การเขียนโปรแกรมเพื่อควบคุมการสั้นของชุดแทนเลียน.....	54

## สารบัญ(ต่อ)

3.7.1	วิธีการเตรียมและติดตั้งเครื่องมือและสายวัด .....	54
3.7.2	ขั้นตอนการสอบเทียบสายเทปวัดระยะที่เป็นมาตรฐานอ้างอิง .....	55
<b>บทที่ 4</b>	<b>ผลการทดลอง.....</b>	<b>58</b>
4.1	ผลการทดลองเกี่ยวกับการเคลื่อนที่ของชุดแท่นเลื่อน .....	58
4.1.1	การเคลื่อนที่ของชุดแท่นเลื่อนไปที่ 10 cm. ....	58
4.1.2	การเคลื่อนที่ของชุดแท่นเลื่อนไปที่ 20 cm. ....	59
4.1.3	การเคลื่อนที่ของชุดแท่นเลื่อนไปที่ 30 cm. ....	60
4.2	ผลการทดลองเกี่ยวกับการควบคุมสมดุลของชุดแท่นเลื่อน .....	60
4.2.1	ผลการวัดค่าความผิดพลาดในแนวแกน พิทซ์ และ ยอร์วี ขณะอยู่ที่ตำแหน่ง 10 cm. ....	60
4.2.2	ผลการวัดค่าความผิดพลาดในแนวแกน พิทซ์ และ ยอร์วี ขณะอยู่ที่ตำแหน่ง 20 cm. ....	63
4.2.3	ผลการวัดค่าความผิดพลาดในแนวแกน พิทซ์ และ ยอร์วี ขณะอยู่ที่ตำแหน่ง 30 cm. ....	65
4.3	ผลการทดลองของชุดแท่นเลื่อนจากการถ่ายภาพที่ได้ในระบบจริง .....	67
4.3.1	ผลของการถ่ายภาพจากกล้องบนชุดแท่นเลื่อนที่ระยะ 10 cm.....	67
4.3.2	ผลของการถ่ายภาพจากกล้องบนชุดแท่นเลื่อนที่ระยะ 20 cm.....	68
4.3.3	ผลของการถ่ายภาพจากกล้องบนชุดแท่นเลื่อนที่ระยะ 30 cm.....	69
<b>บทที่ 5</b>	<b>สรุปผลการทดลองและข้อเสนอแนะ.....</b>	<b>70</b>
5.1	สรุปผลการทดลอง .....	70
5.2	ปัญหาและอุปสรรค.....	70
5.3	ข้อเสนอแนะสำหรับการวิจัยในอนาคต.....	71
<b>บรรณานุกรม.....</b>		<b>72</b>

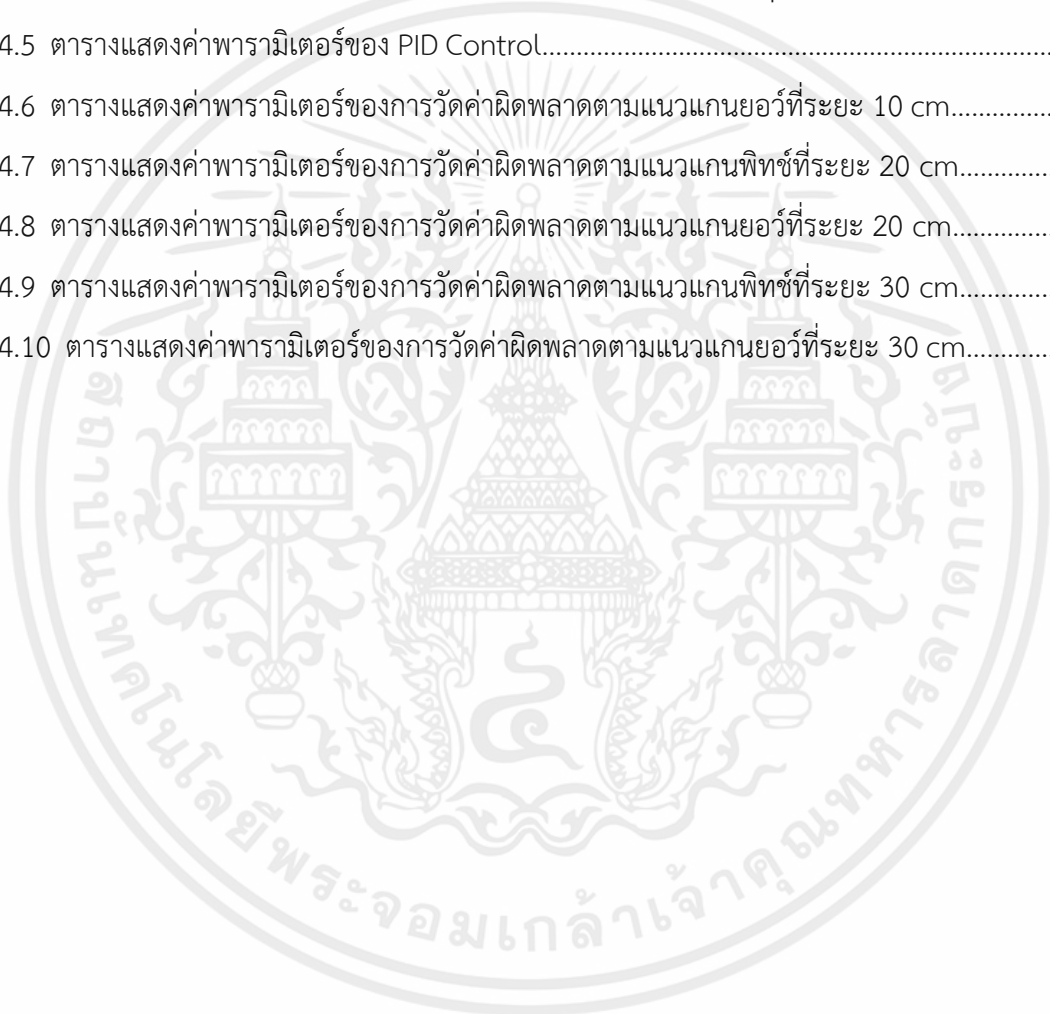
## สารบัญ(ต่อ)

ภาคผนวก .....	75
ภาคผนวก ก .....	76
ก.1 โปรแกรม STM32CubeIDE Code.....	77
ก.2 โปรแกรม Thony IDE Code.....	80
ภาคผนวก ข.....	84
ข.1 การออกแบบและการประกอบฮาร์ดแวร์โมเดลที่ 1 .....	85
ข.2 การออกแบบและการประกอบฮาร์ดแวร์โมเดลที่ 2 .....	87
ภาคผนวก ค .....	89



# สารบัญตาราง

ตารางที่	หน้า
4.1 ตารางแสดงค่าพารามิเตอร์ของ PID Control.....	58
4.2 ตารางแสดงค่าพารามิเตอร์ของผลการตอบสนองจากการปรับ Setpoint ที่ 10 cm.....	59
4.4 ตารางแสดงค่าพารามิเตอร์ของผลการตอบสนองจากการปรับ Setpoint ที่ 30 cm.....	60
4.5 ตารางแสดงค่าพารามิเตอร์ของ PID Control.....	61
4.6 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยวี่ที่ระยะ 10 cm.....	62
4.7 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนพิทซ์ที่ระยะ 20 cm.....	63
4.8 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยวี่ที่ระยะ 20 cm.....	64
4.9 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนพิทซ์ที่ระยะ 30 cm.....	65
4.10 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยวี่ที่ระยะ 30 cm.....	66



# สารบัญรูป

รูปที่	หน้า
2.1 การวัดความตรง.....	4
2.2 แสดงการทำงานของมอเตอร์ไฟฟ้ากระแสตรง.....	12
2.3 แสดงกฎมือซ้ายของเฟลมมิ่ง (Fleming's left hand rule).....	13
2.4 ตัวเข้ารหัสแม่เหล็กการกำหนดค่า Shaft-End.....	14
2.5 ความแข็งแรงของสนามแม่เหล็กที่ตรวจพบโดยองค์ประกอบฮอลล์.....	14
2.6 สนามแม่เหล็กของส่วนประกอบแกน X (Bx) และส่วนประกอบแกน Y.....	15
2.7 Lissajous รูปของความแข็งแรงสนามแม่เหล็ก.....	16
2.8 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป.....	18
2.9 ลักษณะสมบัติของตัวควบคุมแบบสัดส่วน.....	20
2.10 ผลการตอบสนองของระบบ.....	22
2.11 แสดงภาพต้นฉบับก่อนทำการใช้ฟังก์ชัน ROI.....	24
2.12 แสดงภาพหลังทำการใช้ฟังก์ชัน ROI.....	24
2.13 แสดงระดับสีเทา 256 ระดับ.....	25
2.14 แสดงภาพต้นฉบับ.....	26
2.15 แปลงจากภาพสีเป็นภาพระดับสีเทา.....	26
2.16 แสดงภาพที่ผ่านการ Threshold และแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ.....	27
2.17 Raspberry Pi OS.....	28
3.1 แสดงขั้นตอนการดำเนินงาน.....	29
3.2 แสดงการออกแบบชุดสำหรับควบคุมการสั่นบนแท่นเลื่อนโมเดลที่ 1.....	30
3.3 โครงสร้างระบบสอบเทียบสายเทปวัดระยะโมเดลที่ 1.....	31
3.4 แสดงการออกแบบโมเดลของชุดควบคุมการสั่นบน AutoCAD.....	31
3.5 แสดงส่วนประกอบของชุดควบคุมการสั่นหลังการออกแบบ.....	32
3.6 โครงสร้างระบบสอบเทียบสายเทปวัดระยะโมเดลที่ 2.....	32
3.7 โครงสร้างรางอลูมิเนียมโปรไฟล์.....	33

## สารบัญรูป(ต่อ)

รูปที่	หน้า
3.8 แทนเลื่อน.....	33
3.9 ชุดสำหรับควบคุมการสั่นบนแทนเลื่อน.....	34
3.10 NUCLEO F401RE.....	34
3.11 Raspberry Pi 3 Model B.....	35
3.12 Robot Hat.....	36
3.13 มอเตอร์ดีซีเอ็นโค้ดเดอร์ DC Gear Motor + Encoder Hall D-type.....	37
3.14 Motor Driver - Dual TB6612FNG.....	38
3.15 MPU 6050.....	39
3.16 เซอร์โวมอเตอร์ MG90S.....	40
3.17 อะแดปเตอร์ไร้สาย EW-7811UN.....	41
3.18 STM32 CubeIDE.....	41
3.19 MATLAB .....	42
3.20 VNC Viewer.....	43
3.21 PyCharm.....	44
3.22 แสดงหน้าต่างการใช้งาน PyCharm.....	45
3.23 แสดงการเชื่อมต่อระหว่าง ไมโครคอนโทรลเลอร์กับ DC Motor Driver Board.....	46
3.24 แสดงการเซ็ต Pinout ก่อนเริ่มเขียนโปรแกรม.....	46
3.25 แสดงการเซ็ต Clock สำหรับการส่งสัญญาณ.....	47
3.26 แสดงการกำหนดค่า Prescaler และ Counter Period ที่สัมพันธ์กับ Clock.....	47
3.27 แสดงการเชื่อมต่ออะแดปเตอร์ไร้สายเข้ากับ Raspberry Pi 3.....	48
3.28 แสดงขั้นตอนการใส่ระบบปฏิบัติการ Raspbian ลงใน Micro SD.....	49
3.29 แสดงการสร้างเท็กซ์ไฟล์ชื่อ ssh เก็บไว้ในรูทของการ์ด Micro SD.....	50
3.30 แสดงเท็กซ์ไฟล์ชื่อ SSH.....	50
3.31 แสดงการหา IP Address.....	51

## สารบัญรูป(ต่อ)

รูปที่	หน้า
3.32 แสดงการใช้งานโปรแกรม Putty ) เพื่อเชื่อมต่อไปยัง Raspberry PI.....	52
3.33 แสดงการ Login.....	52
3.34 แสดงหน้าต่างกำหนดค่าเริ่มต้น.....	53
3.35 แสดงขั้นตอนหลังจากทำการเลือก เลือก Expand Filesystem.....	53
3.36 แสดงหน้าต่างเริ่มต้นใช้งาน.....	54
3.37 การปรับตั้งสายวัด.....	55
3.38 แสดงการปรับตั้งชุดแทนเลื่อนให้อยู่ที่จุดเริ่มต้น.....	55
3.39 แสดงแผนภูมิการทำงานของชุดโปรแกรมควบคุมและประมวลผล.....	56
4.1 แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 10 cm.....	58
4.2 แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 20 cm.....	59
4.3 แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 30 cm.....	60
4.4 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทซ์ที่ระยะ 10 cm.....	61
4.5 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์วี่ที่ระยะ 10 cm.....	62
4.6 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทซ์ที่ระยะ 20 cm.....	63
4.7 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์วี่ที่ระยะ 20 cm.....	64
4.8 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทซ์ที่ระยะ 30 cm.....	65
4.9 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์วี่ที่ระยะ 30 cm.....	66
4.10 แสดงผลภาพที่เลื่อนไปที่ระยะ 10 cm. จากกล่องบนชุดแทนเลื่อน.....	67
4.11 แสดงผลของชุดแทนเลื่อนที่เลื่อนไปที่ระยะ 10 cm. ก่อนถ่ายภาพ.....	67
4.12 แสดงผลภาพที่เลื่อนไปที่ระยะ 20 cm. จากกล่องบนชุดแทนเลื่อน.....	68
4.13 แสดงผลของชุดแทนเลื่อนที่เลื่อนไปที่ระยะ 20 cm. ก่อนถ่ายภาพ.....	68
4.14 แสดงผลภาพที่เลื่อนไปที่ระยะ 30 cm. จากกล่องบนชุดแทนเลื่อน.....	69
4.15 แสดงผลของชุดแทนเลื่อนที่เลื่อนไปที่ระยะ 30 cm. ก่อนถ่ายภาพ.....	69

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปริญญาบัตร

เนื่องจากในห้องปฏิบัติการจริงและในอุตสาหกรรม การสอบเทียบสายเทปวัดระยะจะมีค่าความไม่แน่นอนของการสอบเทียบเกิดขึ้นเสมอ ดังนั้นจึงต้องทำการศึกษาและควบคุมค่าความไม่แน่นอนของการสอบเทียบของการวัด เพื่อที่จะได้ค่าความไม่แน่นอนตรงตามมาตรฐานขององค์การมาตรวิทยาเชิงพานิชย์สากล (International Organization of Legal Metrology, OIML) โดยอ้างอิงเอกสารมาตรฐาน OIML R35-1 ซึ่งเป็นเอกสารเกี่ยวกับการวัดความยาวของวัสดุ รวมถึงกฎหมาย ชั่ง ตวง วัด ด้วยผู้จัดทำได้มองเห็นถึงปัญหาเหล่านี้ จึงมีแนวคิดในการจัดทำโครงการออกแบบตัวควบคุมพีไอดีสำหรับชุดแท่นเลื่อนในการสอบเทียบสายเทปวัดระยะ โดยการสร้างแบบจำลองการสอบเทียบสายเทปวัดระยะ ซึ่งประกอบด้วย ชุดแท่นเลื่อน, รางอลูมิเนียมโปรไฟล์สำหรับชุดแท่นเลื่อน ระยะไม่เกิน 1.5 m, ชุดระบบวัดความผิดพลาดตามแนวแกน, ชุดการประมวลผลภาพจากกล้อง และชุดควบคุมการเคลื่อนที่ของแท่นเลื่อน พร้อมทั้งออกแบบตัวควบคุมพีไอดีในการควบคุมการเคลื่อนที่ของแท่นเลื่อน สร้างอัลกอริทึมที่ช่วยลดการสั่นที่ทำให้เกิดความผิดพลาดในการวัด ในขณะที่เคลื่อนที่ของชุดแท่นเลื่อนของระบบสอบเทียบสายเทปวัดระยะ

โดยจะต้องศึกษาทฤษฎีและหลักการเกี่ยวกับ สายวัดมาตรฐานตร, หลักการสอบเทียบสายเทปวัดระยะ, หลักการทำงานของระบบสอบเทียบสายเทปวัดระยะ, มาตรฐาน OIML R35-1, การเขียนโปรแกรมด้วยภาษา C, ทฤษฎีระบบควบคุมพีไอดี ในควบคุมควบคุมการเคลื่อนที่ของชุดแท่นเลื่อน เนื่องจากขณะเคลื่อนที่ของชุดแท่นเลื่อนนั้น จะเกิดค่าความไม่แน่นอนหรือความผิดพลาดในแนวแกน ทั้งแกน  $x$ ,  $y$  และ  $z$  ซึ่งมีสาเหตุมาจากการสั่นสะเทือนขณะเคลื่อนที่ของชุดแท่นเลื่อน โดยโครงการนี้ จะศึกษาและลดค่าความไม่แน่นอนที่เกิดขึ้นในการสอบเทียบสายของชุดสอบเทียบสายเทปวัดระยะ ซึ่งมีสาเหตุมาจากการผิดพลาดที่เกิดขึ้นในแนว Lateral และ Vertical ซึ่งก็คือ ค่าความผิดพลาดแบบพิทช์ (Pitch) และแบบยอร์ว (Yaw) ขณะเคลื่อนที่ของชุดแท่นเลื่อน ซึ่งการตรวจสอบข้อผิดพลาดแบบพิทช์ (Pitch) และแบบยอร์ว (Yaw) จะใช้หลักการวัดด้วยยอร์วและพิทช์เซนเซอร์

## 1.2 วัตถุประสงค์ของปริญญาโท

1. เพื่อศึกษา เรียนรู้ เพิ่มพูนประสบการณ์ วิจัย พัฒนาและสร้างทักษะการออกแบบตัวควบคุมพีไอ ดี สำหรับชุดแทนเลียนในการสอบเทียบสายเทปวัดระยะ
2. เพื่อศึกษาและพัฒนาตัวควบคุมพีไอดี สำหรับชุดแทนเลียนในการสอบเทียบสายเทปวัดระยะ ให้เกิดค่าความผิดพลาดน้อยที่สุด เพื่อแก้ปัญหาในห้องปฏิบัติการจริงและอุตสาหกรรม
3. เพื่อศึกษาค่าความผิดพลาดในแนวแกนที่เกิดขึ้น ขณะชุดแทนเลียนเคลื่อนที่ไปยังตำแหน่งที่ได้ทำการกำหนด ซึ่งประกอบด้วยค่าความผิดพลาดแบบพิทช์ (Pitch) และแบบยอร์ว (Yaw)
4. เพื่อศึกษาวิธีการออกแบบและการเขียนโปรแกรมระบบควบคุมความผิดพลาดแบบพิทช์(Pitch) และยอร์ว (Yaw) ในขณะเคลื่อนที่ของชุดแทนเลียน

## 1.3 ขอบเขตของปริญญาโท

1. เป็นการสร้างแทนเลียนพร้อมควบคุมการเคลื่อนที่ของแทนเลียน เพื่อแก้ปัญหาที่เกิดขึ้นจริงในอุตสาหกรรม โดยกำหนด ความยาวของแทนไม่เกิน 1 เมตร
2. สร้างอัลกอริทึมที่ช่วยลดการสั่นที่ทำให้เกิดความผิดพลาดในแนวแกน ซึ่งประกอบด้วยค่าความผิดพลาดแบบพิทช์ (Pitch) และแบบยอร์ว (Yaw) ในขณะเคลื่อนที่ของชุดแทนเลียน
3. ออกแบบ ซอฟต์แวร์ สำหรับชุดควบคุมพีไอดี เพื่อควบคุมการเคลื่อนที่ของแทนเลียน

## 1.4 ขั้นตอนการศึกษา

เริ่มจากศึกษาการทำงานของกระบวนการ โดยเริ่มจากศึกษาข้อมูลของสายเทปวัดระยะมาตรฐาน และหลักการสอบเทียบ หลักการทำงานของระบบสอบเทียบสายวัด การทำงานของอุปกรณ์ต่าง ๆ และในส่วนของซอฟต์แวร์จะต้องศึกษาเกี่ยวกับ การใช้งาน AutoCAD ในการออกแบบฮาร์ดแวร์, การใช้งาน STM32cubeIDE และ การใช้MATLAB R2020a เป็นต้น

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำไปประยุกต์ใช้ในการแก้ปัญหา ในชุดแทนเลื้อนในการสอบเทียบสายเทปวัดระยะ
2. สามารถออกแบบและเขียนโปรแกรมควบคุมความผิดพลาดแบบพิทช์ (Pitch) และ ยอร์ว (Yaw)
3. สามารถพัฒนาระบบสอบเทียบสายเทปวัดระยะ ให้เกิดความถูกต้องและแม่นยำ
4. สามารถนำความรู้ที่ได้ไปใช้ในการประกอบอาชีพต่อไป



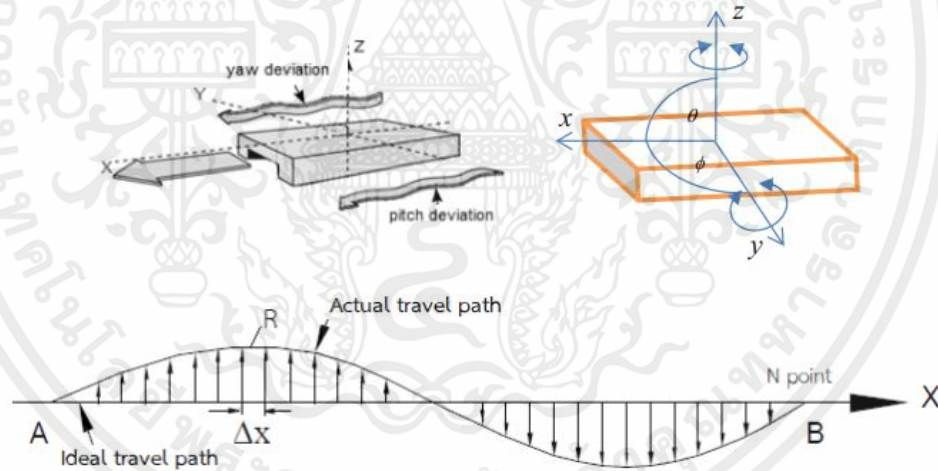
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและหลักการที่เกี่ยวข้อง

### 2.1 หลักการวัดความตรง

พิจารณาหกแกนองศาอิสระต่าง ๆ ที่กำหนดไว้บนชุดแท่นเลื่อนดังนี้ สามแกนองศาอิสระแกน X, Y และ Z อีกสามองศาอิสระเป็นการหมุนรอบแกน X, Y และ Z การเคลื่อนที่เหล่านี้เป็นไปตามกฎมือขวาโดยแกน X และ Y แทนด้วยนิ้วชี้และนิ้วกลางตั้งฉากกันกับแกน Z แทนด้วยนิ้วหัวแม่มือชี้ไปในทิศทางบวกแกนต่าง ๆ ก็ชี้ไปในทิศทางบวกรวมไปถึงการหมุนรอบแกนนั้นด้วยการประกอบด้วยการเคลื่อนที่ไปตามแกน, การหมุนรอบแกนร่วม (Coordinate axes) แกน X และ Y อยู่บนระนาบตามแนวนอน การเคลื่อนที่ของแท่นบนและล่างให้ปรับไปที่แกน X และแกน Z อยู่บนระนาบตามแนวตั้ง โดยในหลักการวัดความตรงนี้จะมีการนำหลักการและทฤษฎีของออยเลอร์ (Euler Angles) มาใช้ร่วมด้วย [1]



รูปที่ 2.1 การวัดความตรง

ที่มา : วันชัย ชินชูศักดิ์ (2561 : 8) เทคนิคการประเมินค่าความไม่แน่นอนของระบบสอบเทียบสายวัด

#### 2.1.1 มุมออยเลอร์ (Euler Angles)

มุมออยเลอร์ (Euler Angles) คือ มุมสมมุติที่ใช้ในการกำหนดการเปลี่ยนจากแกนอ้างอิงใด ๆ เช่น แกนอ้างอิง inertial frame ไปเป็นแกนอ้างอิง object frame ซึ่งแกนอ้างอิง object frame จะเป็นผลลัพธ์ของการหมุนในลำดับดังนี้ ยอร์ (yaw) ไปพิทช์ (pitch) ไปโรล (roll) ซึ่งลำดับการหมุนแกนมี

ความสำคัญกำหนดให้ ยอร์ (yaw) หมุนรอบแกน Down (D) - แกน z ได้  $R_{NED}^1(\psi)$  ดังสมการที่ 2.1 พิตช์ (pitch) หมุนรอบแกน East (E) - แกน y ได้  $R_1^2(\theta)$  ดังสมการที่ 2.2 และ โรล (roll) หมุน รอบแกน North(N) - แกน x ได้  $R_2^0(\phi)$  [2] ดังสมการที่ 2.3

$$R_{NED}^1(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$R_1^2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

$$R_2^0(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.3)$$

ซึ่งจะได้ Transformation matrix ที่เปลี่ยนจากแกนอ้างอิง inertial frame เป็นแกนอ้างอิง object frame เป็นไปดังสมการที่ 2.4

$$\begin{aligned} R_{NED}^0(\phi, \theta, \psi) &= R_2^0(\phi)R_1^2(\theta)R_{NED}^1(\psi) \\ &= \begin{bmatrix} c\theta c\psi & c\theta c\psi & -s\theta \\ c\psi s\theta s\phi - c\theta s\psi & c\theta c\psi + s\theta s\phi s\psi & c\theta s\phi \\ c\phi c\psi s\theta + s\phi s\psi & c\phi s\theta s\psi - c\psi s\phi & c\theta c\phi \end{bmatrix} \end{aligned} \quad (2.4)$$

### 2.1.2 สมการที่เกี่ยวข้องกับหลักการวัดความตรง

ในกรณีการวัดความตรง ดังรูปที่ 2.1 เป็นการวัดระยะที่ตั้งฉากกับแนวการเคลื่อนที่ในระนาบ แนวนอน ให้  $y(x)$  ถูกกำหนดให้เป็นเส้นตรงอ้างอิงผ่านจุดเริ่มต้น  $y(0)$  และจุดสิ้นสุด  $y(L)$ ,  $L$ : ความยาว (Length) ซึ่ง  $y(0) = y(L) = 0$  ค่าความชันกำหนดได้ว่า [1]

$$\frac{dy}{dx} = \tan \phi \quad (2.5)$$

สมมติว่าให้  $R$  เป็นรัศมีส่วนโค้งของแนวการเคลื่อนที่จริงมีค่ามากกว่า  $\Delta x$  หนึ่ง  $\tan$  เท่ากับ  $\phi$  ดังนั้น

$$y(x) = \int_0^x \frac{dy}{dx} dx = \int_0^x \phi(x) dx \quad (2.6)$$

จำนวนจุดทั้งหมด  $N$  จุด เท่ากับ  $L / \Delta x$  , ให้  $x_n = n\Delta x$ ;

$$y(x_n) = \sum_{i=0}^n \phi(x_{i-1}) \Delta x + const \quad (2.7)$$

เพราะว่า  $Y(0) = Y(L) = 0$  , สมการที่ 2.7 จะได้เป็น

$$y(x_n) = \sum_{i=0}^n \phi(x_{i-1}) \Delta x - \frac{n}{N} \sum_{i=0}^N \phi(x_{i-1}) \Delta x \quad (2.8)$$

ที่  $y(x_n)$  เป็นค่าความตรงของระนาบแนวนอน,  $\phi(x_n)$  มุมยอร์ที่วัดได้  $\Delta x$  ค่ามุมการวัดที่เพิ่มขึ้น และ  $N$  เป็นจำนวนจุดที่ต้องการวัดทั้งหมด ดังนั้นการวัดยอร์ด้วยชุดแทนเลื่อนบนรางอลูมิเนียมเปรียบเสมือนเป็นการวัดความตรงของราง

## 2.2 หลักการประเมินค่าความไม่แน่นอน

หลักการประเมินค่าความไม่แน่นอนของการสอบเทียบสายเทปวัดระยะนั้น จะใช้หลักการประเมินเชิงสถิติในรูปแบบทางคณิตศาสตร์ ตามเอกสารมาตรฐานของ GUM [1] ประกอบด้วยหัวข้อต่อไปนี้

**2.2.1 ค่าเฉลี่ยของการวัด ( $\bar{X}$ )** พิจารณาจากผลรวมของการวัดทั้งหมดหารด้วยจำนวนครั้งที่วัด ( $n$ ) ดังสมการที่ 5.1 เมื่อค่าวัดใด ๆ ( $x_i$ ) ที่  $x_i = 1, 2, 3, \dots, n$  พิจารณาผลรวมการวัดนั้นจะได้ค่าเป็นดังสมการที่ 2.16 และค่าเฉลี่ยของการวัด ดังสมการที่ 2.17

$$\sum_{i=1}^n x_i = x_1 + x_2 + x_3 + \dots + x_n \quad (2.16)$$

$$\bar{X} = \frac{\sum_{i=1}^n (x_i)}{n} \quad (2.17)$$

2.2.2 ค่าเบี่ยงเบนมาตรฐาน ( $\sigma_{n-1}$ ) ค่าเบี่ยงเบนมาตรฐานจากการวัดเป็นค่าเฉลี่ยรากที่สองของผลรวมของผลต่างระหว่างค่าเฉลี่ย ( $\bar{X}$ ) กับค่าวัดใด ๆ ( $x_i$ ) ดังสมการที่ 2.18 ต่อไปนี้

เมื่อ

$$\sum_{i=1}^n (\bar{X} - x_i)^2 = (\bar{X} - x_1)^2 + (\bar{X} - x_2)^2 + \dots + (\bar{X} - x_n)^2 \quad (2.18)$$

ดังนั้น

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^n (\bar{X} - x_i)^2}{n-1}} \quad (2.19)$$

2.2.3 ค่าความไม่แน่นอนชนิดเอ ( $U_a$ ) เป็นผลจากความคลาดเคลื่อนของการสุ่มที่ได้จากการวัด (Random error) ได้จากการนำค่าเบี่ยงเบนมาตรฐาน ( $\sigma_{n-1}$ ) ของการวัดมาเฉลี่ยด้วยรากที่สอง (Square root) ของจำนวนครั้งที่วัดหรือค่าเบี่ยงเบนเฉลี่ย (Standard deviation of mean) เราเรียกตามจากแหล่งที่มาของการวัดว่า การวัดซ้ำได้ (Repeatability) ดังสมการที่ 2.20

$$u_a = u(x_i) = \frac{\sigma_{n-1}}{\sqrt{n}} \quad (2.20)$$

2.2.4 การแจกแจงความน่าจะเป็น เมื่อพิจารณาการแจกแจงความน่าจะเป็นจากพื้นที่ใต้กราฟทั้งหมดคิดเป็นร้อยละ 100 หรือ  $P_{(x)} = 1$  ซึ่งเป็นที่ทราบกันอยู่แล้วว่าการอินทิเกรต พื้นที่ใต้กราฟช่วงตั้งแต่  $\pm\sigma, \pm 2\sigma$  และ  $\pm 3\sigma$  จะได้  $P_{(x)}$  เท่ากับ 68.27 (68%), 95.45 (95%) และ 99.73 (99%) ตามลำดับ การแจกแจงความน่าจะเป็นมีส่วนเกี่ยวข้องกับและสำคัญกับกระบวนการวัดเป็นอย่างมาก ณ ที่นี้จะกล่าวถึงเฉพาะการแจกแจงความน่าจะเป็นที่อาจเกี่ยวข้องกับเท่านั้น ได้แก่ การแจกแจงความน่าจะเป็นแบบปกติ, การแจกแจงแบบสี่เหลี่ยมผืนผ้า และ การแจกแจงแบบสามเหลี่ยม ดังแสดงในรายละเอียดตามหัวข้อดังต่อไปนี้

**2.2.4.1 การแจกแจงความน่าจะเป็นแบบปกติ** โดยทั่วไปการวัดที่มีความสมบูรณ์จัดให้ มีลักษณะโค้งแบบระฆังคว่ำ โดยมีความสมมาตรกันทั้งด้านบวกและลบ การแจกแจงดังกล่าวได้แก่ การวัด ข้ำได้ ซึ่งมีความสัมพันธ์ต่อจำนวนครั้งของการวัดด้วย หากว่าจำนวนครั้งในการวัดมาก โอกาสความน่าจะเป็นอันเข้าสู่อนันต์จะยิ่งมากตามไปด้วย หรือเป็นค่าความแปรปรวนของการวัด ดังแสดงในสมการที่ 2.22

$$\text{เมื่อ } P_{(x)} = \frac{e^{-(x-\bar{x})^2/2\sigma^2}}{\sigma\sqrt{2\pi}} \quad (2.21)$$

ดังนั้น

$$\sigma^2 = \int_{-\sigma}^{+\sigma} P_{(x)} dx \quad (2.22)$$

**2.2.4.2 การแจกแจงความน่าจะเป็นแบบสี่เหลี่ยมผืนผ้า** พิจารณาการแจกแจงความน่าจะเป็นแบบสี่เหลี่ยมผืนผ้า ซึ่งเป็นปริมาณที่มีขอบเขตที่ชัดเจนไม่เปลี่ยนแปลงตลอดการวัดด้วยความแปรปรวน ซึ่งแสดงดังสมการที่ 2.23 ประเภทค่าความไม่แน่นอนของการแจกแจงแบบสี่เหลี่ยมผืนผ้าดังกล่าว ได้แก่ เกณฑ์ในการยอมรับ ความแม่นยำในการวัด และความละเอียดการอ่าน (Resolution)

$$\sigma^2 = \int_{-\sigma}^{+\sigma} P_{(x)} \cdot (x - \bar{x})^2 dx \quad (2.23)$$

**2.2.4.3 ความน่าจะเป็นของการแจกแจงแบบสามเหลี่ยม** พิจารณาการแจกแจงความน่าจะเป็นแบบสามเหลี่ยม ด้วยความแปรปรวนดังสมการที่ 2.24 ประเภทค่าความไม่แน่นอนของการแจกแจงแบบสามเหลี่ยมนั้นมีการแจกแจงดังกล่าวต้องมีความชัดเจนเชิงข้อมูล และรูปธรรม ได้แก่ การควบคุมอุณหภูมิ (Controller temperature) เป็นต้น

$$\sigma^2 = \int_{-a}^{+a} \frac{(a-x)}{a^2} \cdot (x - \bar{x})^2 dx \quad (2.24)$$

เมื่อ  $P_{(x)} = \frac{(a-x)}{a^2}$  กรณีที่พิจารณาด้านเดียวก็กำหนดให้  $\bar{x} = 0$  คงเหลือเทอม  $x^2$  เท่านั้น

### 2.2.5 ค่าความไม่แน่นอนชนิดบี ( $U_b$ )

เป็นค่าความไม่แน่นอนที่ได้จากค่าความคลาดเคลื่อนของระบบการวัด ทั้งหมดมารวมกันแบบรากที่สองของแต่ละปริมาณยกกำลังสอง (Root sum square) กล่าวคือ นำค่าความไม่แน่นอนจากแหล่งต่างๆ ที่พิจารณาได้มายกกำลังสองแล้วบวกกัน แล้วถอดรากที่สองของผลบวก ก็จะได้ค่าความไม่แน่นอนชนิดบี จะเขียนได้เป็น  $u_b^2 = u_{b1}^2 + u_{b2}^2 + \dots + u_n^2$  หรือ สามารถเขียนได้ดังสมการที่ 2.25

$$u_b = \sqrt{u_{b1}^2 + u_{b2}^2 + \dots + u_{bn}^2} \quad (2.25)$$

### 2.2.6 ค่าความไม่แน่นอนมาตรฐาน ( $u_i$ )

การรวมค่าความไม่แน่นอนชนิดต่างๆเข้าด้วยกัน เนื่องจากค่าความไม่แน่นอนดังกล่าวมาจากคนละแหล่งที่มา ดังนั้นควรพิจารณาก่อนว่าค่าความไม่แน่นอนชนิดใด ๆ นั้นต้องเป็นความไม่แน่นอนมาตรฐานเท่านั้น หากว่าไม่ใช่ความไม่แน่นอนมาตรฐานต้องแปลงให้เป็นความไม่แน่นอนมาตรฐาน ที่ระดับความเชื่อมั่นที่ 68% ก่อนเสมอ ดังกรณีตัวอย่างต่อไปนี้

1. ค่าความละเอียดการอ่านของเครื่องมือวัด ต้องแปลงค่าความละเอียดการอ่านให้อยู่ในรูปของค่าความไม่แน่นอน โดยสังเกตเครื่องหมายบวกลบ ( $\pm$ ) ให้ทำ semi-range โดยการหารด้วยสอง โดยแบ่งออกเป็นช่วงบวกและลบก่อน ( $u_{res}$ ) ต่อมาแปลงให้เป็นค่าความไม่แน่นอนมาตรฐาน กรณีนี้เป็นการแจกแจงแบบสี่เหลี่ยมให้หารด้วย  $\sqrt{3}$  ทั้งนี้ขึ้นอยู่กับชนิดของการแจกแจง

2. ค่าความไม่แน่นอนจากใบรับรองผลการสอบเทียบ ( $u_{STD}$ ) ที่ระบุด้วยระดับความเชื่อมั่นไว้ที่ 95% วิธีการแปลงให้เป็นค่าความไม่แน่นอนมาตรฐาน ( $u_{STD} / k$ ) ซึ่งในกรณีนี้ สามารถพิจารณาให้เป็นการแจกแจงแบบปกติ ซึ่งจะต้องทำการหารด้วย 2 ทั้งนี้จะขึ้นกับค่า k ของใบรับรองนั้น ๆ

### 2.2.7 ค่าความไม่แน่นอนรวม ( $u_c$ )

ได้จากแบบรากที่สองของผลรวมระหว่างค่าความไม่แน่นอนชนิดเอยกกำลังสอง กับค่าความไม่แน่นอนชนิดบียกกำลังสองจะได้เป็น  $u_c^2 = u_a^2 + u_b^2$  หรือสามารถเขียนใหม่ดังสมการที่ 2.26 ที่ระดับความเชื่อมั่น 68%

$$u_c = \sqrt{u_a^2 + u_b^2} \quad (2.26)$$

## 2.2.8 ค่าความไม่แน่นอนขยาย ( $u_{ex}$ )

การคำนวณหาค่า  $k_p$  (Coverage factor) ขององศาอิสระ  $V_{eff}$  ดังแสดงในสมการที่ 2.27 เพื่อปรับขยายค่าความไม่แน่นอนที่ระดับความเชื่อมั่นจาก 68% ไปที่ระดับความเชื่อมั่นต่าง ๆ เช่น 95% หรือ 99% ตามต้องการ

$$V_{eff} = \frac{u_c^4}{\sum_{i=1}^n \frac{u_i^4}{V_i}} = \frac{u_c^4}{\frac{u_a^4}{V_a} + \frac{u_b^4}{V_b}} \quad (2.27)$$

เมื่อได้ค่า  $V_{eff}$  หาค่าจาก Student's T-Distribution หรือจะใช้ฟังก์ชัน tinv ในโปรแกรมไมโครซอฟท์เอกเซล ดังนั้น จะได้ค่าความไม่แน่นอนพร้อมรายงานค่าความไม่แน่นอนที่ระดับความเชื่อมั่นที่ต้องการ

$$u_{ex} = k u_c \quad (2.28)$$

## 2.3 ตัวตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ

IMU Sensor เป็นอุปกรณ์ตรวจรู้สำหรับตรวจจับการเคลื่อนที่แบบหมุนของแกนทั้งสามแกนหรือที่เรียกกันว่า Pitch, Roll และ Yaw ซึ่งอาศัยหลักการทำงานของ Gyroscope Accelerometer [2]

### 2.3.1 ไจโรสโคป (Gyroscope)

ไจโรสโคปคืออุปกรณ์ที่ใช้ในการวัดการเคลื่อนที่แบบหมุน ประเภทหนึ่งของไจโรคือไจโร MEMS (Microelectromechanical System) ไจโรประเภทนี้คือเซ็นเซอร์ขนาดเล็กที่วัดความเร็วเชิงมุมในหน่วยองศาต่อวินาที ( $^{\circ}/s$ ) และ รอบต่อวินาที (RPS) เครื่องวัดการหมุน MEMS สามแกนซึ่งสามารถเป็นได้พบใน MPU-6050 ดังที่อธิบายไว้ข้างต้น ไจโรสโคปสามารถวัดความเร็วเชิงมุมและความสัมพันธ์รอบสามแกนที่แตกต่างกัน x, y และ z ซึ่งแสดงแกนที่แตกต่างกันและการหมุนรอบแกนเหล่านี้ของ MEMS สามแกนไจโรสโคป เมื่อเลือกไจโรสโคปเฉพาะช่วงการวัดของเซ็นเซอร์จำเป็นที่จะนำมาพิจารณาช่วงการวัดหรือช่วงเต็มสเกลคือค่าสูงสุดของความเร็วเชิงมุมซึ่งไจโรสามารถวัดได้ ขึ้นอยู่กับสำหรับสิ่งที่วัดได้ช่วงที่เลือกอาจแตกต่างกันไป MPU-6050 มีช่วงเต็มสเกล  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  และ  $\pm 2000$   $^{\circ}/s$

ข้อเสียของการใช้ไจโรสโคปสำหรับการวัดคือมีแนวโน้มที่จะล่องลอย ซึ่งหมายความว่า การอ่านไจโรสโคปเป็นศูนย์เริ่มต้นจะล่องลอยไปตามกาลเวลาเนื่องจากการรวมความไม่สมบูรณ์ภายใน และ เสี่ยงรบกวนภายในอุปกรณ์ และ นอกจากนี้ยังมีการชดเชยค่าคงที่ในแกนของไจโรสโคปซึ่งจะต้องได้รับการวัด และ ชดเชยก่อนใช้งาน

### 2.3.2 ตัววัดความเร่ง (Accelerometer)

เป็นอุปกรณ์ที่ใช้วัดความเร่งหรือการเปลี่ยนแปลงความเร็วในหนึ่งหน่วยเวลา ซึ่งจะอยู่ในหน่วย  $m/s^2$  โดยความเร่งที่วัดได้อาจรวมถึงการสั่นสะเทือนด้วย และ ข้อเสียของ Accelerometer คือมีความไวต่อการสั่นสะเทือนสูง รวมไปถึง Mechanical Noise ด้วย ซึ่งข้อเสียทั้งสองนี้ อาจนำไปสู่การวัดค่าที่ไม่ถูกต้องได้

### 2.3.3 Sensor fusion

ในการวัดความเร่งแม้ตัววัดความเร่งจะให้ค่าในสถานะที่ค่อนข้างจะเสถียรก็ตาม แต่จะไวต่อการสั่นสะเทือนและสัญญาณรบกวนแบบอื่น ๆ ด้วย ดังนั้นการใช้ IMU จึงมักใช้ สัญญาณจากไจโรมาช่วยปรับลดค่าความผิดพลาดของตัววัดความเร่ง เนื่องจากมีคุณสมบัติในการวัดการหมุนเชิงมุมเป็นหลัก และมีความไวต่อการเคลื่อนที่เชิงเส้นตรง แต่ไจโรก็ยังคงถูกรบกวนสัญญาณในรูปแบบอื่นเช่น การเลื่อนไหลของสัญญาณ (Drift) แต่หากมีการรวมสัญญาณจากเซนเซอร์ทั้งสองเข้าด้วยกันอย่างเหมาะสมก็อาจจะได้สัญญาณที่แสดงค่าความเอียง (Inclination) ได้ถูกต้อง วิธีการต่าง ๆ ที่นำมาใช้ในการหลอมรวมสัญญาณ (Data Fusion) มีพื้นฐานมาจาก Complementary filter ซึ่งอาจเริ่มต้นด้วยการพิจารณาจากอินทิเกรเตอร์อันดับหนึ่งซึ่งมีสมการเป็น  $\dot{x} = u$  สัญญาณที่วัดได้จากสองเซนเซอร์ อาจมีรูปแบบแสดงได้ดังนี้

$$y_x = L(s)x + \mu_x, y_u = u + \mu_u + b(t) \quad (2.9)$$

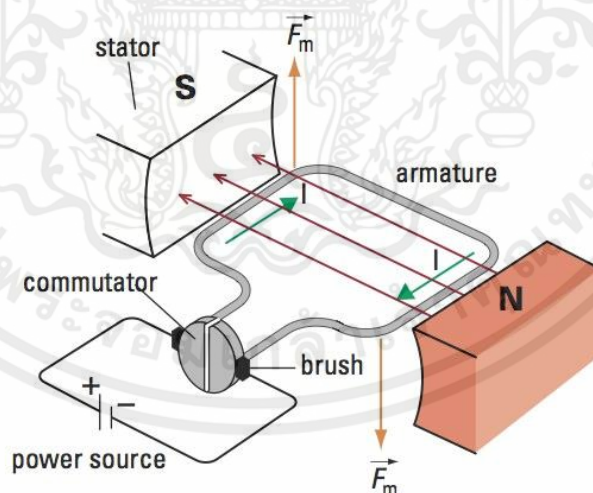
เมื่อ  $L(s)$  เป็นตัวกรองสัญญาณผ่านต่ำที่ขึ้นอยู่กับคุณสมบัติของเซนเซอร์  $\mu$  ซึ่งเป็นสัญญาณรบกวนในการวัดทั้งสองแบบ และ  $b(t)$  เป็นสัญญาณความเคลื่อนไหวย่านความถี่ต่ำที่สามารถหาค่าได้ ตามปกติตัวกรองความถี่ต่ำผ่าน  $L(s)$  จะมีค่าประมาณเท่ากับ 1 ตลอดย่านความถี่ที่ทำการวัด  $y_x$  ค่า  $y_x$  และ  $y_u$  สามารถรวมกันเพื่อให้ได้ค่า  $\hat{x}$  ซึ่งเป็นค่าประมาณของ  $x$  โดยใช้ตัวกรองดังต่อไปนี้

$$\hat{x} = F_1(s)y_x + F_2(s)\frac{y_u}{s} \quad (2.10)$$

โดยที่  $F_1(s)$  เป็นตัวกรองผ่านต่ำ และ  $F_2(s)$  เป็นตัวกรองผ่านสูง ค่าอัตราความเร็ว  $y_u$  จะถูกอินทิเกรตก่อนที่จะถูกกรองด้วย  $F_2(s)$  ซึ่งจะเก็บเฉพาะค่าคาดคะเน (Prediction) ในย่านความถี่สูงเท่านั้น และ ในขณะที่ค่าของ  $y_u/s$  ในย่านความถี่นั้นจะมีค่าไม่แน่นอนเนื่องจากมีผลของการอินทิเกรต  $\mu_u + b(t)$  เข้ามาร่วมด้วย ซึ่งตัวกรองในรูปแบบดังแสดงข้างต้นเรียกว่า Complementary filter

## 2.4 ทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

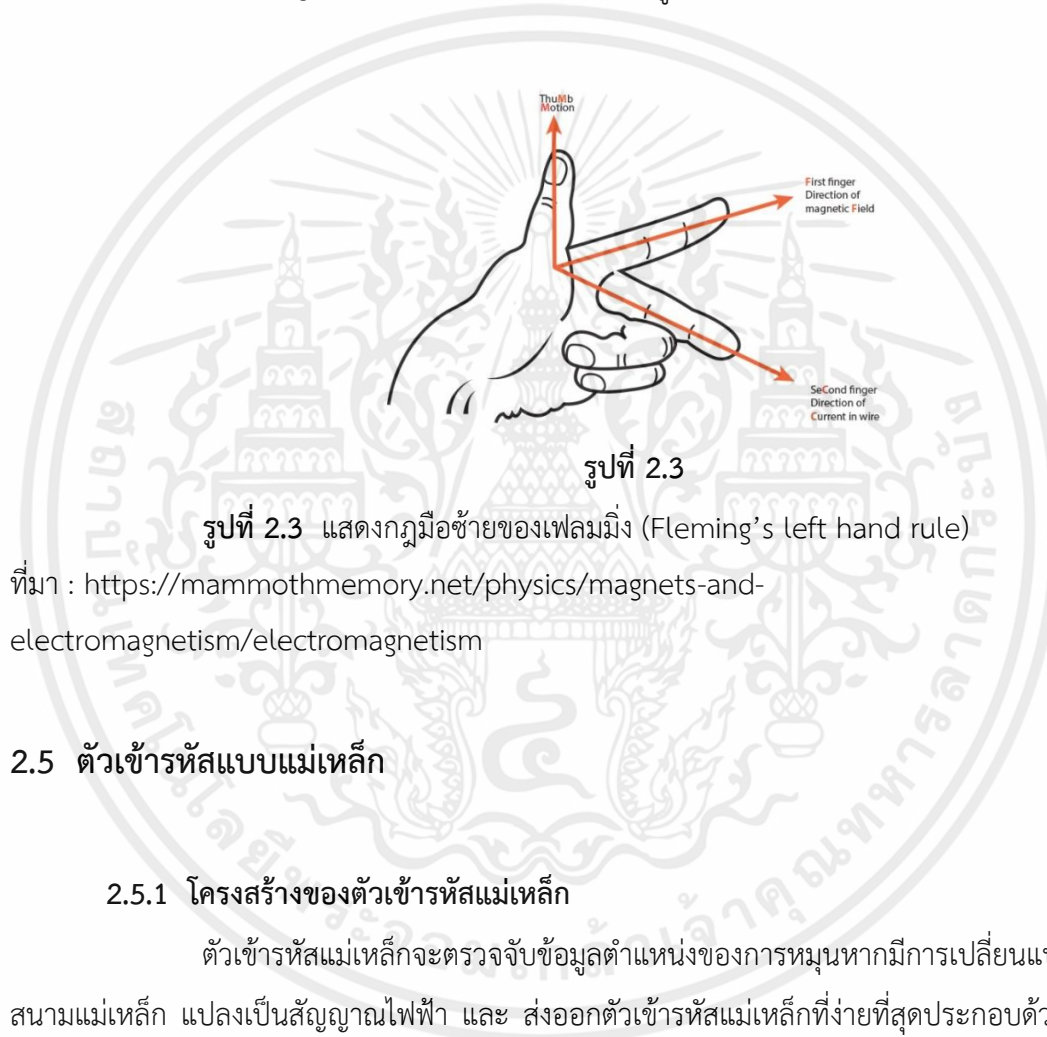
มอเตอร์ไฟฟ้ากระแสตรงเป็นอุปกรณ์ที่แปลงจากพลังงานไฟฟ้าเป็นพลังงานกล โดยประกอบด้วย แม่เหล็กถาวร 2 ขั้ว ซึ่งวางอยู่ระหว่างขดลวดตัวนำ ขดลวดตัวนำจะได้รับแรงดันไฟฟ้ากระแสตรงป้อนให้ในการทำงาน ทำให้เกิดอำนาจแม่เหล็ก 2 ขั้ว มีขั้วแม่เหล็กเหมือนกันวางใกล้กันเกิดแรงผลักดันทำให้ขดลวดตัวนำหมุนเคลื่อนที่ได้การทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง [3] แสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดงการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

ที่มา : <https://physics.stackexchange.com/questions/496925/potential-drop-across-inductor-vs-potential-drop-across-rotating-coils-in-b-fi>

เมื่อมีกระแสไหลผ่านเข้าไปในมอเตอร์กระแสจะแบ่งออกไป 2 ทาง คือ ส่วนที่หนึ่งจะผ่านเข้าไปที่ขดลวดสนามแม่เหล็ก (Field coil) ทำให้เกิดสนามแม่เหล็กขึ้นและอีกส่วนหนึ่งจะผ่านแปลงถ่านคาร์บอนและผ่านคอมมิวเตเตอร์ เข้าไปในขดลวดอาร์เมเจอร์ทำให้เกิดสนามแม่เหล็กขึ้นเช่นกัน ซึ่งทั้งสองสนามจะเกิดขึ้นขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กและจะไม่มีการติดกัน จะมีแต่ที่หักล้างและมีการเสริมกัน ซึ่งทำให้เกิดแรงบิดในอาร์เมเจอร์ ทำให้อาร์เมเจอร์หมุนซึ่งในการหมุนนั้นจะเป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (Fleming's left hand rule) ซึ่งแสดงดังรูปที่ 2.3



รูปที่ 2.3

รูปที่ 2.3 แสดงกฎมือซ้ายของเฟลมมิ่ง (Fleming's left hand rule)

ที่มา : <https://mammothmemory.net/physics/magnets-and-electromagnetism/electromagnetism>

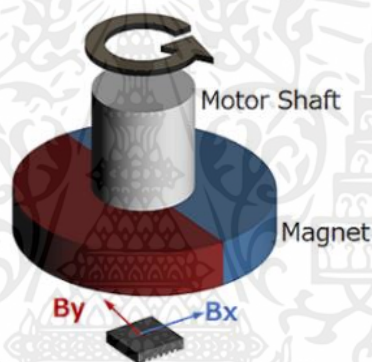
## 2.5 ตัวเข้ารหัสแบบแม่เหล็ก

### 2.5.1 โครงสร้างของตัวเข้ารหัสแม่เหล็ก

ตัวเข้ารหัสแม่เหล็กจะตรวจจับข้อมูลตำแหน่งของการหมุนหากมีการเปลี่ยนแปลง ของสนามแม่เหล็ก แปลงเป็นสัญญาณไฟฟ้า และ ส่งออกตัวเข้ารหัสแม่เหล็กที่ง่ายที่สุดประกอบด้วยแม่เหล็กถาวรและเซ็นเซอร์แม่เหล็ก แม่เหล็กถาวรจะติดอยู่กับปลายลำตัวหมุนเช่นเฟลามาเตอร์ และ เซ็นเซอร์แม่เหล็กจะได้รับการแก้ไขในสถานะที่ติดตั้งบนกระดาน PCB ในตำแหน่งที่ได้รับสนามแม่เหล็กที่สร้างขึ้นโดยแม่เหล็กถาวร และเมื่อแม่เหล็กถาวรติดกับเฟลามาเตอร์หมุนทิศทางของสนามแม่เหล็กที่ตรวจพบโดยเซ็นเซอร์แม่เหล็กจะเปลี่ยนไปเป็นผลให้ตัวเข้ารหัสตรวจจับตำแหน่งการหมุน และ ความเร็วของเฟลามาเตอร์ [4]

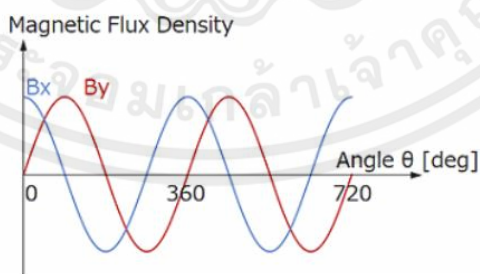
## 2.5.2 หลักการของการเข้ารหัสแม่เหล็ก

1. การตรวจจับการเคลื่อนไหวแบบหมุนเมื่อสนามแม่เหล็กเปลี่ยน และ แปลงเป็นสัญญาณไฟฟ้า เมื่อเพลามอเตอร์ทำการหมุนสนามแม่เหล็กที่สร้างขึ้นโดยแม่เหล็กถาวรที่ติดกับปลายเพล่าจะหมุนเช่นกัน ซึ่งในเวลาเดียวกันสนามแม่เหล็กจะทำการหมุนด้วยความแข็งแรงคงที่ในบริเวณใกล้กับศูนย์กลางของแกนหมุน องค์ประกอบฮอลล์ตรวจจับการเปลี่ยนแปลงของการกระจายสนามแม่เหล็กนี้ และ แปลงออกเป็นสัญญาณไฟฟ้า โดยองค์ประกอบฮอลล์จะเป็นเซ็นเซอร์แม่เหล็กที่สามารถตรวจจับความแข็งแรงของสนามแม่เหล็กในทิศทางเดียวเท่านั้น ดังนั้นเพื่อตรวจจับตำแหน่งการหมุนของระนาบการหมุน XY จำเป็นต้องมีองค์ประกอบ Hall สำหรับไว้ตรวจจับความแข็งแรงของส่วนประกอบแกน X ( $B_x$ ) และ องค์ประกอบ Hall สำหรับตรวจจับความแข็งแรงของส่วนประกอบแกน Y [4]



รูปที่ 2.4 ตัวเข้ารหัสแม่เหล็กการกำหนดค่า Shaft-End

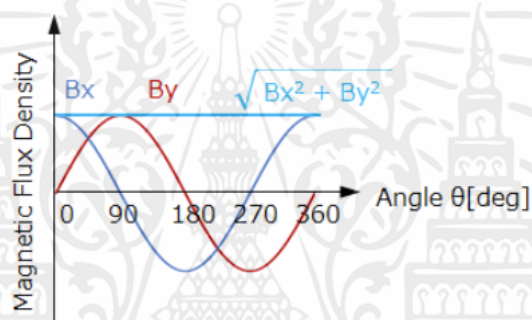
ที่มา : <https://www.akm.com/kr/ko/technology/technical-tutorial/basic-knowledge-encoder/magnetic-encoder/>



รูปที่ 2.5 ความแข็งแรงของสนามแม่เหล็กที่ตรวจพบโดยองค์ประกอบฮอลล์

ที่มา : <https://www.akm.com/kr/ko/technology/technical-tutorial/basic-knowledge-encoder/magnetic-encoder/>

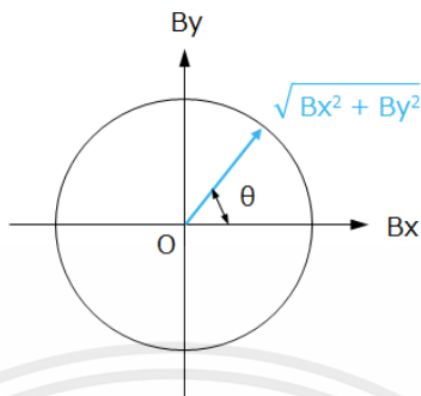
2. การแปลงสัญญาณไฟฟ้าขององค์ประกอบฮอลล์เป็นข้อมูลในหน่วยของมุม ข้อมูลสนามแม่เหล็กของส่วนประกอบแกน X และ ส่วนประกอบแกน Y จะถูกแปลงเป็นสัญญาณไฟฟ้าโดยองค์ประกอบจะถูกแปลงเป็นสัญญาณดิจิทัลโดยตัวแปลง AD นอกจากนี้ยังถูกส่งไปยังวงจรเลขคณิตและแปลงเป็นข้อมูลมุมโดยใช้ฟังก์ชันตรีโกณมิติ รูปประกอบที่ได้จากการสังเคราะห์ส่วนประกอบแกน X และ ออร์โธโกนาลส่วนประกอบแกน Y ซึ่งกันและกันเรียกว่ารูป Lissajous (หรือรูปคลื่น Lissajous) ภายใต้การกำหนดค่า Shaft-End โดยไม่จัดแนวผิดตำแหน่งความแข็งแรงของสนามแม่เหล็กหมุนจะไม่เปลี่ยนแปลง ดังนั้นรูป Lissajous จึงวาดวงกลมที่สมบูรณ์แบบ ดังนั้นผลลัพธ์เอาต์พุตมุมที่แปลงโดยฟังก์ชันตรีโกณมิติจึงมีข้อผิดพลาดเป็นศูนย์



รูปที่ 2.6 สนามแม่เหล็กของส่วนประกอบแกน X ( $B_x$ ) และส่วนประกอบแกน Y

ที่มา : <https://www.akm.com/kr/ko/technology/technical-tutorial/basic-knowledge-encoder/magnetic-encoder/>

เมื่อส่วนประกอบแกน X เป็น  $B_x$  และส่วนประกอบแกน Y คือ โดยมุมการหมุน  $\theta$  สามารถรับเป็นมุมสมบูรณ์โดยการคำนวณ  $\arctan$  สิ่งนี้สามารถแสดงผลได้ด้วยวิธีการสมบูรณ์ที่สามารถนำมาใช้ในส่วนที่ 3 นั่นคือตัวเข้ารหัสแม่เหล็กเป็นหลักเข้ารหัสตรวจจับมุมสมบูรณ์ นอกจากนี้โดยการแปลงข้อมูลมุมสมบูรณ์เป็นข้อมูล A-phase, B-phase และ Z-phase มันสามารถส่งออกเป็นสัญญาณพัลส์ในวิธีการที่เพิ่มขึ้นหรือวิธีการลอคสมบูรณ์ ในเวลานี้หากความละเอียดของข้อมูลมุมสมบูรณ์สูงพอก็เป็นไปได้ที่จะส่งสัญญาณชีพจรของความละเอียดใด ๆ เช่น 360 ppr, 1000 ppr, 2500 ppr และ 4000 ppr



รูปที่ 2.7 Lissajous รูปของความแข็งแรงสนามแม่เหล็ก

ที่มา : <https://www.akm.com/kr/ko/technology/technical-tutorial/basic-knowledge-encoder/magnetic-encoder/>

## 2.6 ทฤษฎีการควบคุมแบบพีไอดี (PID Controller)

เครื่องควบคุมหรือตัวควบคุม เป็นส่วนประกอบที่สำคัญของระบบควบคุมแบบอัตโนมัติคือ การใช้ตัวควบคุม (Controller) ทำหน้าที่ในการสั่งการ ด้วยการสร้างสัญญาณสำหรับการควบคุมกระบวนการต่าง ๆ ให้มีผลตอบสนองเป็นไปตามต้องการ ซึ่งสัญญาณดังกล่าวจะเป็นไปตามกฎ และรูปแบบของการควบคุมที่ผู้ควบคุมได้เลือก และกำหนดไว้ล่วงหน้านั้น ไม่ว่าจะเป็นการควบคุมด้วยมือหรืออัตโนมัติ โดยสัญญาณควบคุมจะขึ้นอยู่กับค่าความคลาดเคลื่อนของการควบคุม เพื่อนำค่าที่ได้นี้ไปสร้างสัญญาณเพื่อทำการควบคุมกระบวนการต่อไป โดยผลการควบคุมจะถูกวัดและป้อนกลับไปยังตัวควบคุมเพื่อทำการเปรียบเทียบตัดสิน ออกคำสั่ง และ ปรับใหม่ ซึ่งการกระทำจะดำเนินต่อไปซ้ำ ๆ กันเช่นนี้ตามลำดับ

ปัจจุบันเครื่องควบคุมสามารถแบ่งตามโครงสร้างการทำงานได้ 3 ชนิด คือเครื่องควบคุมแบบนิวเมติกส์ที่ทำงานโดยใช้สัญญาณลม เครื่องควบคุมแบบอิเล็กทรอนิกส์ที่ทำงานโดยใช้วงจรอิเล็กทรอนิกส์เชิงเส้น (Linear Circuit) ในการสร้างสัญญาณควบคุม และเครื่องควบคุมแบบดิจิทัลทำงานโดยใช้วงจรตรรกะ (Logic Circuit) หรือไมโครโปรเซสเซอร์สร้างสัญญาณควบคุม โดยเครื่องควบคุมที่ถูกนำมาใช้ และเป็นที่รู้จักกันดีมากที่สุดในการอุตสาหกรรมการผลิตในปัจจุบันก็คือการใช้ตัวควบคุมแบบ PID [5]

### 2.6.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี

ระบบควบคุมแบบ Closed-loop นั้นเป็นระบบควบคุมแบบหนึ่งซึ่งสัญญาณเอาต์พุตจะมีผลโดยตรงต่อการควบคุม ดังนั้นระบบควบคุมแบบ Closed-loop ก็คือ ระบบควบคุมแบบป้อนกลับ

สัญญาณค่าความคลาดเคลื่อนซึ่งเป็นสัญญาณแตกต่างระหว่างสัญญาณอินพุทกับสัญญาณป้อนกลับจะถูกป้อนให้ตัวควบคุมเพื่อที่จะได้ลดความคลาดเคลื่อนให้น้อยลง และ จะทำให้เอาต์พุทของระบบมีค่าตามที่ต้องการ ซึ่งสัญญาณที่ใช้ในการป้อนกลับนี้อาจจะเป็นสัญญาณเอาต์พุทโดยตรงหรือเป็นสัญญาณที่เป็นฟังก์ชันของสัญญาณเอาต์พุท โดยระบบควบคุมแบบป้อนกลับโดยทั่วไป ประกอบด้วยอุปกรณ์ต่าง ๆ ดังต่อไปนี้

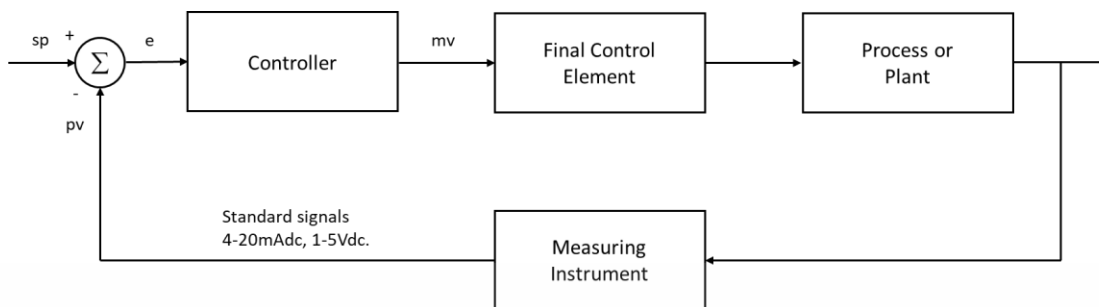
1. **ตัวควบคุม (Controller)** คือ เครื่องมือหรืออุปกรณ์ที่ใช้ในการสร้างสัญญาณควบคุม เพื่อทำหน้าที่ควบคุมให้ระบบ หรือกระบวนการที่ต้องการควบคุมเพื่อส่งสัญญาณควบคุมออกไปควบคุมโพรเซส ซึ่งเราสามารถจะตั้งเป้าหมายให้กับตัวควบคุมนี้ได้ โดยตัวควบคุมจะมีอยู่หลายแบบ เช่น ตัวควบคุมแบบ ON-OFF หรือตัวควบคุมแบบ Proportional (P), ตัวควบคุมแบบ Integral (I), ตัวควบคุมแบบ Derivative (D) หรือการใช้ตัวควบคุมหลาย ๆ แบบรวมกัน เช่น ตัวควบคุมแบบ PI, PD และ ตัวควบคุมแบบ PID

2. **กระบวนการ (Process or Plant)** คือ กระบวนการที่ถูกควบคุมหรือกระบวนการทางฟิสิกส์ที่เราต้องการควบคุมให้มีสถานะตามต้องการ ขณะสถานะการทำงานหรือสภาพแวดล้อมอาจเปลี่ยนแปลงอยู่ตลอดเวลาของกระบวนการซึ่ง ได้แก่ อุณหภูมิ, ความดัน, อัตราการไหล หรือ ระดับของของไหล เป็นต้น

3. **อุปกรณ์ควบคุมสุดท้าย (Final Control Element)** คือ อุปกรณ์ที่ทำหน้าที่ปรับสถานะของกระบวนการด้วยสถานะของกระบวนการเปลี่ยนแปลงตามค่าสัญญาณควบคุม (Manipulated Variable) เช่น วาล์วควบคุม (Control Valve), อินเวอร์เตอร์ (Inverter), Actuator ต่าง ๆ เป็นต้น

4. **อุปกรณ์วัด (Measuring Instruments)** คือ อุปกรณ์ซึ่งได้แก่ เซนเซอร์ (Sensors), ทรานสดิวเซอร์ (Transducer), อุปกรณ์แปลงสัญญาณ (Converter) หรือวัดสัญญาณอื่น ๆ ที่มีเอาต์พุทตามสัญญาณมาตรฐาน

5. **สัญญาณมาตรฐาน (Standard Signal)** คือ การเชื่อมต่ออุปกรณ์ในระบบให้สามารถควบคุมแบบอัตโนมัติและให้ทำงานได้ตามต้องการซึ่งจำเป็นต้องมีมาตรฐานรองรับ ระบบควบคุมอัตโนมัตินั้นจะมีการเปลี่ยนแปลงระบบตัวอุปกรณ์ทางเครื่องมือวัดมาตั้งแต่ยุคการใช้ลม ไฟฟ้า และ ปัจจุบันมีการใช้สัญญาณดิจิทัล สัญญาณลม 3-15 psi, สัญญาณไฟฟ้า 1-5 VDC หรือ 4-20 mA ซึ่งแสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป

ที่มา : JSS (2013 : 1) การทดลอง CS06: ลักษณะสมบัติของตัวควบคุม PID

การควบคุมกระบวนการทางอุตสาหกรรม โดยทั่วไปนิยมใช้ตัวควบคุมแบบพีไอดี (PID) เพราะรูปแบบของตัวควบคุมเป็นตัวควบคุมที่สามารถควบคุมกระบวนการต่าง ๆ ได้อย่างกว้างขวางเนื่องจากมีโครงสร้างการทำงานที่ไม่ซับซ้อน สามารถเข้าใจได้ง่าย การใช้งานตัวควบคุมพีไอดีนี้ขึ้นอยู่กับปรับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสม เพื่อให้ได้ผลตอบสนองของกระบวนการตามที่ต้องการ

ตัวควบคุม PID ประกอบด้วยตัวควบคุมแบบ Proportional (P), ตัวควบคุมแบบ Integral (I), ตัวควบคุมแบบ Derivative (D) ซึ่งมีฟังก์ชันถ่ายโอน (Transfer function) ดังนี้

$$u(t) = K_p(e(t)) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \quad (2.11)$$

โดยที่  $K_p$  คือ ค่าอัตราขยายของตัวควบคุมแบบ P (Proportional Gain)  
 $T_i$  คือ ค่าเวลา Integral (Integral Time)  
 $T_d$  คือ ค่าเวลา Derivative (Derivative Time)

จากรูปที่ 10 จะเห็นได้ว่าสัญญาณควบคุมหรือ ตัวแปรปรับกระบวนการ ( $mv$ ) ที่ได้จากตัวควบคุม PID จะถูกกำหนดด้วยความสัมพันธ์ระหว่างค่าสัญญาณขาเข้าด้วยตัวควบคุมกับตัวแปรกระบวนการ ( $pv$ ) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย ( $sp$ ) โดยที่ความสัมพันธ์ดังกล่าวจะขึ้นอยู่กับกฎเกณฑ์การควบคุมที่ผู้ควบคุมปรับแต่งไว้ล่วงหน้า

โดย  $u(t)$  คือสัญญาณควบคุม  $e(t)$  คือค่าความคลาดเคลื่อนของสัญญาณออกจากค่ากำหนด ตัวควบคุม PID ประกอบไปเทคนิคการควบคุมพื้นฐาน 3 แบบ แบบสัดส่วน (Proportional หรือ P) แบบอินทิกรัล (Integral หรือ I) และแบบอนุพันธ์ (Derivative หรือ D) แต่ละแบบสามารถนำมาประกอบกันเพื่อให้ได้ตัวควบคุมตามที่ต้องการ ซึ่งตัวควบคุมมีพารามิเตอร์ 3 ตัว คือค่าอัตราขยายแบบสัดส่วน ( $K_p$ ) ค่า

Integral time ( $T_i$ ) และ Derivative time ( $T_d$ ) โดยรายละเอียดของแต่ละแบบจะแสดงดังต่อไปนี้

## 2.6.2 Proportional Action

การควบคุมแบบสัดส่วนเป็นเทคนิคที่ง่ายที่สุด หลักการคือสัญญาณควบคุม  $u(t)$  จากตัวควบคุมที่ส่งไปปรับกระบวนการซึ่งมีค่าเป็นสัดส่วนกับความคลาดเคลื่อน สามารถเขียนได้ดังสมการ 2.12

$$u(t) = K_p(e(t)) \quad (2.12)$$

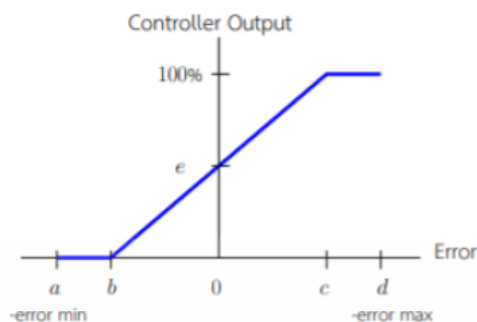
โดยที่  $K_p$  คือค่าอัตราขยายและ

$$e(t) \triangleq \text{ความคลาดเคลื่อน} = \text{ค่ากำหนด} - \text{ค่าวัด} \quad (2.13)$$

ตัวควบคุมบางตัวสัญญาณเข้าและสัญญาณออกอาจมีหน่วยต่างกัน เช่นการเปลี่ยนแปลงของอุณหภูมิที่ส่งผลทำให้เกิดการเปลี่ยนแปลงความดัน เพื่อหลีกเลี่ยงการแปลงหน่วยความสัมพันธ์ระหว่างสัญญาณออกและสัญญาณเข้าของตัวควบคุมอาจแสดงเป็นแถบสัดส่วน (Proportional Band หรือ %PB) โดยที่แถบสัดส่วนคือพิสัยของสัญญาณเข้าที่ทำให้ตัวควบคุมปฏิบัติงานเต็มพิสัยการทำงาน หรือถ้ามองจากตัวควบคุม แถบสัดส่วนคือช่วงความคลาดเคลื่อนที่ทำให้สัญญาณออกของตัวควบคุมเปลี่ยนแปลงจากค่าสูงสุดไปต่ำสุดโดยแสดง เป็นเปอร์เซ็นต์ของพิสัยสัญญาณเข้าตัวควบคุม และ ความสัมพันธ์ระหว่างอัตราขยายและเปอร์เซ็นต์แถบสัดส่วนคือ

$$K_p = \frac{100}{\%PB} \quad (2.14)$$

ลักษณะสมบัติของการควบคุมแบบสัดส่วนแสดงไว้ในรูปที่ 2.9



รูปที่ 2.9 ลักษณะสมบัติของตัวควบคุมแบบสัดส่วน

ที่มา : JSS (2013 : 2) การทดลอง CS06: ลักษณะสมบัติของตัวควบคุม PID

จากรูปที่ 2.9 ค่าเปอร์เซ็นต์แถบสัดส่วน (%PB) คือระยะ  $bc$  แม้ความคลาดเคลื่อนเป็นศูนย์ ยังมีสัญญาณค่าหนึ่ง ออกจากตัวควบคุมที่ป้อนให้กับกระบวนการ ค่านี้ทำหน้าที่เป็น ไบแอส (bias) ทำให้ระบบทำงานที่จุดทำงานต่อไปได้ โดยทั่วไปสัญญาณค่านี้มักจะถูกตั้งให้เท่ากับ 50% ของสัญญาณขาออกสูงสุดของตัวควบคุม

$$\text{สัญญาณออก} = \frac{\% \text{ ความคลาดเคลื่อน}}{\% \text{ แถบสัดส่วน}} + 50 \% \quad (2.15)$$

### 2.6.3 Integral Action

ผลตอบของการควบคุมแบบสัดส่วนร่วมกับการควบคุมแบบอินทิกรัล อธิบายได้ดังสมการที่ 2.16

$$u(t) = K_p(e(t) + \frac{1}{T} \int_0^t e(\tau) d\tau) \quad (2.16)$$

เมื่อ  $K_p$  คืออัตราขยาย,  $e(t)$  คือความคลาดเคลื่อน และ  $T_i$  คือ Integral time (วินาที)

เมื่อเปรียบเทียบกับสมการของตัวควบคุมแบบสัดส่วน ความแตกต่างอยู่ตรงที่เทอมไบอัสนั่นคือตัวควบคุมแบบสัดส่วนถูกจำกัดด้วยส่วนไบอัสเป็นค่าคงที่ ส่วนการควบคุมแบบอินทิกรัล มีการสะสมความคลาดเคลื่อนในการปรับแต่งไบอัส (นั่นคือทำหน้าที่เป็นตัวอินทิกรัล) และ จะหยุดสะสมเมื่อความคลาดเคลื่อนของระบบเป็นศูนย์ เมื่อผลตอบเข้าที่สมบูรณ์แล้ว เทอมไบอัสของระบบจะมีค่าน้อยเพียงใดขึ้นอยู่กับลักษณะของการรบกวน (Disturbance) การทำงานในลักษณะเช่นนี้มีลักษณะคล้ายกับฟังก์ชันรีเซ็ต ด้วยมือ (Manual-reset function) ดังนั้นในบางครั้งจึงเรียกตัวอินทิกรัลว่า ฟังก์ชันรีเซ็ต

(Reset function) คุณสมบัติของตัวอินทิกรัลในการกำจัดความคลาดเคลื่อน (หรือออฟเซต) เป็นข้อดีอย่างมาก จึงเป็นที่นิยมใช้กับระบบควบคุมป้อนกลับ อย่างไรก็ตาม ตัวอินทิกรัลก็มีข้อเสีย นั่นคือทำให้เกิดการล่าช้า และทำให้ช่วงเวลาของการแกว่งยาวนานขึ้น โดยทั่วไป ระบบแบบสัดส่วนรวมกับอินทิกรัล จะมีช่วงเวลาของการแกว่งยาวนานกว่าระบบเชิงสัดส่วนอย่างเดียว 50% หรือ  $TPI = 1.5TP$  สำหรับระบบที่มีค่าคงตัวเวลา (Time constant) น้อย (เช่น ระบบควบคุมอัตราการไหล) ปัญหาจะไม่มีผลมากนัก แต่สำหรับระบบที่มีค่าคงตัวเวลามาก (เช่น ระบบควบคุมระดับ) ปัญหานี้จะมีผลมาก จนทำให้ระบบเข้าสู่จุดวิกฤติที่ไม่สามารถยอมรับได้

การควบคุมแบบอินทิกรัล มีลักษณะเช่นเดียวกับการควบคุมสัดส่วนตรงผลกระทบของการเพิ่มอัตราขยายของตัวควบคุมหากอัตราขยายมีค่ามากเกินไปจะทำให้ผลตอบของระบบมีการแกว่ง โดยทั่วไป Integral time ( $T_i = 1/K_i$  sec โดยที่  $K_i = \text{repeats/sec}$ ) เป็นตัวแสดงว่า อัตราการตอบสนองของกระบวนการต่อสัญญาณการควบคุม ค่า  $T_i$  ที่น้อยกว่า จะทำให้ตัวควบคุมมีการตอบสนองที่เร็วกว่าในระยะเริ่มต้น โดยที่ความคลาดเคลื่อนยังเป็นค่าบวกอยู่ ดังนั้นกว่าความคลาดเคลื่อนจะเป็นศูนย์ (ซึ่งทำให้เทอม  $\int_0^t e(t) d\tau$  หดทำงาน) เทอมไบแอสก็จะมีค่าสูงกว่าที่ต้องการดังนั้นผลตอบสนองจึงเกิดส่วนพุ่งเกิน (Overshoot) สูงกว่าค่ากำหนด เป็นผลให้ตัวอินทิกรัลทำหน้าที่ปรับให้ความคลาดเคลื่อนมีค่าลดลง การใช้ตัวอินทิกรัลในการควบคุม ควรระวังในเรื่องของความคลาดเคลื่อนขนาดใหญ่ (เช่น เกิดการเปลี่ยนแปลงค่ากำหนดขนาดใหญ่) เพราะจะทำให้เกิดปัญหา Integral windup ถึงแม้ว่า  $T_i$  มีค่าถูกต้องในสภาวะการทำงานธรรมดาแต่สัญญาณควบคุมอาจถึงจุดอิ่มตัวขณะผลตอบเกิดส่วนพุ่งเกิน

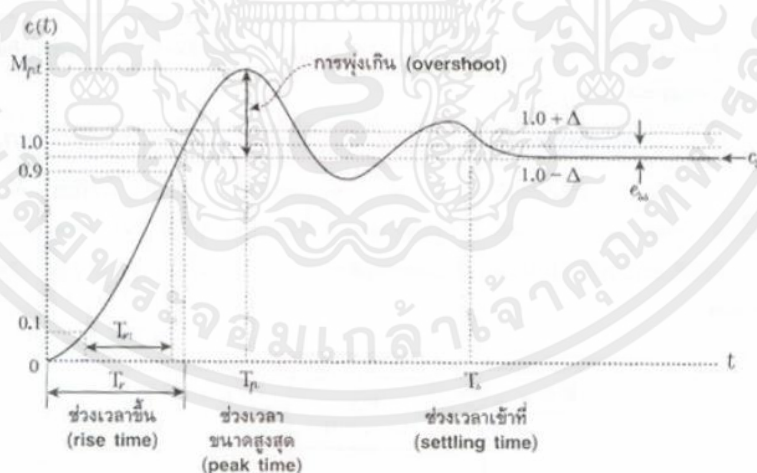
#### 2.6.4 Derivative Action

ตัวควบคุมแบบสัดส่วน และ แบบอินทิกรัลต่างก็มีข้อจำกัดอยู่ที่ความคลาดเคลื่อนขนาดใหญ่ ซึ่งเป็นปัญหาต่อการควบคุมกระบวนการแต่ความคลาดเคลื่อนขนาดใหญ่นี้ สามารถรู้ได้ล่วงหน้าโดยพิจารณาจากแนวโน้มของความคลาดเคลื่อน หรืออัตราการเปลี่ยนแปลงของสัญญาณนั่นเอง ตัวอนุพันธ์มีหลักการทำงาน คือ ตัวควบคุมตอบสนองต่ออัตราการเปลี่ยนแปลงของความคลาดเคลื่อน ถึงแม้ว่าความคลาดเคลื่อนมียังค่าเล็กน้อย สัญญาณออกของตัวอนุพันธ์ ไม่ได้สัมพันธ์กับขนาดของความคลาดเคลื่อน แต่ขึ้นอยู่กับอัตราการเปลี่ยนแปลงของความคลาดเคลื่อน ถ้าความคลาดเคลื่อนมีค่าคงที่ ตัวอนุพันธ์จะให้สัญญาณออกเป็นศูนย์ คุณสมบัติข้อนี้มีผลดีคือ ตัวควบคุมจะมีผลตอบสนองที่เกิดก่อนที่ความคลาดเคลื่อนจะเพิ่มมากขึ้น และทำให้ระบบมีผลตอบสนองเร็วขึ้น ตัวควบคุมแบบอนุพันธ์สามารถเขียนดังสมการที่ 2.17

$$u(t) = K_p(e(t) + T_d(\frac{de(t)}{dt})) \quad (2.17)$$

โดย Derivative time ( $T_d$ ) เป็นเวลาที่แสดงถึงผลตอบสนองเนื่องจากตัวอนุพันธ์ การเพิ่ม Derivative time จะทำให้ผลตอบสนองของตัวอนุพันธ์มีค่ามากขึ้น เนื่องจากตัวอนุพันธ์มีความไวต่อการเปลี่ยนแปลงมาก ดังนั้นจึงนิยมใช้กับค่าที่วัดได้เท่านั้น แต่ไม่ใช้กับค่ากำหนด เพราะการเปลี่ยนค่ากำหนดมักจะเป็นแบบขั้น (Step) ทำให้ผลตอบสนองของตัวอนุพันธ์เป็นพัลส์ และทำให้เกิดการกระแทก (Bump) ของอุปกรณ์ในกระบวนการสำหรับค่ากำหนดใช้เฉพาะกับตัวควบคุมสัดส่วนและอินทิกรัลตัวอนุพันธ์ คือ ตัวควบคุมที่ก่อให้เกิดผลตรงข้ามกับตัวอินทิกรัล ดังนั้น จึงสามารถใช้ในการปรับปรุงกระบวนการที่มีการล่าช้าทางเวลา (Time lag) มาก ๆ ทำให้ผลตอบสนองรวดเร็วขึ้น และช่วงเวลากว้างที่สั้นลง ข้อเสียของตัวอนุพันธ์ คือ มีความไวต่อสัญญาณรบกวนเป็นอย่างมาก เพราะมีผลตอบสนองโดยตรง ต่ออัตราการเปลี่ยนแปลงของสัญญาณที่วัดได้ ดังนั้นแม้ว่าสัญญาณรบกวนจะมีขนาดเล็กก็ตาม แต่ก็อาจก่อให้เกิดการเปลี่ยนแปลงต่อสัญญาณออกของตัวควบคุม ดังนั้นจึงเป็นไปได้ที่จะใช้ตัวอนุพันธ์ในการควบคุมผลของสัญญาณรบกวน

### 2.6.5 ผลการตอบสนองของระบบ



รูปที่ 2.10 ผลการตอบสนองของระบบ

ที่มา : <https://sites.google.com/site/suppapongclub/ceaa-luk-rabbpid>

### 2.6.5.1 Rise time

ช่วงเวลาดำเนิน (Rise Time,  $T_r$ ) หมายถึง ช่วงเวลาที่ผลตอบสนองเมื่อสัญญาณเอาต์พุตเพิ่มจาก 10% จนถึง 90% หรือจาก 5% ถึง 95% หรือจาก 0% ถึง 100% ดังนั้นการกำหนดช่วงเวลาดำเนิน จำเป็นต้องบอกด้วยว่าวัดโดยใช้ช่วงเวลาไหน [6]

### 2.6.5.2 Peak Time

เวลาของค่ายอด (Peak Time,  $T_p$ ) หมายถึง เวลาที่สัญญาณผลการตอบสนองมีค่าสูงสุดค่าแรกของการตอบสนองนั้น

### 2.6.5.3 Maximum percent overshoot

โอเวอร์ชูดสูงสุด (Maximum percent overshoot,  $M_p$ ) หมายถึง ค่าการตอบสนองสูงสุดที่วัดจากสถานะอยู่ตัวสุดท้าย (Final Steady State) การบอกค่าโอเวอร์ชูดสูงสุดมักจะบอกเป็นเปอร์เซ็นต์ของ Maximum percent overshoot ซึ่งจะเป็นค่าแตกต่างสูงสุด ระหว่างอินพุต และเอาต์พุตของระบบ ถ้าผลสนองของระบบมีค่าที่ภาวะคงที่ไม่เท่ากับหนึ่งแล้วจะได้ว่า เป็นไปดังสมการที่ 2.18

$$\text{Maximum percent overshoot} = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2.18)$$

### 2.6.5.4 Settling time

เวลาเข้าที่ (Settling Time,  $T_s$ ) หมายถึง เวลาที่ผลการตอบสนองลดลงจนเริ่มเข้าไปอยู่ในช่วงที่กำหนด ซึ่งจะวัดเทียบกับค่าสุดท้ายของผลการตอบสนองในสถานะอยู่ตัวสุดท้าย (Final Steady State) ค่าที่นิยมกำหนดสำหรับช่วงนี้มักจะบอกเป็นเปอร์เซ็นต์ เช่น 1%, 2% หรือ 5% เป็นต้น

## 2.7 ทฤษฎีการประมวลผลภาพ

เป็นกระบวนการจัดการและวิเคราะห์รูปภาพให้เป็นข้อมูลในแบบดิจิทัล โดยใช้คอมพิวเตอร์การเพื่อให้ได้ข้อมูลที่เรากำลังต้องการทั้งในเชิงคุณภาพและปริมาณ (ขนาด รูปร่าง) โดยมีขั้นตอนต่าง ๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการ

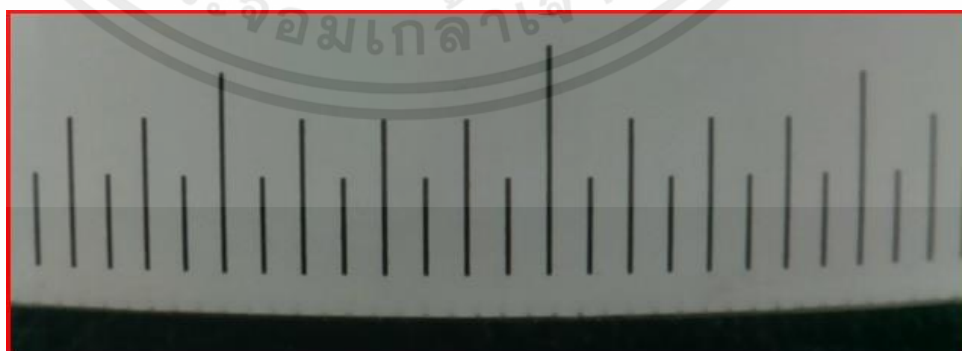
เคลื่อนของวัตถุในภาพ จากนั้นสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ [7]

### 2.7.1 Region-of-interest (ROI)

Region-of-interest (ROI) คือบริเวณที่เราสนใจ อาจจะเป็นบริเวณใดภายในภาพก็ได้ โดยการตีกรอบล้อมรอบบริเวณที่สนใจ ด้วยวงกลม กรอบสี่เหลี่ยม หรือกรอบรูปเหลี่ยมใด ๆ เพื่อนำภาพเฉพาะส่วนดังกล่าวมาประมวลผล โดยไม่มีผลกระทบต่อส่วนอื่น ๆ ซึ่งใน 1 ภาพ สามารถกำหนดได้หลาย ๆ ROI เมื่อกำหนดตำแหน่งต่าง ๆ แล้ว จะสร้าง Mask ที่เป็น Binary Mask สำหรับใช้กำหนดขอบเขตที่จะมีการเปลี่ยนแปลงภายในรูปภาพ โดยให้ค่าส่วนที่สนใจเป็น 1 หรือสีขาว และให้ส่วนอื่น ๆ เป็น 0 หรือสีดำ[8]



รูปที่ 2.11 แสดงภาพต้นฉบับก่อนทำการใช้ฟังก์ชัน ROI



รูปที่ 2.12 แสดงภาพหลังทำการใช้ฟังก์ชัน ROI

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.7.2 RGB Image

RGB Image หรือ Truecolor Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด  $m \times n \times 3$  โดยที่  $m$  คือความยาว และ  $n$  คือความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือสีแดง(Red) สีเขียว(Green) และ สีน้ำเงิน(Blue) ซึ่งระบบสี RGB เป็นระบบสีที่ประกอบด้วยแม่สีหลักสามสีคือ แดง(Red), เขียว(Green) และ น้ำเงิน(Blue) RGB นั้นเป็นระบบสีแสง และเป็นแบบ Additive คือ ถ้าไม่มีสีใดเลยจะมองเห็นเป็นสีดำ และในทางกลับกัน หากมีครบทุกสีจะมองเห็นเป็นสีขาว จะต่างกับระบบสีแบบ Subtractive หรือระบบสีแบบ CMYK ที่เป็นสีที่เกิดจากการสะท้อน

### 2.7.3 Grayscale Image

ภาพระดับสีเทาเป็นภาพซึ่งค่าในแต่ละจุดภาพ คือ ค่าความเข้มของสีแต่ละตำแหน่งของจุดภาพนั้น ซึ่งค่าที่เป็นไม่ได้ของภาพระดับสีเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่าง เช่น ภาพระดับสีเทา 8 บิตที่ระดับสีทั้งหมด 256 ระดับ โดยนิยมระบุในช่วง 0-1 หรือ 0-255 แสดงระดับสีเทา [9] ดังรูปที่ 2.13



รูปที่ 2.13 แสดงระดับสีเทา 256 ระดับ

ที่มา : <http://babsoftsoft.blogspot.com/2015/10/image-processing-convert-gray-scale.html>

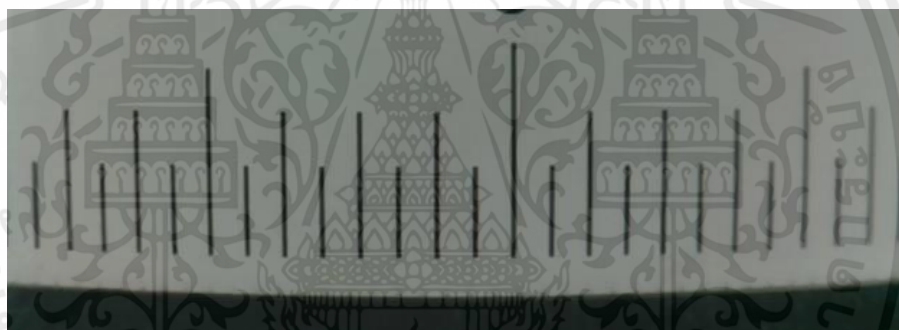
ภาพระดับสีเทาจะมีการไล่ระดับความอ่อนแก่ของสี ซึ่งจะอยู่ระหว่างสีขาว และ สีดำ Halftone Image อย่างต่อเนื่อง โดยที่ค่าแต่ละพิกเซล ของภาพจะหมายถึงความเข้มแสงแต่ละตำแหน่งของพิกเซล ที่อยู่ในรูประดับสีเทา การเปลี่ยนภาพจากระบบสี อาร์จีบี (RGB) เป็นระดับสีเทา (Gray Scale) จะใช้สมการดังนี้ในการเปลี่ยน

$$Y = 0.3R + 0.59G + 0.11B \quad (2.19)$$

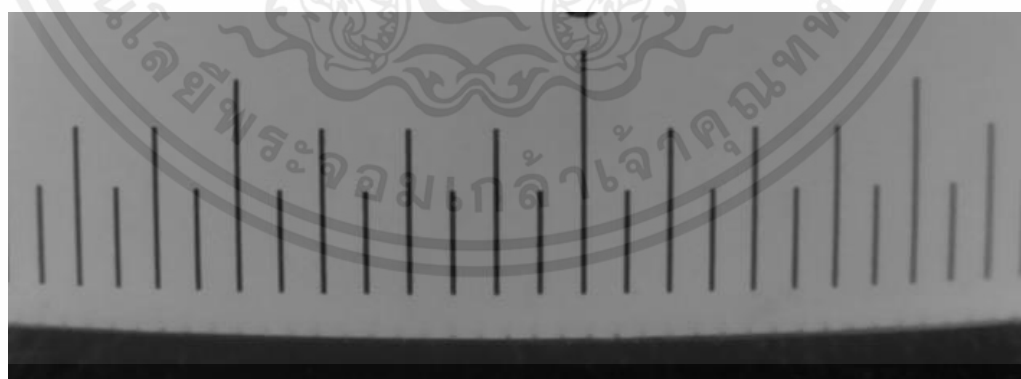
โดย Y แทน ค่าระดับสีเทา ณ จุดพิกเซล ที่ต้องการ R แทน ค่าสีแดง ณ จุด ที่ต้องการหา G แทน ค่าสีเขียว ณ จุด ที่ต้องการหา B แทน ค่าสีน้ำเงิน ณ จุด ที่ต้องการหา

### 2.7.3.1 การแปลงภาพสี RGB ให้เป็นภาพระดับเทา (RGB to Gray)

ภาพที่รับเข้ามาในขั้นตอนแรกเป็นภาพที่อยู่ในระบบปริภูมิสีแบบ RGB ดังนั้นแต่ละพิกัดของภาพจะประกอบด้วยค่าของเซตที่แสดงถึง ค่าของ R ค่าของ G และ B จากนั้นระบบจะทำการเปลี่ยนภาพสี RGB ให้เป็นภาพระดับสีเทา (Grayscale) เพื่อให้สามารถวิเคราะห์ภาพได้ง่ายขึ้นเพราะเมื่อแปลงภาพเป็นระดับสีเทาแล้วจะทำให้แต่ละจุดภาพของภาพจะเหลือเพียงค่าความเข้มของสีมีค่าตั้งแต่ 0 ถึง 255 เมื่อแปลงจากภาพสี RGB เป็นภาพระดับสีเทา (Grayscale) จะสามารถแสดงได้ดังรูปที่ 2.14 ซึ่งแสดงภาพต้นฉบับ



รูปที่ 2.14 แสดงภาพต้นฉบับ



รูปที่ 2.15 แปลงจากภาพสีเป็นภาพระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.7.3.2 แปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ

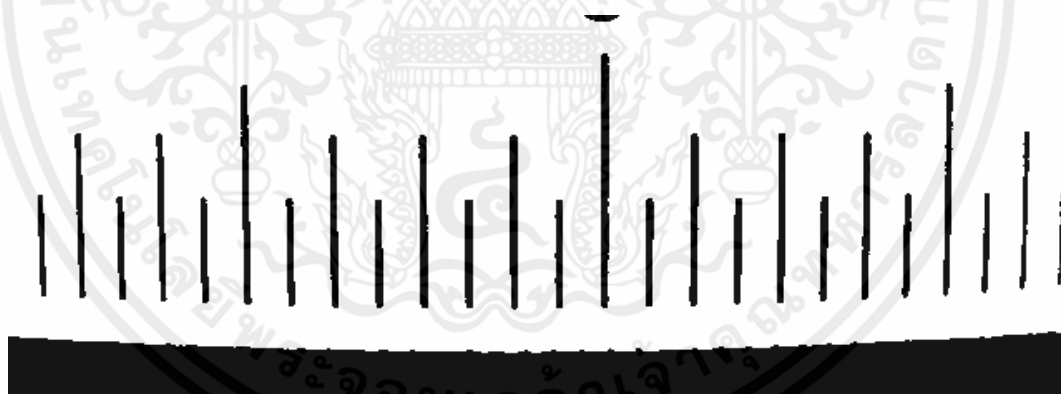
หมายถึง ภาพที่ประกอบด้วยสีขาวและสีดำเป็นหลัก ในทางดิจิทัลหมายความว่ามีความมีเพียง 2 สถานะ คือ 0 และ 1 ถ้าพิกเซลใดมีค่าเป็น 0 หมายความว่าพิกเซลนั้นมีสีดำ พิกเซลใดมีค่าเป็น 1 หมายความว่าพิกเซลนั้นมีสีขาว การแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ จะต้อง กำหนดค่าความเข้มที่ต้องการอ้างอิงหรือค่าขีดแบ่ง (Threshold Value) โดยผู้ใช้งานสามารถกำหนด ได้เองหรือใช้อัลกอริทึมในการหาค่า

### 2.7.4 Threshold

เป็นหลักการที่ใช้ค่าคงที่ค่าหนึ่ง ในการเปรียบเทียบกับค่าของ Pixel ในแต่ละพื้นที่ ถ้าค่าของ Pixel ในพื้นที่นั้นมีค่าน้อยกว่าค่าคงที่ ก็จะเปลี่ยนค่า Pixel ของพื้นที่นั้นเป็น 0 แต่ถ้าค่าของ Pixel ในพื้นที่นั้นมีค่ามากกว่าก็จะเปลี่ยนค่า Pixel ของพื้นที่นั้นเป็น 255 โดยภาพ input จะต้องเป็นภาพ Grayscale

$$g(x, y) = 0 \quad \text{if } f(x, y) \geq \text{Threshold value} \quad (2.20)$$

$$g(x, y) = 255 \quad \text{if } f(x, y) < \text{Threshold value} \quad (2.21)$$



รูปที่ 2.16 แสดงภาพที่ผ่านการ Threshold และแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ

## 2.8 ระบบปฏิบัติการ Raspbian



รูปที่ 2.17 Raspberry Pi OS

ที่มา : <https://www.raspberrypi.org/software/>

Raspbian เป็นระบบปฏิบัติการที่มีรากฐานมาจากระบบปฏิบัติการ Debian ซึ่งแตกตัวออกมาจากระบบปฏิบัติการ Linux โดยได้รับการออกแบบให้รันบนฮาร์ดแวร์ Raspberry Pi ดังนั้น คำสั่งต่าง ๆ ของ Raspbian จึงเหมือนกับคำสั่งที่มีใช้งานบนระบบปฏิบัติการ Linux รวมทั้ง Ubuntu เกือบทั้งหมด จะมีแตกต่างกันบ้างเล็กน้อยเท่านั้น Raspbian มีให้เลือกติดตั้งอยู่ 2 แบบคือ Raspbian Jessie with PIXEL และ Raspbian Jessie Lite เนื่องจาก Raspbian Jessie with PIXEL (ใช้เนื้อที่ติดตั้งประมาณ 4 GB) มีซอฟต์แวร์ต่าง ๆ ติดตั้งมาพร้อมใช้งานจำนวนหนึ่ง เช่น LibreOffice Writer (Word), LibreOffice Calc (Excel), LibreOffice Impress (Power Point), LibreOffice Base (Access), Chromium Web Browser, Games, Programming Language: Python, Node-RED และ ซอฟต์แวร์อื่นๆ [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานในการการออกแบบตัวควบคุมพีไอดีสำหรับชุดแทนเลียนในการสอบเทียบสายเทปวัดระยะ แสดงดังรูปที่ 3.1

ขั้นตอนที่ 1 : เริ่มต้นและศึกษาองค์ประกอบของกระบวนการ
ขั้นตอนที่ 2 : ออกแบบและทำการสร้าง Hardware
ขั้นตอนที่ 3 : ทำการเขียนโปรแกรม
ขั้นตอนที่ 4 : ทำการหาค่าพารามิเตอร์การควบคุมพีไอดี
ขั้นตอนที่ 5 : ทดสอบความสามารถในการควบคุม
ขั้นตอนที่ 6 : สรุปผล
ขั้นตอนที่ 7 : สิ้นสุด

รูปที่ 3.1 แสดงขั้นตอนการดำเนินงาน

#### 3.2 โครงสร้างและการออกแบบชุดแทนเลียนสำหรับการสอบเทียบสายเทปวัดระยะ

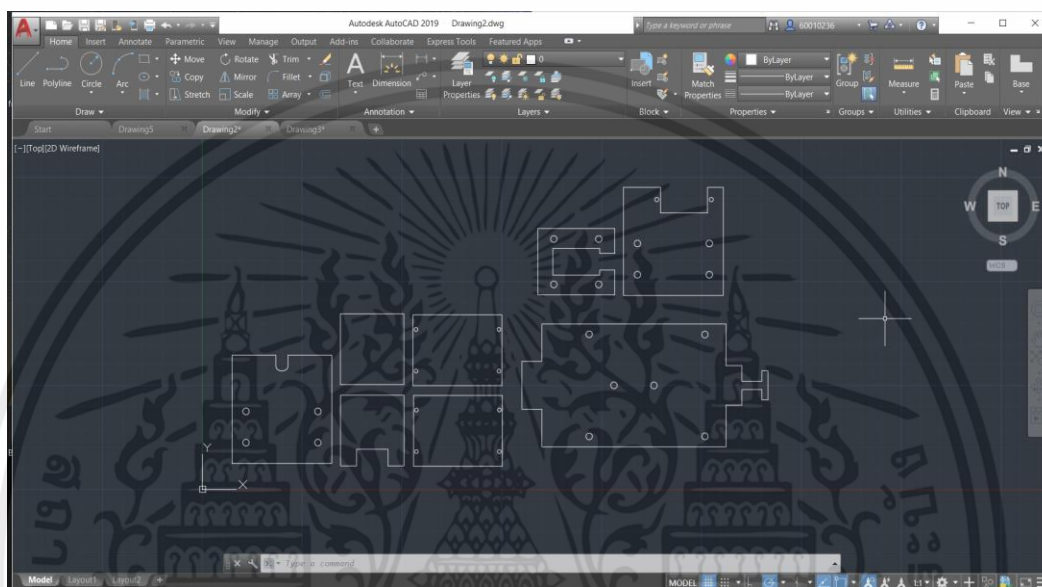
ในหัวข้อนี้อธิบายรายละเอียดของระบบสอบเทียบสายเทปวัดระยะ ซึ่งประกอบไปด้วยการออกแบบโครงสร้างและองค์ประกอบทางฮาร์ดแวร์ (Hardware) รวมถึงการใช้งานซอฟต์แวร์ที่เกี่ยวข้อง ดังนี้

##### 3.2.1 ชุดแทนเลียนสำหรับการสอบเทียบสายเทปวัดระยะโมเดลที่ 1

โครงสร้างของระบบสอบเทียบสายเทปวัดระยะประกอบขึ้นจาก อลูมิเนียมโปรไฟล์ ประกอบกัน และมีแทนเลียน ซึ่งใช้ล้อ (V-Wheel) สำหรับยึดบนอลูมิเนียมโปรไฟล์และยึดแทนเลียนเข้ากับสายพานที่ถูกต้องเข้ากับมอเตอร์ ทำให้แทนสามารถเลื่อนบนอลูมิเนียมโปรไฟล์ได้ พร้อมทั้งมีชุดสำหรับ

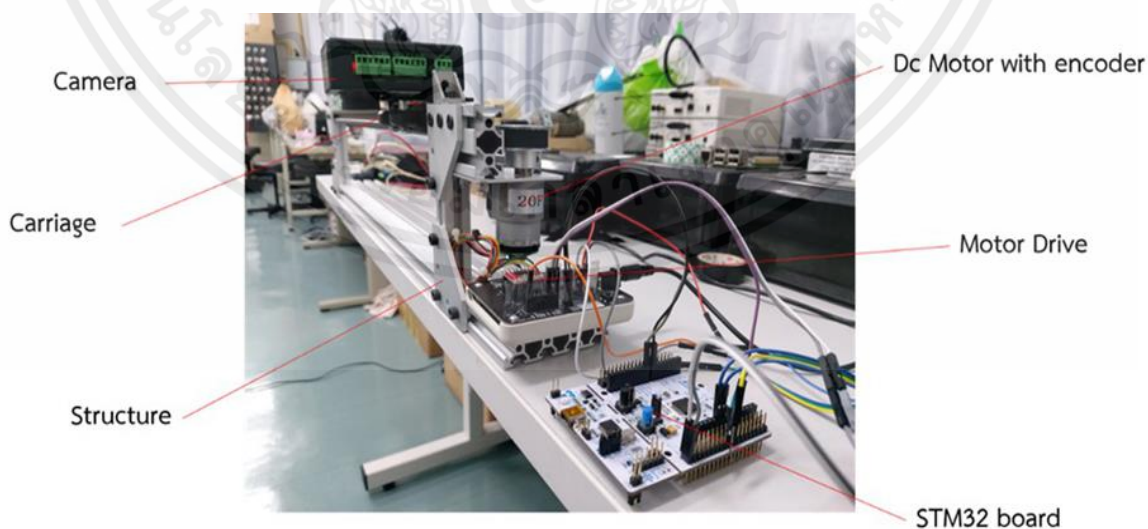
ควบคุมการสั่นบนแท่นเลื่อน โดยองค์ประกอบทางฮาร์ดแวร์แสดงดังรูปที่ 3.3 และ การออกแบบซึ่งประกอบไปด้วยอุปกรณ์ต่าง ๆ ดังต่อไปนี้

### 3.2.1.1 การออกแบบชุดแท่นเลื่อนโมเดลที่ 1



รูปที่ 3.2 แสดงการออกแบบชุดสำหรับควบคุมการสั่นบนแท่นเลื่อนโมเดลที่ 1

### 3.2.1.2 โครงสร้างชุดแท่นเลื่อนโมเดลที่ 1



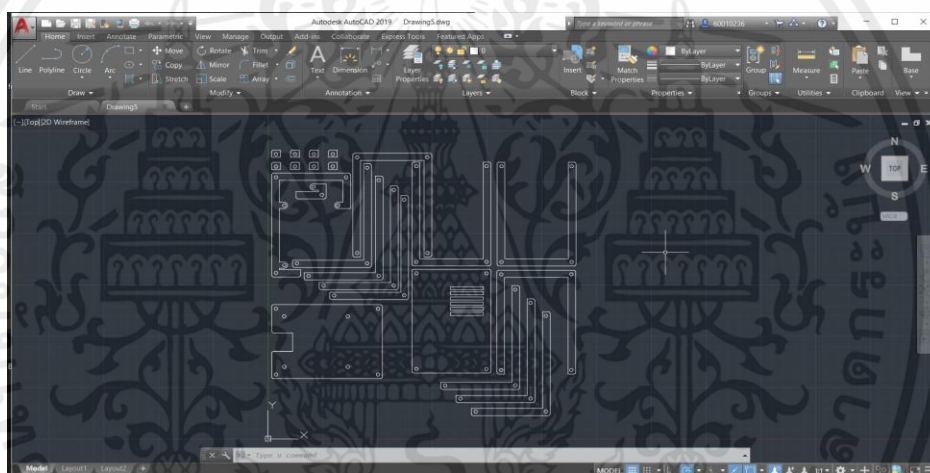
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปที่ 3.3 โครงสร้างระบบสอบเทียบสายเทปวัดระยะโมเดลที่ 1

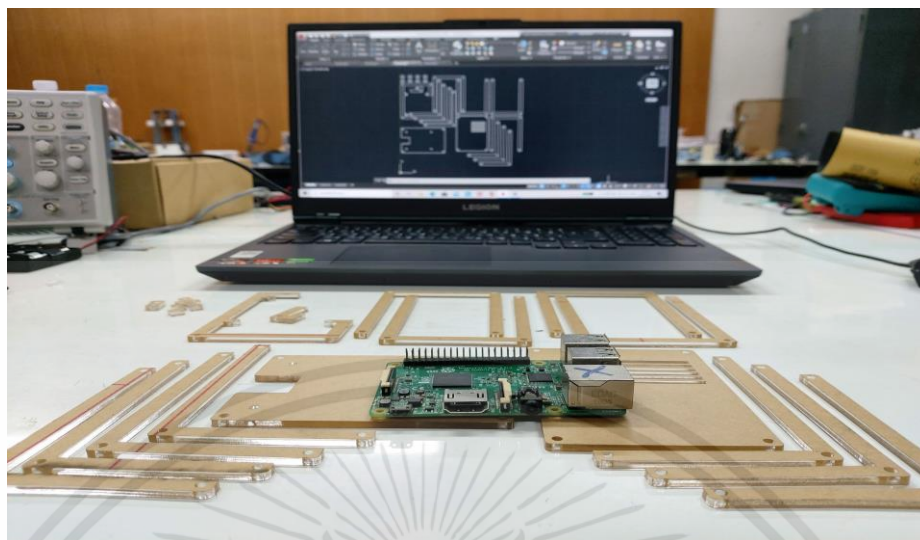
#### 3.2.2 ชุดแทนเลื่อนสำหรับการสอบเทียบสายเทปวัดระยะโมเดลที่ 2

สำหรับชุดแทนเลื่อนในการสอบเทียบสายเทปวัดระยะโมเดลที่ 2 นี้จะมีการเปลี่ยนแปลงจากโมเดลที่ 1 โดยมีการเปลี่ยนแปลงโมเดลในส่วนของชุดควบคุมการสั่งของแทนเลื่อนโดยได้ทำการออกแบบบนซอฟต์แวร์ AutoCAD โดยทำการปรับโมเดลให้มีขนาดที่เหมาะสม และมีน้ำหนักเบา รวมถึงมีการออกแบบเคสสำหรับใส่ Raspberry Pi 3 พร้อมติดเคสเข้ากับชุดควบคุมการสั่ง ซึ่งแสดงดังรูปที่ 3.4

##### 3.2.2.1 การออกแบบชุดแทนเลื่อนโมเดลที่ 2

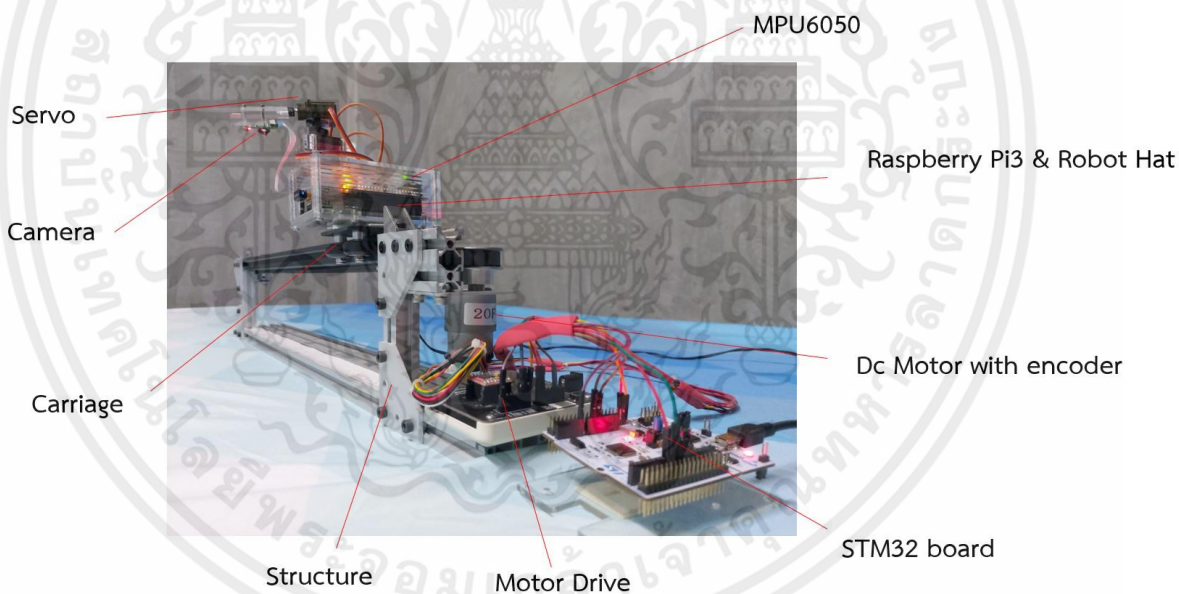


รูปที่ 3.4 แสดงการออกแบบโมเดลของชุดควบคุมการสั่งบน AutoCAD



รูปที่ 3.5 แสดงส่วนประกอบของชุดควบคุมการสั่นหลังการออกแบบ

### 3.2.2.2 โครงสร้างของชุดแทนเลื้อนโมเดลที่ 2

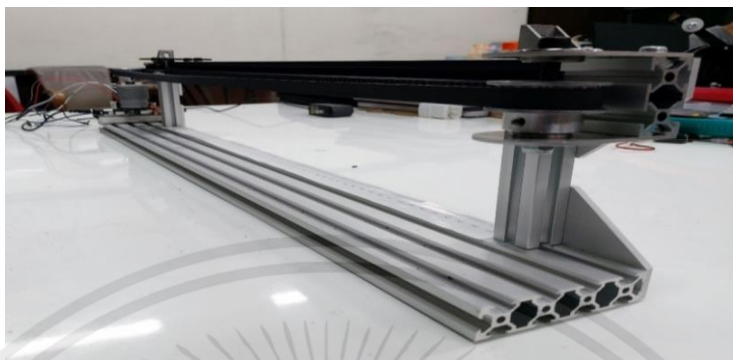


รูปที่ 3.6 โครงสร้างระบบสอบเทียบสายเทปวัดระยะโมเดลที่ 2

โดยโครงสร้างและองค์ประกอบต่างๆของชุดแทนเลื้อนในการสอบเทียบสายเทปวัดระยะ ผู้แต่งปริญญาโทได้ทำการเลือกโครงสร้างของชุดแทนเลื้อนโมเดลที่ 2 ในการจัดทำโครงงาน ซึ่งประกอบไปด้วยอุปกรณ์ทางด้านฮาร์ดแวร์ต่าง ๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. รางอลูมิเนียมโปรไฟล์สำหรับชุดแท่นเลื่อน



รูปที่ 3.7 โครงสร้างรางอลูมิเนียมโปรไฟล์

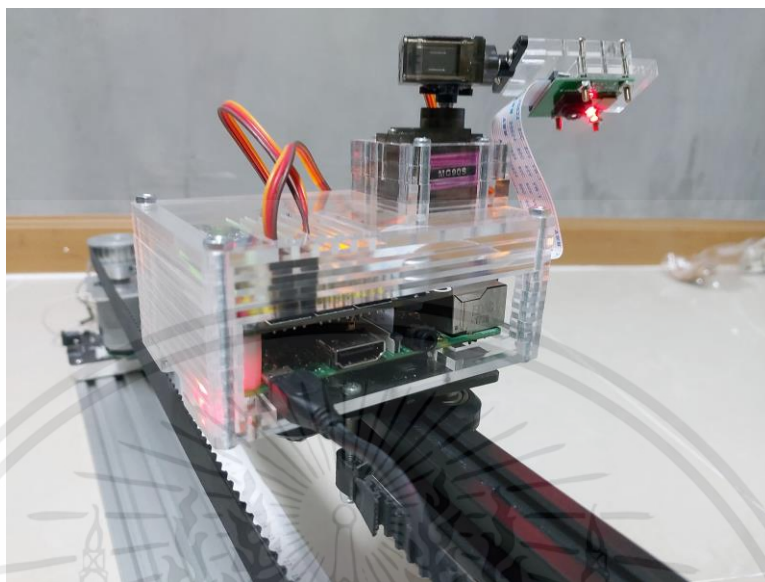
## 2. แท่นเลื่อน

แท่นเลื่อน ซึ่งประกอบไปด้วยฐานอะคริลิกยึดติดกับล้อ (V-Wheel) สำหรับเคลื่อนที่บนรางอลูมิเนียมโปรไฟล์ แสดงดังรูปที่ 3.8



รูปที่ 3.8 แท่นเลื่อน

### 3. ชุดสำหรับควบคุมการสั่นบนแท่นเลื่อน

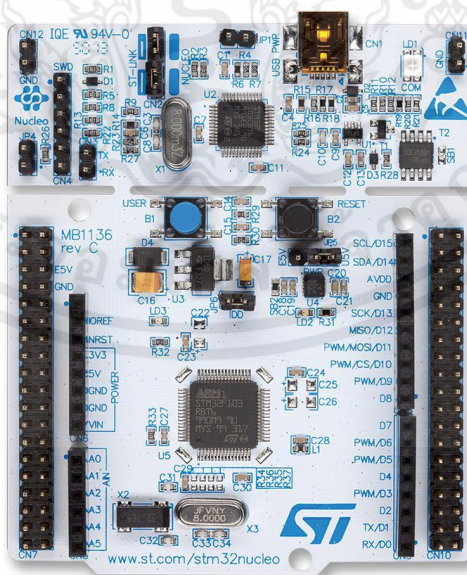


รูปที่ 3.9 ชุดสำหรับควบคุมการสั่นบนแท่นเลื่อน

### 3.3 องค์ประกอบทางด้านฮาร์ดแวร์

ซึ่งองค์ประกอบทางด้านฮาร์ดแวร์ของชุดแท่นเลื่อนในการสอบเทียบสายเทปวัดระยะ มีดังนี้

#### 3.3.1 บอร์ดไมโครคอนโทรลเลอร์ STM32 NUCLEO F401RE



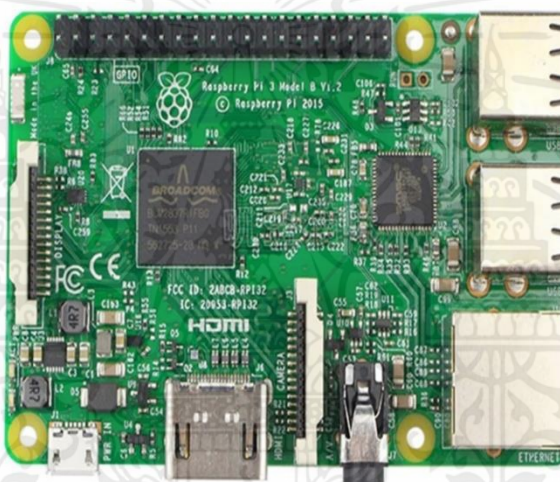
รูปที่ 3.10 NUCLEO F401RE

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มา : <https://th.element14.com/stmicroelectronics/nucleo-f401re/nucleo-board-mcu/dp/2394223>

อุปกรณ์ STM32F401xD/xE ใช้คอร์ ARM®Cortex® -M4 32 บิตประสิทธิภาพสูงที่ทำงานที่ความถี่สูงถึง 84 MHz คอร์Cortex®-M4 มีความแม่นยำเดี่ยวหน่วยจุดลอยตัว (FPU) ซึ่งรองรับคำสั่ง และประเภทข้อมูลการประมวลผลข้อมูลที่มีความแม่นยำเดี่ยวของ ARM ทั้งหมด นอกจากนี้ยังใช้ชุดคำสั่งDS และหน่วยป้องกันหน่วยความจำ (MPU) ซึ่งช่วยเพิ่มความปลอดภัยของการใช้งาน [11]

### 3.3.2 บอร์ด Raspberry Pi 3 Model B



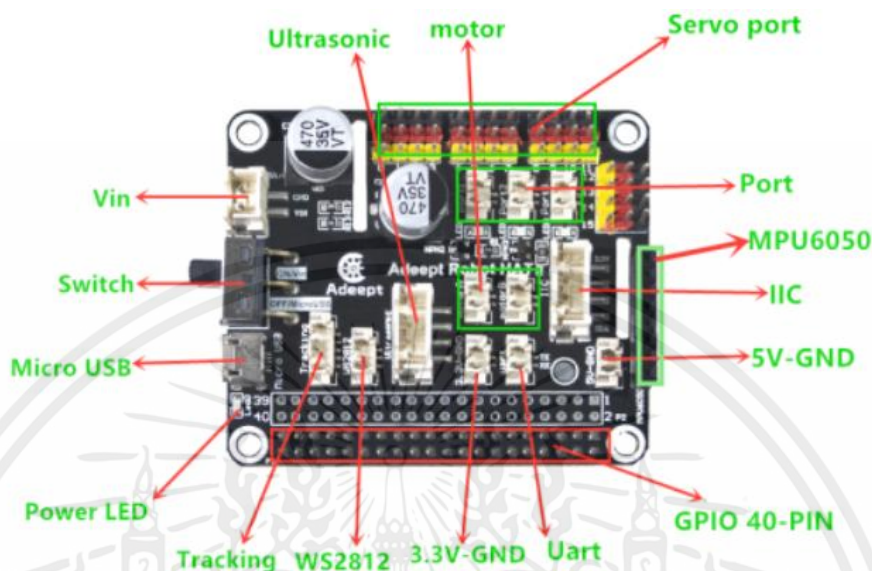
รูปที่ 3.11 Raspberry Pi 3 Model B

ที่มา : <https://www.adeept.com/learn/tutorial-158.html>

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดิโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ได้อีกด้วย [12]

### 3.3.3 Adept RaspClaws Robot Hat



รูปที่ 3.12 Robot Hat

ที่มา : <https://www.adept.com/learn/tutorial-159.html>

บอร์ด Adept Robot HAT มีอินเทอร์เฟซสำหรับเชื่อมต่อเซ็นเซอร์และโมดูลฮาร์ดแวร์อิเล็กทรอนิกส์อื่น ๆ ทำให้สามารถอินเทอร์เฟซได้ง่าย ซึ่งทำงานร่วมกับบอร์ด Raspberry Pi [13]

- Vin : Interface Vin เป็น interface สำหรับแหล่งจ่ายไฟภายนอก
- Switch : ใช้ในการเปิด/ปิด Robot Hat
- Micro USB : เชื่อมต่อ Robot Hat เข้ากับคอมพิวเตอร์
- Power LED : ใช้ระบุสถานะพลังงานของ Robot Hat
- Tracking : ใช้ interface กับ tracking module
- WS2812 : ใช้ interface กับ WS2812 module
- 3.3V-GND : interface กับแหล่งจ่ายไฟฟ้า 3.3 V
- UART : UART interface.
- GPIO 40-PIN : ใช้เชื่อมต่อกับ module ภายนอก
- 5V-GND : interface กับแหล่งจ่ายไฟฟ้า 5 V

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- I2C : ใช้ interface กับ module หน้าจอ OLED
- MPU6050 : ใช้ interface กับ MPU6050
- Servo port : ใช้ interface กับ servo motor
- Motor : แบ่งออกเป็นอินเทอร์เฟสมอเตอร์ 1 และมอเตอร์ 2

### 3.3.4 มอเตอร์ไฟฟ้ากระแสตรงพร้อม Encoder

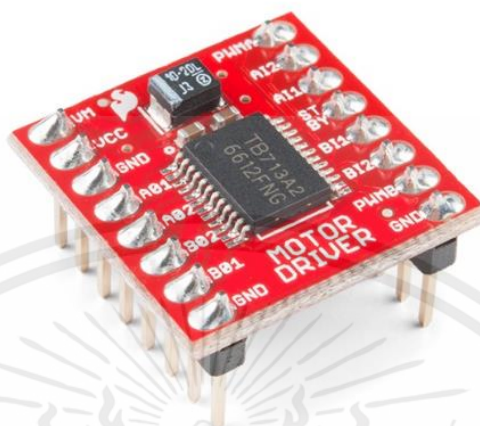


รูปที่ 3.13 มอเตอร์ดีซีเอ็นโค้ดเดอร์ DC Gear Motor + Encoder Hall D-type

ที่มา : <https://s2ins.com/product/motor-gear-encoder-h-dc12v-333rpm-shaft6mm-2/>

บอร์ด quadrature hall effect encoder ออกแบบมาเพื่อให้พอดีกับเพลาด้านหลังของมอเตอร์เกียร์ hall effect sensor ทั้งสองตัววางห่างกัน 90 องศาเพื่อรับและสร้างเอาต์พุต A และ B สองตัวซึ่งอยู่ห่างกันด้วยมุมเฟส 90 องศา และช่วยให้กำหนดทิศทางการหมุนได้ เนื่องจาก encoder ติดตั้งเข้ากับเพลลาของมอเตอร์ และต้องการวัดการหมุนของเพลลาเกียร์ (เพลลาขาออก) ดังนั้นความละเอียดของ encoder จึงขึ้นอยู่กับอัตราทดของเกียร์

### 3.3.5 บอร์ดสำหรับไดร์มอเตอร์ไฟฟ้ากระแสตรง Dual TB6612FNG

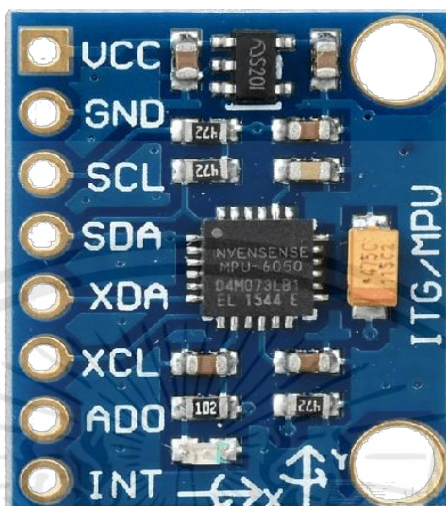


รูปที่ 3.14 Motor Driver - Dual TB6612FNG

ที่มา : <https://www.arduitronics.com/product/1803/motor-driver-dual-tb6612fng-with-headers-%E0%B9%81%E0%B8%97%E0%B9%89%E0%B8%88%E0%B8%B2%E0%B8%81-sparkfun>

TB6612FNG Motor Driver สามารถควบคุมมอเตอร์กระแสตรงได้ถึงสองตัวที่กระแสคงที่ 1.2A (3.2A สูงสุด) สามารถใช้สัญญาณอินพุตสองสัญญาณ (IN1 และ IN2) เพื่อควบคุมมอเตอร์เอาต์พุตมอเตอร์สองตัว (A และ B) สามารถควบคุมแยกกันได้และความเร็วของมอเตอร์แต่ละตัวจะถูกควบคุมผ่านสัญญาณอินพุต PWM ที่มีความถี่สูงถึง 100kHz แรงดันไฟฟาลอจิก (VCC) สามารถอยู่ในช่วง 2.7-5.5VDC ในขณะที่แหล่งจ่ายมอเตอร์ (VM) จำกัด ไว้ที่แรงดันไฟฟ้าสูงสุด 15VDC กระแสเอาต์พุต 1.2A ต่อช่องสัญญาณ ขาทั้งหมดของ TB6612FNG แยกออกเป็นสอง โดยจะถูกจัดเรียงให้พินอินพุตอยู่ที่ด้านหนึ่งและพินเอาต์พุตอยู่อีกด้านหนึ่ง[14]

### 3.3.6 เซนเซอร์วัดความเอียง MPU 6050



รูปที่ 3.15 MPU 6050

ที่มา : <http://www.shop.artronacademy.com/product/64/gy-521-mpu-6050-mpu6050-mpu-6050-module-3-axis-analog-gyro-sensors-3-axis-accelerometer-module>

เซนเซอร์ไจโรสโคปเป็นโมดูลที่ใช้วัดความเร่งชนิดสามแกนรุ่น GY-521 MPU-6050 โดยโมดูลนี้จะประกอบด้วย เซนเซอร์วัดความเร่ง (Accelerometer) และเซนเซอร์ไจโรสโคป (Gyroscope) ทำหน้าที่ตรวจจับการเคลื่อนไหวที่สามารถวัดความเร็วเชิงมุมและการหักเหของแกนที่ความเร็วสูงสุด 30 องศาต่อวินาที และความแตกต่างกันที่ทางออกแบบอนาล็อก ใช้ ADC 16 บิตสาม ตามลำดับสำหรับไจโรสโคป และมาตรการความเร่ง จากนั้นแปลงปริมาณอนาล็อกที่วัดได้เป็นปริมาณดิจิทัลที่ส่งออกได้ ไจโรสโคปสามารถวัดความเร็วเชิงมุมได้และเครื่องวัดความเร่งจะทำการเร่งความเร็ว เพื่อติดตามการเคลื่อนไหวที่รวดเร็ว และ ซ้ำอย่างแม่นยำช่วงการวัดของเซ็นเซอร์สามารถปรับได้: ช่วงของไจโรสโคป คือ  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  และ  $\pm 2000$  ° / วินาที (dps หรือ องศาต่อวินาที) และ ของมาตรวัดความเร่ง คือ  $\pm 2$ ,  $\pm 4$ ,  $\pm 8$  และ  $\pm 16g$  (ความเร่งโน้มถ่วง) รองรับช่วงการทำงานของแรงดันไฟฟ้าของ VCC อยู่ที่ 3-5 VDC [15]

### 3.3.7 เซอร์โวมอเตอร์ MG90S



รูปที่ 3.16 เซอร์โวมอเตอร์ MG90S

ที่มา : <https://www.basic-robot.com/product/70007-56856/micro-metal-gear-servo-mg90s-270-%E0%B8%AD%E0%B8%87%E0%B8%A8%E0%B8%B2>

เซอร์โวเป็นตัวขับเคลื่อนที่กำหนดตำแหน่ง (มุม) ซึ่งเหมาะสำหรับระบบควบคุมที่ต้องการการเปลี่ยนแปลงมุมคงที่และสามารถรักษาสมดุลได้ สามารถใช้สัญญาณ PWM เพื่อควบคุมมุมในการหมุน ซึ่งบนบอร์ด Raspberry Pi driver board Robot HAT มีชิป PCA9685 ที่ใช้ควบคุมเซอร์โวเป็นพิเศษ Raspberry Pi ใช้การ อินเทอร์เฟซแบบ I2C เพื่อสื่อสารกับ PCA9685 ในการควบคุมโดยส่งสัญญาณพัลส์จากไมโครคอนโทรลเลอร์ พัลส์เหล่านี้จะบอกกลไกการทำงานของเซอร์โวว่าจะเคลื่อนที่ไปที่ตำแหน่งใด โดย Servo ที่นำมาใช้จะใช้รุ่น Servo MG90S มีขนาดเล็ก แข็งแรง น้ำหนัก 13.4 g เฟือง Servo ทำจากโลหะ ทอร์ก 2.2 kg-cm ที่แรงดัน 6 V ความเร็วในการหมุน 0.08 sec/60degree (6v) หมุนได้ 180 องศา [16]

### 3.3.8 อะแดปเตอร์ไร้สาย EW-7811Un



รูปที่ 3.17 อะแดปเตอร์ไร้สาย EW-7811Un

ที่มา : <https://www.advice.co.th/product/wireless-network/wireless-usb-adapter/wireless-usb-adapter-edimax-ew-7811un-v2-nano-n150-lifetime-forever->

EW-7811Un เป็นอะแดปเตอร์ไร้สายนาโน USB ที่สามารถรองรับขนาดข้อมูลได้สูงถึง 150Mbps เมื่อเชื่อมต่อกับอุปกรณ์ไร้สาย 802.11n สามารถเสียบเข้ากับพอร์ต USB ของคอมพิวเตอร์และเข้าถึงเครือข่ายไร้สายได้ทันทีสอดคล้องกับมาตรฐาน 802.11n แบบไร้สาย มีอัตราข้อมูลสูงสุด 150Mbps รองรับ Green WLAN รองรับ WPS (Wi-Fi Protected Setup) การออกแบบที่กะทัดรัดและพกพา [17]

## 3.4 ซอฟต์แวร์ที่ใช้พัฒนาระบบ

### 3.4.1 STM32CubeIDE



รูปที่ 3.18 STM32CubeIDE

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

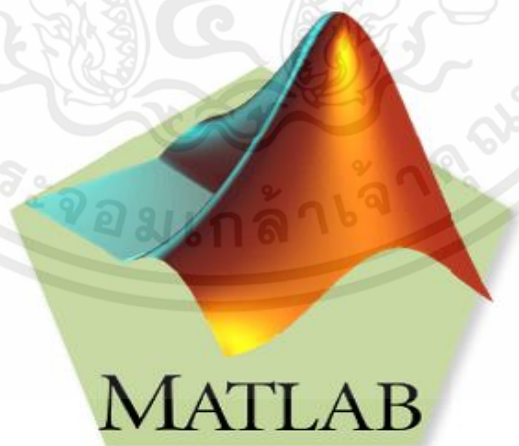
ที่มา : <https://www.disk91.com/2020/technology/programming/code-optimization-with-stm32-cube-ide/>

STM32CubeIDE เป็นแพลตฟอร์มการพัฒนา C หรือ C ++ ขั้นสูงที่มีการกำหนดค่าอุปกรณ์ต่อพ่วงการสร้างโค้ดการคอมไพล์โค้ดและคุณสมบัติการดีบั๊กสำหรับไมโครคอนโทรลเลอร์ และไมโครโปรเซสเซอร์ STM32 มันขึ้นอยู่กับกรอบEclipse® / CDT และ GCC toolchain สำหรับการพัฒนาและ GDB สำหรับการดีบั๊ก อนุญาตให้รวมปลั๊กอินที่มีอยู่หลายร้อยรายการเข้าด้วยกันเพื่อให้คุณลักษณะของEclipse® IDE สมบูรณ์ [18]

STM32CubeIDE รวมการกำหนดค่า STM32 และ ฟังก์ชันการสร้างโปรเจ็คจาก STM32CubeMX เพื่อมอบประสบการณ์การใช้งานเครื่องมือแบบ all-in-one และประหยัดเวลาในการติดตั้งและพัฒนา หลังจากการเลือก STM32 MCU หรือ MPU ที่วางเปล่าหรือไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ที่มีการกำหนดค่าไว้ล่วงหน้าจากการเลือกบอร์ด หรือ การเลือกตัวอย่างโปรเจ็คจะถูกสร้างขึ้นและสร้างโค้ดเริ่มต้น

STM32CubeIDE ยังมีคุณสมบัติการดีบั๊กมาตรฐาน และ ขั้นสูง รวมถึงมุมมองของการลงทะเบียนคอร์ของ CPU ความทรงจำ และ การลงทะเบียนอุปกรณ์ต่อพ่วง

### 3.4.2 MATLAB



รูปที่ 3.19 MATLAB

ที่มา : <http://www.mindphp.com/forums/viewtopic.php?p=94080>

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Matlab เป็นภาษาคอมพิวเตอร์ระดับสูงที่มาพร้อมด้วยสภาพแวดล้อมการทำงานเชิงโต้ตอบ ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์สมัยก่อน เช่น ภาษา C, ภาษา C++ หรือ ภาษา Fortran [19]

Matlab เป็นภาษาคอมพิวเตอร์ระดับสูงที่ใช้สำหรับคำนวณเชิงตัวเลข Numerical Computing แสดงผลกราฟฟิก และเขียนแอปพลิเคชัน ทำให้เราสามารถคำนวณผลลัพธ์ พัฒนาอัลกอริทึม สร้างแบบจำลอง และ แอปพลิเคชันได้ง่ายและรวดเร็วมาก ซึ่งภายในซอฟต์แวร์ Matlab ประกอบด้วย ภาษาคอมพิวเตอร์ Toolbox กลุ่มฟังก์ชันสำเร็จรูปในแต่ละสาขาวิชา และฟังก์ชันพื้นฐานจำนวนมาก ทำให้การวิเคราะห์ทำได้หลากหลายวิธี พร้อมกับคำตอบที่รวดเร็ว

สามารถนำ Matlab ไปประยุกต์ใช้งานได้หลายสาขา ทั้งการประมวลผลสัญญาณ Signal Processing การสื่อสาร Communication การประมวลผลภาพและวิดีโอ Image and Video Processing ระบบควบคุม Control System การวัดและควบคุม Instruments and Control และอื่น ๆ

### 3.4.3 VNC Viewer



รูปที่ 3.20 VNC Viewer

ที่มา : <https://play.google.com/store/apps/details?hl=it&id=com.realvnc.viewer.android>

VNC ย่อมาจาก Virtual Network Computing เป็นโปรแกรมประเภท Remote Control จากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง หรือระหว่างเครื่อง Client ไปยังเครื่อง Server พุดง่าย ๆ VNC คือระบบแบ่งปันทรัพยากรหน้าจอคอมพิวเตอร์ร่วมกัน รวมถึง Input

Device เช่น Mouse และ Keyboard สำหรับทำการ Remote Control คอมพิวเตอร์เครื่องอื่น ๆ และยัง สามารถทำงานร่วมกันได้หลายเครื่องพร้อม ๆ กันรวมทั้งมือถือ เช่น iPhone หรืออุปกรณ์ อิเล็กทรอนิกส์ อื่น ๆ [20]

### 3.4.4 Pycharm

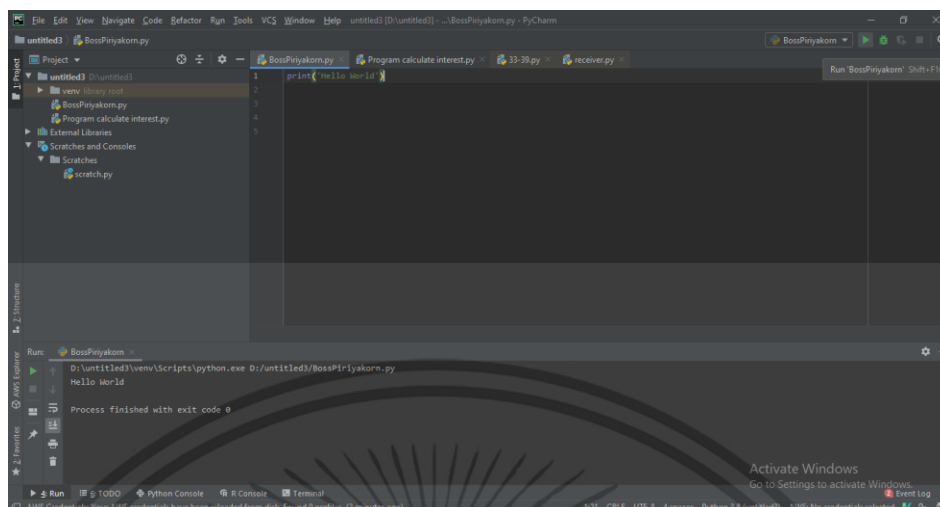


รูปที่ 3.21 PyCharm

ที่มา : <https://www.hostingdynamo.com/hosting-know/windows-10-l-python/>

โปรแกรมที่ใช้หัดเขียนภาษา Python เป็น IDE ที่นิยมมากค่ายหนึ่งสำหรับผู้เริ่มต้นเขียนภาษาไพธอน ซึ่ง Python เป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม หรือก็คือสามารถใช้ได้ทั้งบนระบบ OS Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่วินโดวส์ FreeBSD โดยแบ่งออกเป็น 2 Edition คือ Community Edition และ Professional Edition โดย professional Edition นั้นจะเสียเงินเพื่อเพิ่มเครื่องมือต่าง ๆ เข้าไป ซึ่งจุดเด่นของ Pycharm คือ มาพร้อมกับ Django IDE และเป็นตัว debug ของ JavaScript, Python และ Django [21] [22]

หลังจากติดตั้งไฟล์ .exe แล้วจะขึ้นหน้าต่าง ดังรูปที่ ซึ่งสามารถ สร้างโปรเจค สร้างไฟล์ .py เพื่อใช้งานได้ทันที



รูปที่ 3.22 แสดงหน้าต่างการใช้งาน PyCharm

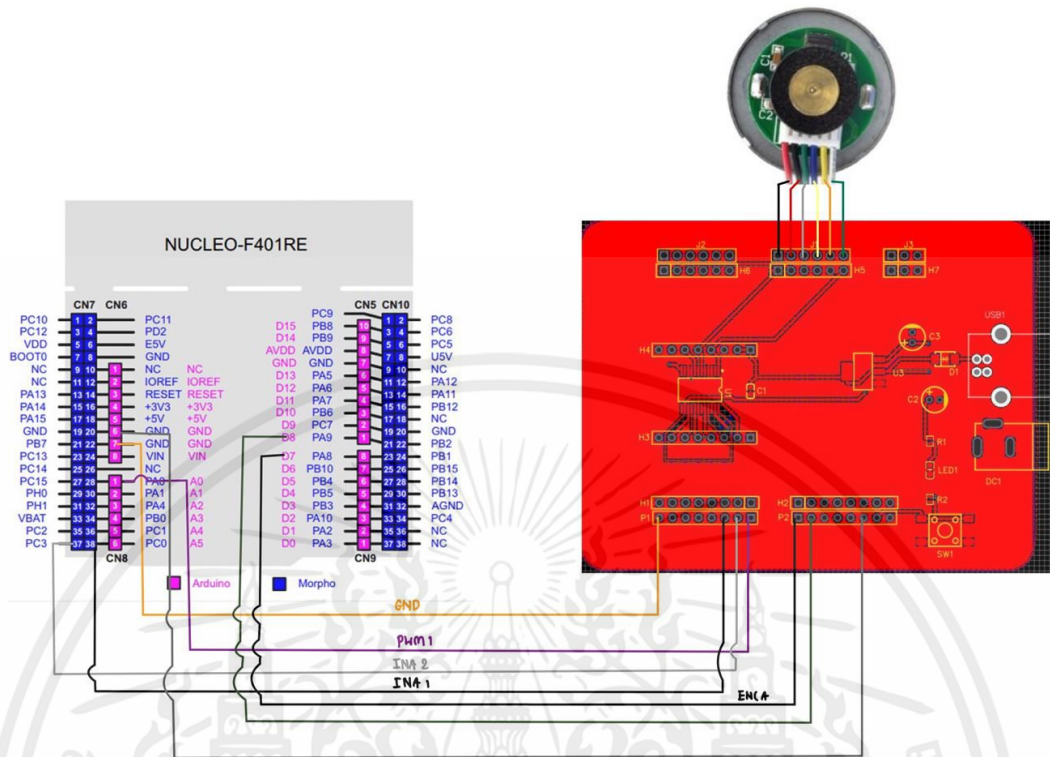
ที่มา : <https://stackpython.medium.com/ide>

### 3.5 การควบคุมการเคลื่อนที่ของชุดแทนเลื่อน

ในส่วนของการควบคุมการเคลื่อนที่ของชุดแทนเลื่อนในระบบจริงนั้น จะใช้ไมโครคอนโทรลเลอร์ STM32 F401RE และใช้ซอฟต์แวร์ STM32 CubeIDE และ ใช้ภาษา C ในการเขียนโปรแกรม โดยขั้นตอนและวิธีการใช้งานเป็นดังต่อไปนี้

#### 3.5.1 การเชื่อมต่อระหว่างบอร์ด STM32 F401RE กับบอร์ด Drive DC Motor

ในการควบคุมการทำงานของมอเตอร์เพื่อบังคับให้ชุดแทนเลื่อนสามารถเคลื่อนที่ ไปยัง ณ ตำแหน่งต่าง ๆ ที่กำหนดไว้ จำเป็นที่จะต้องเชื่อมต่อบอร์ด ไมโครคอนโทรลเลอร์ STM32 F401RE เข้ากับบอร์ด Drive DC Motor ดังรูปที่ 3.10

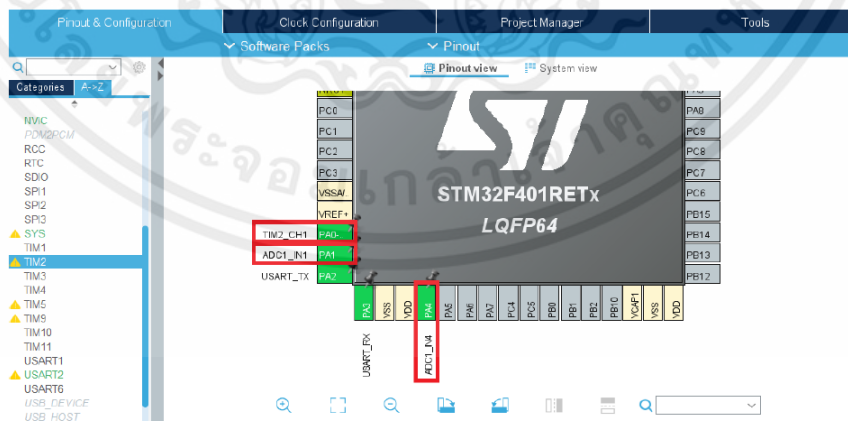


รูปที่ 3.23 แสดงการเชื่อมต่อระหว่าง ไมโครคอนโทรลเลอร์ กับ DC Motor Driver Board

### 3.5.2 การตั้งค่า Pinout และ Clock บน STM32 CubeIDE

ในการตั้งค่า Pinout และ Clock เพื่อเริ่มการเขียนโปรแกรมสามารถทำได้ดังต่อไปนี้

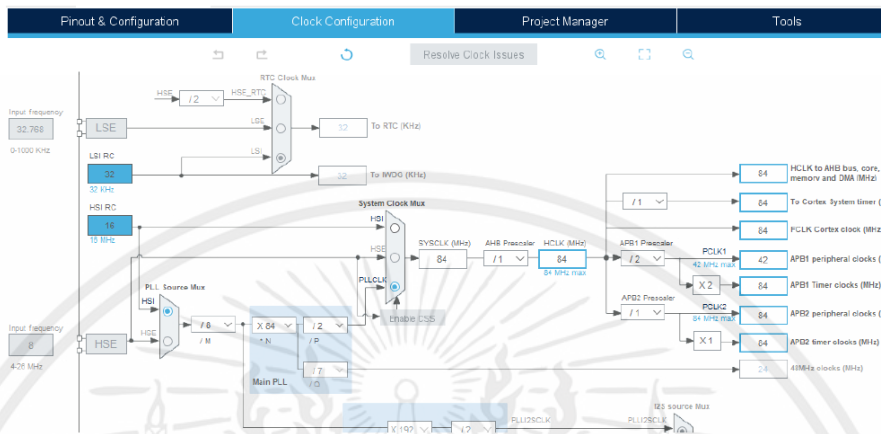
1. ทำการเช็คค่า Pinout TIM2\_CH1, ADC1\_IN1 และ ADC1\_IN2 ดังรูปที่ 3.11



รูปที่ 3.24 แสดงการเช็ค Pinout ก่อนเริ่มเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

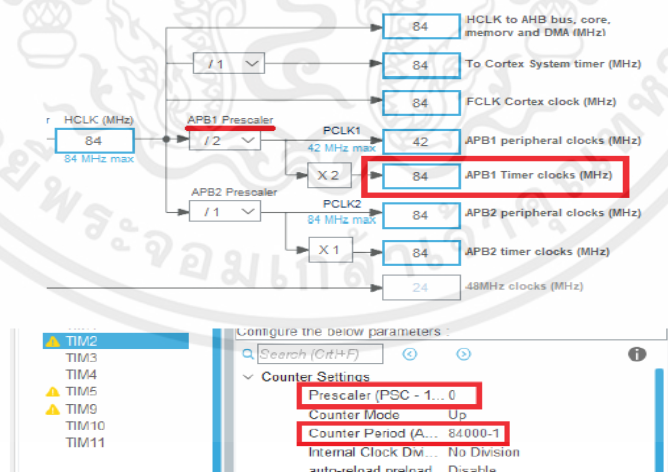
2. ทำการเซ็ท HCLK ในการส่งสัญญาณไปที่ 84 MHz เนื่องจาก TIM 2 ใช้เป็นฐานในการสร้างเวลาของ PWM ซึ่งจะต้องสร้างความถี่ให้ตรงกับสเปคของชุดขับเคลื่อนมอเตอร์ ซึ่งมีค่า 1kHz ดังรูปที่ 3.12



รูปที่ 3.25 แสดงการเซ็ท Clock สำหรับการส่งสัญญาณ

3. ทำการเซ็ท Prescaler และ Counter Period ที่มีความสัมพันธ์กับ Clock และ สเปคของชุดขับเคลื่อนมอเตอร์ แสดงดังสมการที่ 3.1 และ รูปที่ 3.13 โดยจะได้ความถี่ตรงตามที่กำหนด ซึ่งอยู่ที่ 1 kHz

$$\text{Update event} = \frac{\text{High Speed Clock}}{(\text{Prescaler}+1)(\text{Period}+1)} \quad (3.1)$$



รูปที่ 3.26 แสดงการกำหนดค่า Prescaler และ Counter Period ที่สัมพันธ์กับ Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.3 การเขียนโปรแกรมในการเคลื่อนที่สำหรับชุดแทนเลื่อน

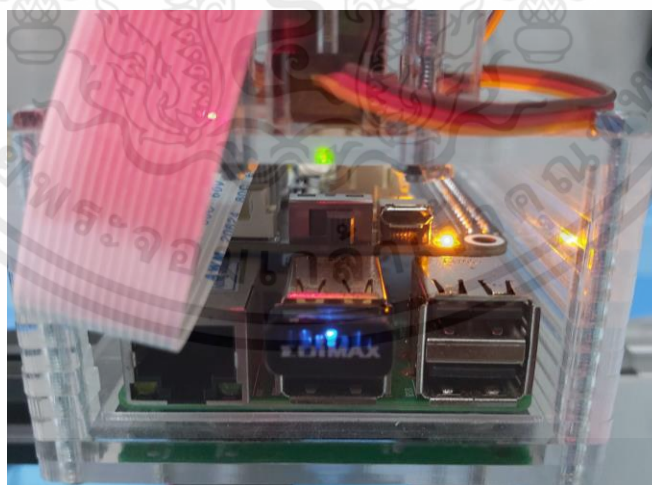
ทำการเขียนโปรแกรมการควบคุม PID สำหรับชุดแทนเลื่อนโดยมีการเขียนโปรแกรมรับค่าป้อนกลับจาก Encoder เป็นสัญญาณพัลส์ตั้งแต่ 0 ถึง 2616 พัลส์ ซึ่งเป็น PV ของระบบ เปรียบเทียบเป็นระยะได้ตั้งแต่ 0 ถึง 49.27 cm. โดยรับค่า Setpoint เป็นหน่วย cm. และนำมาหาค่าความผิดพลาดที่เกิดจากผลต่างของ ค่าเป้าหมายกับผลตอบสนอง ควบคุมผ่าน PID Controller ซึ่งใช้ Library ของ arm\_math.h โดยส่งสัญญาณเอาต์พุตออกมาเป็น Duty cycle ให้แทนเลื่อนสามารถเคลื่อนที่ไปยังตำแหน่งที่กำหนด ซึ่งจะแสดง source code บางส่วนในภาคผนวก ข.

### 3.6 การควบคุมการสั่นของชุดแทนเลื่อน

ในส่วนของการควบคุมการสั่นของชุดแทนเลื่อนในระบบจริง ผู้จัดทำจะใช้ไมโครคอนโทรลเลอร์ Raspberry Pi 3 Model B ซึ่งใช้ระบบปฏิบัติการ Raspbian และใช้ภาษา Python ในการเขียนโปรแกรม โดยขั้นตอนและวิธีการใช้งานเป็นดังต่อไปนี้

#### 3.6.1 การส่งข้อมูลระหว่างอุปกรณ์

การควบคุมการสั่นของชุดแทนเลื่อนด้วยบอร์ด Raspberry Pi 3 จะต้องทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านสาย LAN หรือใช้อะแดปเตอร์ไร้สาย เพื่อทำการเชื่อมต่อผ่าน WIFI แสดงดังรูปที่ 3.14

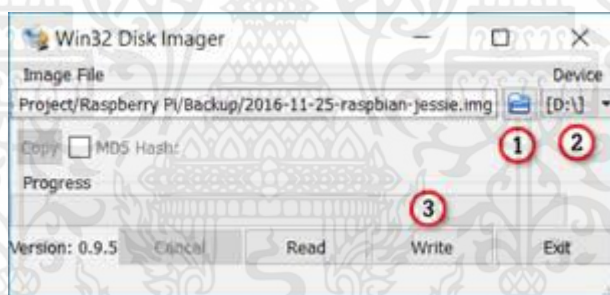


รูปที่ 3.27 แสดงการเชื่อมต่ออะแดปเตอร์ไร้สายเข้ากับ Raspberry Pi 3

### 3.6.2 การติดตั้งระบบปฏิบัติการ Raspbian

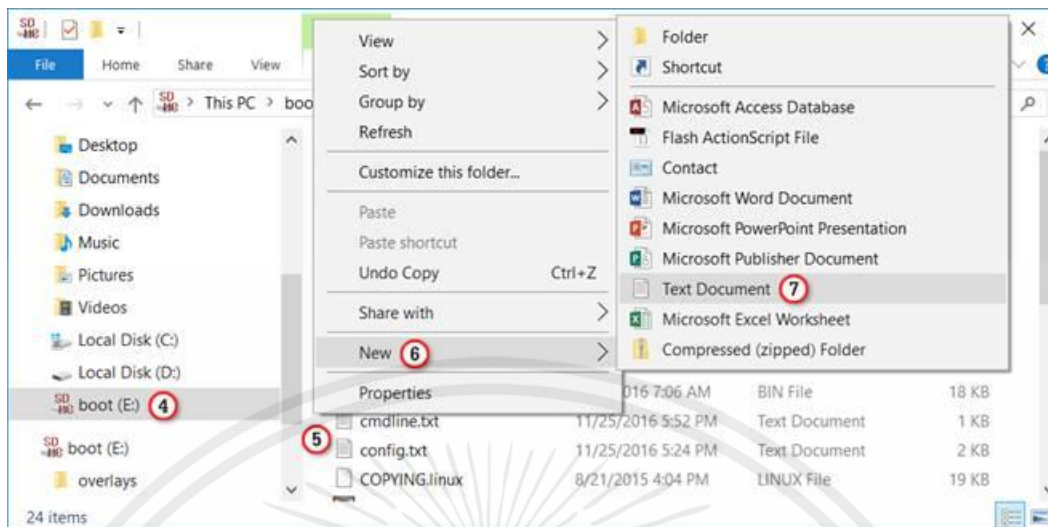
เนื่องจากการติดตั้ง และ ตั้งค่าเริ่มต้นให้กับระบบปฏิบัติการ Raspbian บน Raspberry Pi นั้นทำได้ 2 แบบคือแบบไม่ต่อจอภาพเมาส์และคีย์บอร์ด หรือแบบต่อจอภาพเมาส์และคีย์บอร์ด โดยผู้แต่งปริญญาานิพนธ์ได้เลือกวิธีแรกในการติดตั้งระบบปฏิบัติการ Raspbian คือแบบไม่ต่อจอภาพเมาส์และคีย์บอร์ด ซึ่งขั้นตอนการติดตั้ง Raspbian ลงบน Raspberry PI3 แสดงดังต่อไปนี้ [18]

1. ดาวน์โหลด Raspbian จาก <https://www.raspberrypi.org/downloads/raspbian/> โดยเลือกดาวน์โหลด Raspbian Jessie with PIXEL แบบ .zip
2. เมื่อแตกไฟล์.zip ออกมาจะได้ไฟล์.img จากนั้นให้ทำการใส่การ์ด Micro SD เข้าไปที่เครื่องคอมพิวเตอร์ แล้วเรียกโปรแกรม Win32 Disk Imager ออกมาใช้งาน คลิกปุ่ม Browse(1) แล้วไปยังโฟลเดอร์ที่เก็บไฟล์.imgและเลือกไฟล์.img แล้วเลือกไดรฟ์ (2) ที่บรรจุ Micro SD แล้วคลิก ปุ่ม Write(3) เมื่อปรากฏไดอะล็อกเตือน คลิกปุ่ม Yes แล้วรอนจนกระทั่งปรากฏไดอะล็อกWrite Successful แล้วคลิกปุ่ม OK แสดงว่าระบบปฏิบัติการ Raspbian ได้ถูกบรรจุลงการ์ด Micro SD เรียบร้อยแล้ว ดังรูปที่ 3.15

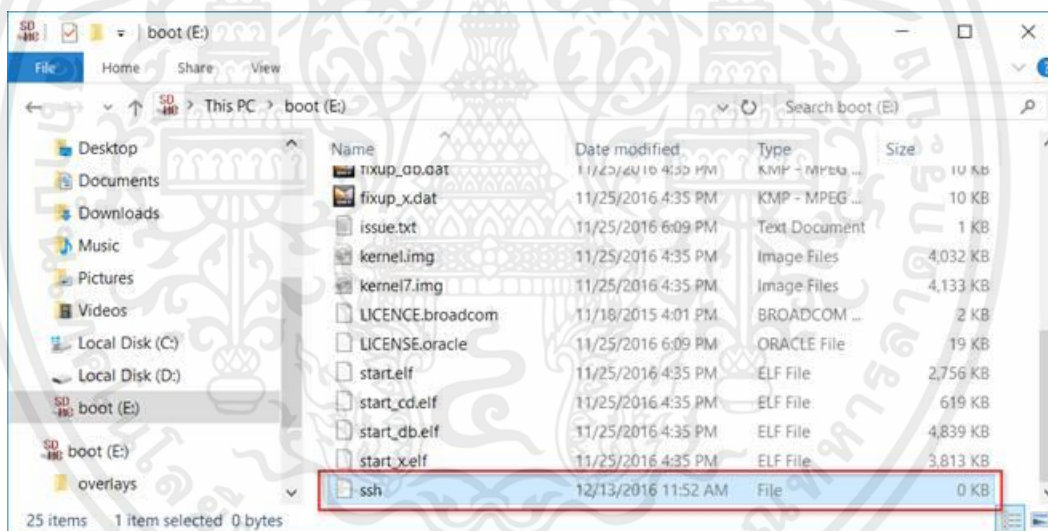


รูปที่ 3.28 แสดงขั้นตอนการใส่ระบบปฏิบัติการ Raspbian ลงใน Micro SD  
ที่มา : <http://www.raspberrypi.org/b/1>

3. เปิดโหมด ssh เพื่อให้สามารถリモทควบคุมการตั้งค่าเริ่มต้นผ่าน PC หรือ Mac ได้ โดยใช้วิธีเดียวกันในการสร้างเทกซ์ไฟล์ชื่อ ssh เก็บไว้ในรูทของการ์ด Micro SD โปรแกรมก็จะเปิดโหมด ssh ให้โดยอัตโนมัติ และสามารถเรียกหน้าต่าง File Explorer แล้วเลือกไดรฟ์ SD Card(4) แล้วคลิกขวาบนพื้นที่ว่างของหน้าต่างไฟล์ด้านขวา (5) แล้วเลือกคำสั่ง New(6) => Text Document (7) แล้วเปลี่ยนชื่อไฟล์เป็น ssh ซึ่งไม่จำเป็นต้องมีข้อมูลในไฟล์ ซึ่งแสดงดังรูปที่ 3.16 และ 3.17



รูปที่ 3.29 แสดงการสร้างเทกซ์ไฟล์ชื่อ ssh เก็บไว้ในรูทของการ์ด Micro SD  
ที่มา : <http://www.raspberrypi.org/b/1>



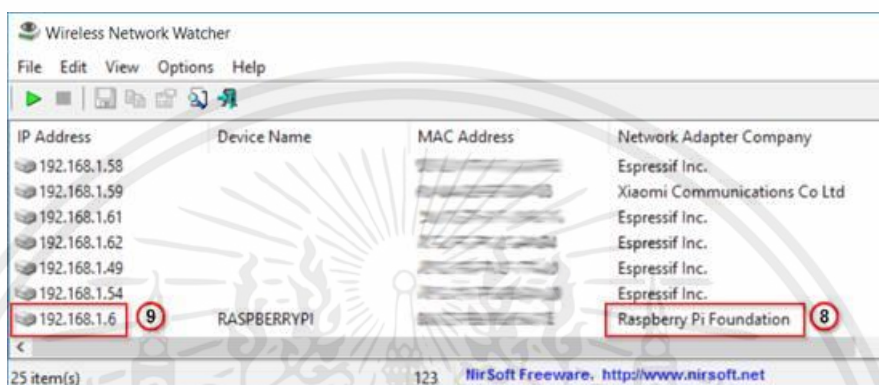
รูปที่ 3.30 แสดงเทกซ์ไฟล์ชื่อ ssh  
ที่มา : <http://www.raspberrypi.org/b/1>

4. ถอดการ์ด Micro SD ออกจากเครื่องคอมพิวเตอร์ โดยก่อนจะใส่ Micro SD เข้าไปยัง Raspberry PI3 จะต้องแน่ใจว่ายังไม่จ่ายไฟเข้าตัวเครื่อง Raspberry PI3 เพื่อป้องกันการเสียหายที่มักเกิดกับ Micro SD

5. เมื่อใส่ Micro SD เป็นที่เรียบร้อยแล้ว เสียบสาย LAN เข้าที่พอร์ต LAN ของ Raspberry PI3 ปลายสายอีกด้านหนึ่งเสียบเข้าที่พอร์ต LAN ที่ยังว่างอยู่ของเราเตอร์ เริ่มทำการ Boot เครื่อง โดย

จ่ายไฟเข้า Raspberry PI หรือเสียบสาย Micro USB เข้าตัวเครื่อง Raspberry PI โปรแกรมจะเริ่มบูทเข้าสู่โหมด Desktop

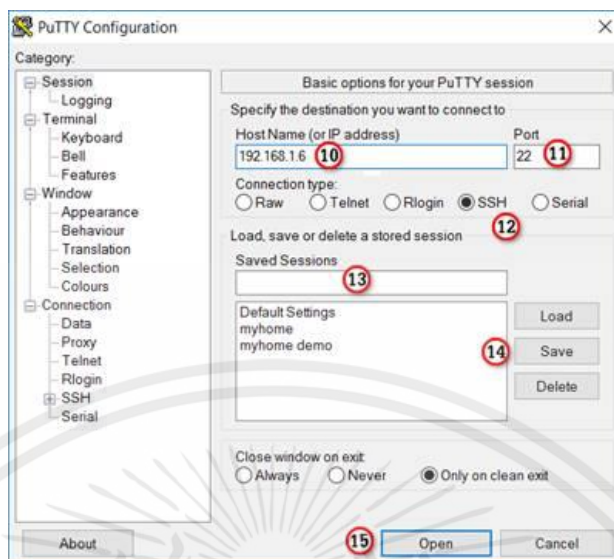
6. ทำการเรียกโปรแกรม Wireless Network Watcher ออกมาใช้งาน จากนั้นให้ทำการหา Raspberry PI Foundation (8) จะพบ IP Addresss (9) ของเครื่อง ซึ่งแสดงได้ดังรูปที่ 3.18



รูปที่ 3.31 แสดงการหา IP Addresss

ที่มา : <http://www.raspberrypi.org/b/1>

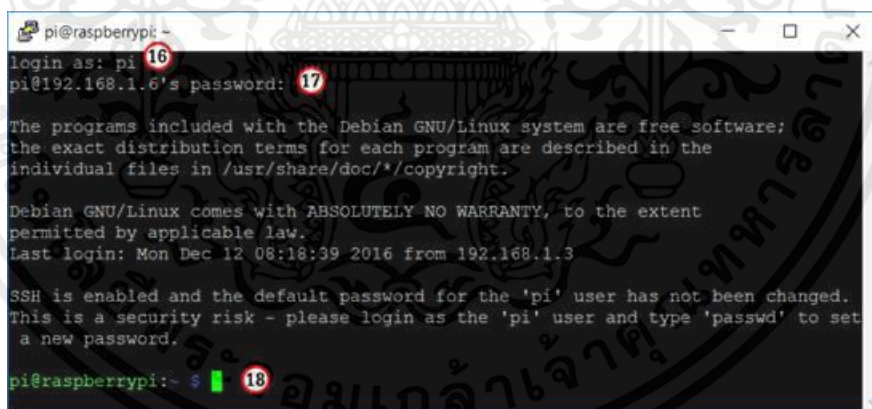
7. เปิดโปรแกรม Putty และ ทำการป้อน IP Address ใน Host Name (10) ให้แน่ใจว่าปรากฏหมายเลขพอร์ต 22 (11) เลือก ssh (12) เป็นที่เรียบร้อยแล้ว จากนั้นให้ทำการบันทึกการตั้งค่า โดยตั้งชื่อในช่อง Saved Sessions (13) แล้วคลิกปุ่ม Save (14) เพื่อบันทึกชื่อ แล้วคลิกปุ่ม Open (15) จากนั้นตัวโปรแกรม Putty จะทำการเชื่อมต่อกับ Raspberry PI3 ตามที่อยู่ IP ที่ทำการป้อนไว้ ซึ่งแสดงได้ดังรูปที่ 3.19



รูปที่ 3.32 แสดงการใช้งานโปรแกรม Putty เพื่อเชื่อมต่อไปยัง Raspberry PI

ที่มา : <http://www.raspberrypi.org/b/1>

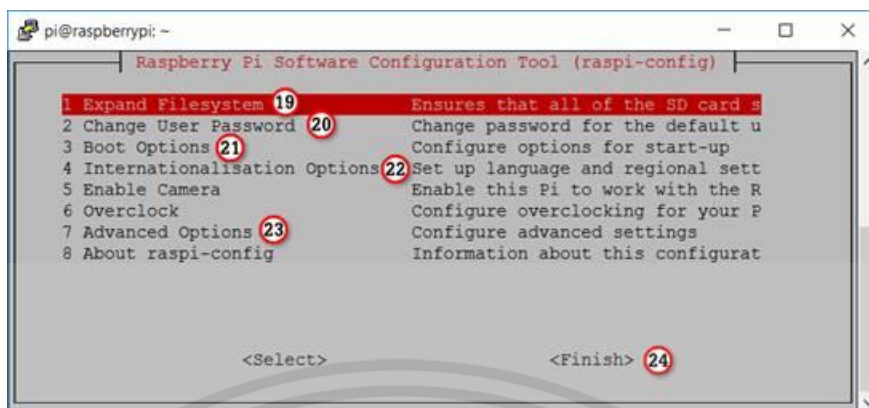
8. ป้อนชื่อยูสเซอร์ pi (16) และพาสเวิร์ด raspberry (17) เมื่อป้อนพาสเวิร์ดเรียบร้อยแล้ว จะปรากฏพร้อมพ์ (Prompt) รอรับคำสั่ง pi@raspberrypi:~ \$ (18) ดังรูปที่ 3.20



รูปที่ 3.33 แสดงการ login

ที่มา : <http://www.raspberrypi.org/b/1>

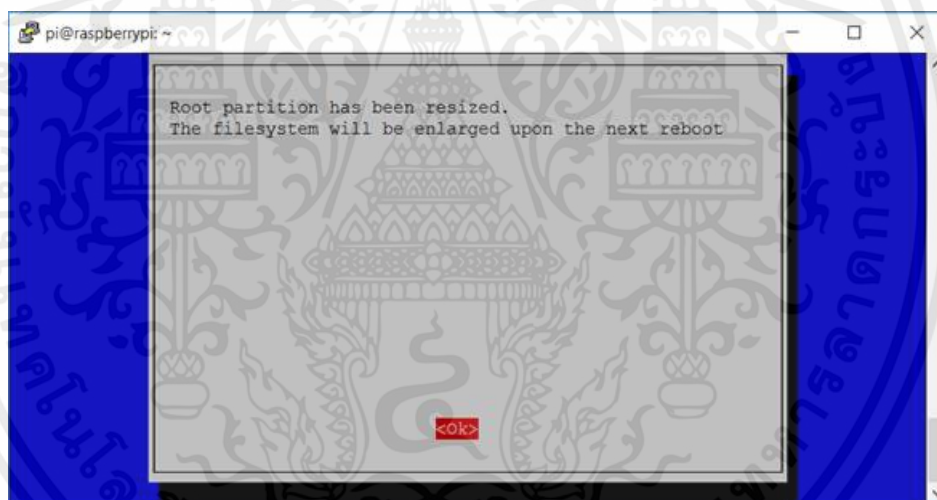
9. กำหนดค่าเริ่มต้น โดยพิมพ์คำสั่ง pi@raspberrypi:~ \$ sudo raspi-config ซึ่งดังรูปที่ 3.21



รูปที่ 3.34 แสดงหน้าต่างกำหนดค่าเริ่มต้น

ที่มา : <http://www.raspberrypi.org/b/1>

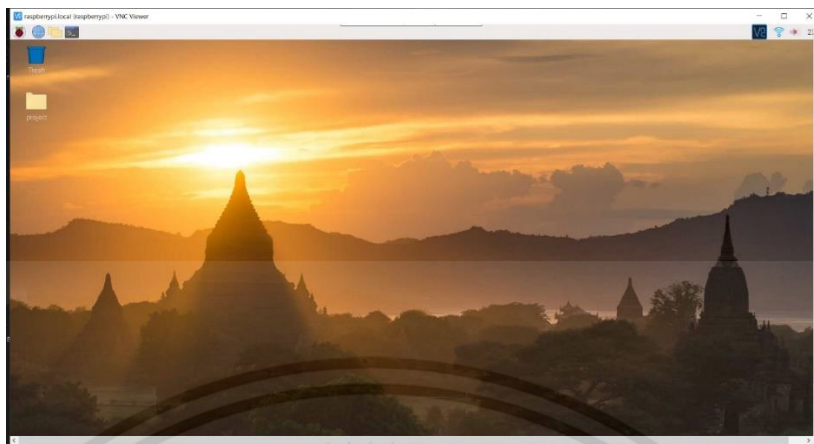
10. เลือก Expand Filesystem (19) จะปรากฏจอภาพดังรูปข้างล่างนี้โปรแกรมจะรายงานว่า รุกพาร์ทติชั่นได้ถูกปรับขนาด ระบบไฟล์จะถูกขยายเมื่อบูท(Boot)เครื่องในครั้งต่อไป แสดงดังรูปที่ 3.22



รูปที่ 3.35 แสดงขึ้นตอนหลังจากทำการเลือก เลือก Expand Filesystem

ที่มา : <http://www.raspberrypi.org/b/1>

11. ทำการ Enable VNC และ เลือก Finish เพื่อจบขั้นตอนในการติดตั้ง จากนั้นให้ทำการ เชื่อมต่อกับ VNC จะสามารถเข้าใช้งานระบบปฏิบัติการ Raspbian บน Raspberry Pi ได้ แสดงดังรูปที่ 3.23



รูปที่ 3.36 แสดงหน้าต่างเริ่มต้นใช้งาน

### 3.6.3 การเขียนโปรแกรมเพื่อควบคุมการสั่นของชุดแท่นเลื่อน

ทำการเขียนโปรแกรมการควบคุม PID สำหรับชุดควบคุมการสั่นของแท่นเลื่อนโดยมีการรับค่า ป้อนกลับของมุมพิทช์และยอร์จาก MPU6050 ซึ่งเป็น PV ของระบบ โดยรับค่า Setpoint เป็นหน่วยองศา และนำมาหา error โดยการเปรียบเทียบกับมุมพิทช์และยอร์ที่ได้กำหนดไว้เพื่อควบคุมผ่าน PID Controller ซึ่งใช้ Library PID.py และแปลงสัญญาณเอาต์พุตที่ได้จาก PID Controller เป็นสัญญาณ PWM เพื่อควบคุมมุมในการเคลื่อนที่ของ Servo Motor ซึ่งจะแสดง source code บางส่วนไว้ในภาคผนวก ข.

## 3.7 การสอบเทียบสายวัดระยะ

ขั้นตอนการสอบเทียบสายวัดมาตรฐานด้วยระบบสอบเทียบสายวัดมีดังนี้

### 3.7.1 วิธีการเตรียมและติดตั้งเครื่องมือและสายวัด

1. การเตรียม ทำความสะอาดสายวัดและทำการติดสายวัดกับรางอลูมิเนียมตรง
2. ทำการปรับตั้งสายวัดให้อยู่แนวเดียวกับกล่องตลอดช่วงการวัดก่อน สังเกตเส้นสีแดงซึ่งจะต้องอยู่ในแนวเดียวกับเส้นที่อยู่บนสายวัด ดังรูปที่ 3.24 เพื่อป้องกันความคลาดเคลื่อนก่อนทำการสอบเทียบ



รูปที่ 3.37 การปรับตั้งสายวัด

3. ทำการปรับตั้งชุดแทนเลื่อนให้อยู่ที่จุดเริ่มต้นซึ่งแสดงดังรูปที่ 3.25 และตรวจสอบแนวของสายเทปวัดระยะเพื่อป้องกันความคลาดเคลื่อนอีกครั้ง



รูปที่ 3.38 แสดงการปรับตั้งชุดแทนเลื่อนให้อยู่ที่จุดเริ่มต้น

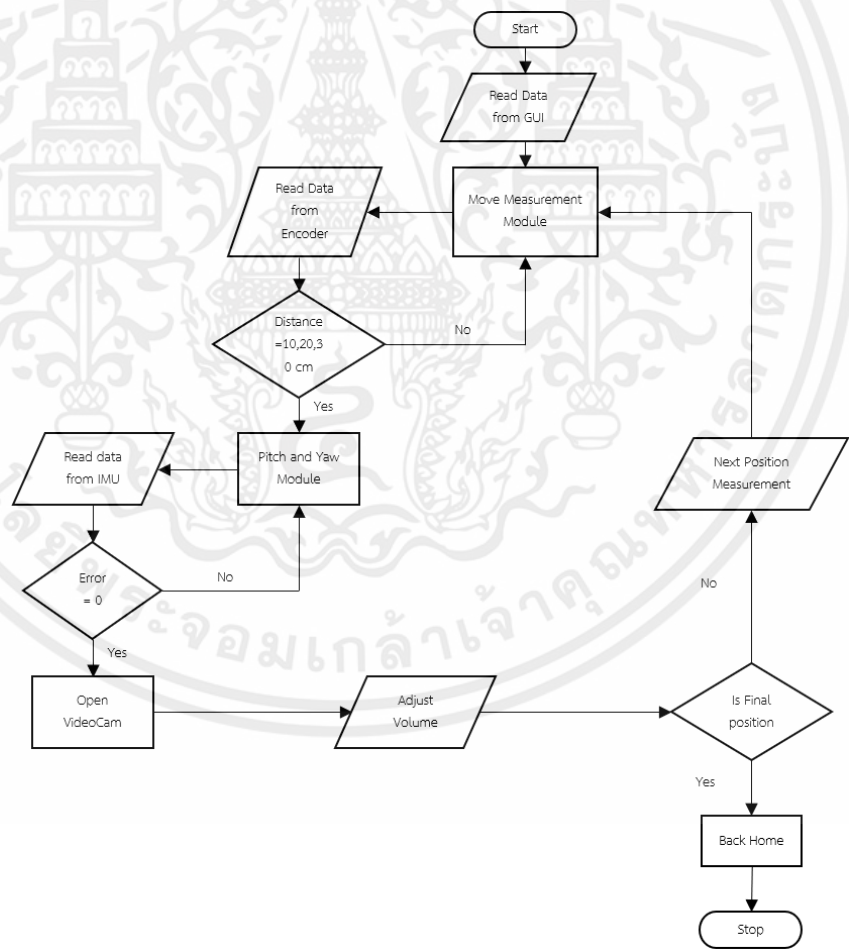
### 3.7.2 ขั้นตอนการสอบเทียบสายเทปวัดระยะที่เป็นมาตรฐานอ้างอิง

1. ตรวจสอบการปรับตั้งสายวัดซึ่งวางอยู่บนรางเหล็กตรง ก่อนทำการสอบเทียบทุกครั้ง เพื่อให้แน่ใจว่า ตำแหน่งของสายเทปวัดระยะขนานกับแนวขอบและอยู่แนวเดียวกับกล่องตลอดช่วงการวัด ซึ่ง สังเกตได้จากเส้นสีแดง
2. จากนั้นวัดไปตามลำดับไปถึงจุดที่กำหนด โดยจะมีการกำหนดจุดในการวัดแบ่งเป็น 10, 20 และ 30 cm. ตามลำดับ

- 3. เมื่อถึงจุดSetpoint ที่ระยะ 10, 20 และ 30 cm. ตามลำดับ ให้สังเกตว่าเส้นสีแดงทับกับเส้นสเกลของสายเทปวัดระยะที่เป็นมาตรฐานอ้างอิงหรือไม่
- 4. หากสังเกตว่าเส้นสีแดงทับกับ เส้นสเกลของสายเทปวัดระยะที่เป็นมาตรฐานอ้างอิง ไม่สนิทให้ทำการปรับ VR เพื่อชดเชยให้แทนเลื่อนเคลื่อนไปตรงกับระยะ ณ จุดที่ทำการวัด
- 5. ซึ่งลำดับขั้นตอนต่าง ๆ จะถูกทำซ้ำจนกว่าระยะของแทนเลื่อนตรงกับระยะ ณ จุดที่ทำการวัดทุกระยะตามลำดับ

### 3.7.3 การทำงานของระบบควบคุมและประมวลผล

การทำงานของระบบควบคุมและประมวลผล จะมีการทำงานแบบกึ่งอัตโนมัติ ซึ่งขั้นตอนของการทำงานมีดังต่อไปนี้



รูปที่ 3.39 แสดงแผนภูมิการทำงานของชุดโปรแกรมควบคุมและประมวลผล

1. เริ่มต้นการทำงานของมอเตอร์จะได้รับคำสั่งให้เคลื่อนแท่นเลื่อนจากจุดเริ่มต้นไปยังจุดที่กำหนด และเมื่อเอ็นโค้ดเดอร์ตรวจสอบแล้วว่าถึงระยะที่กำหนดจะสั่งให้มอเตอร์หยุดหมุนทันที
2. หลังจากที่แท่นเลื่อนเคลื่อนไปยังจุดที่กำหนดแล้ว ชุดสำหรับควบคุมการสั่นบนแท่นเลื่อนจะทำงานโดยการสั่งให้ควบคุมค่าความผิดพลาดที่เกิดจากการสั่นให้เหลือน้อยที่สุด
3. หลังจากที่ชุดสำหรับควบคุมการสั่นบนแท่นเลื่อนทำงานเสร็จเรียบร้อยแล้ว กล้องดิจิตอลจะได้รับคำสั่งให้ทำการเปิดวิดีโอเพื่อดูระยะการเคลื่อนที่ของชุดแท่นเลื่อน
4. หากเส้นสีแดงที่ทำการมาร์คไว้ ไม่ตรงกับสเกลของค่าที่เซตไว้ ให้ทำการปรับ VR เพื่อปรับค่าชดเชยให้ตรงกับระยะที่กำหนดไว้
5. เมื่อทำการวัดถึงตำแหน่งสุดท้ายตามที่ได้ตั้งโปรแกรมการวัดไว้ระบบควบคุมจะสั่งให้ชุดแท่นเลื่อนเคลื่อนกลับมายังจุดเริ่มต้นและทำการรับค่า Setpoint อีกครั้ง

## บทที่ 4

### ผลการทดลอง

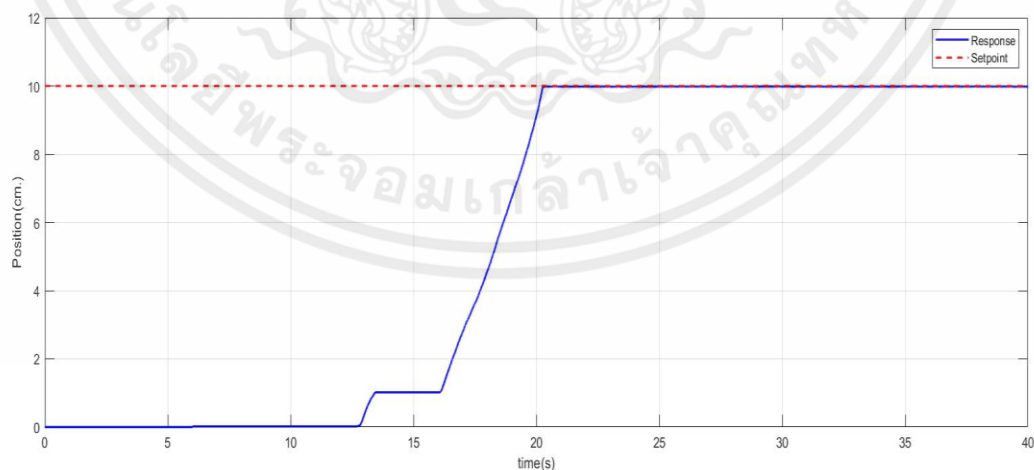
#### 4.1 ผลการทดลองเกี่ยวกับการเคลื่อนที่ของชุดแท่นเลื่อน

การทดสอบการเคลื่อนที่ของชุดแท่นเลื่อนในการสอบเทียบสายเทปวัดระยะนั้น ได้ทำการเลือกค่าเป้าหมายของการทดลองการเคลื่อนที่อยู่ที่ 10, 20 และ 30 cm. ตามลำดับ โดยใช้พารามิเตอร์ PID ที่ได้จากการออกแบบไว้ ดังตารางที่ 4.1

ตารางที่ 4.1 ตารางแสดงค่าพารามิเตอร์ของ PID Control

Parameters	Gains
Kp	0.001
Ki	0.001
Kd	1.5

##### 4.1.1 การเคลื่อนที่ของชุดแท่นเลื่อนไปที่ 10 cm.



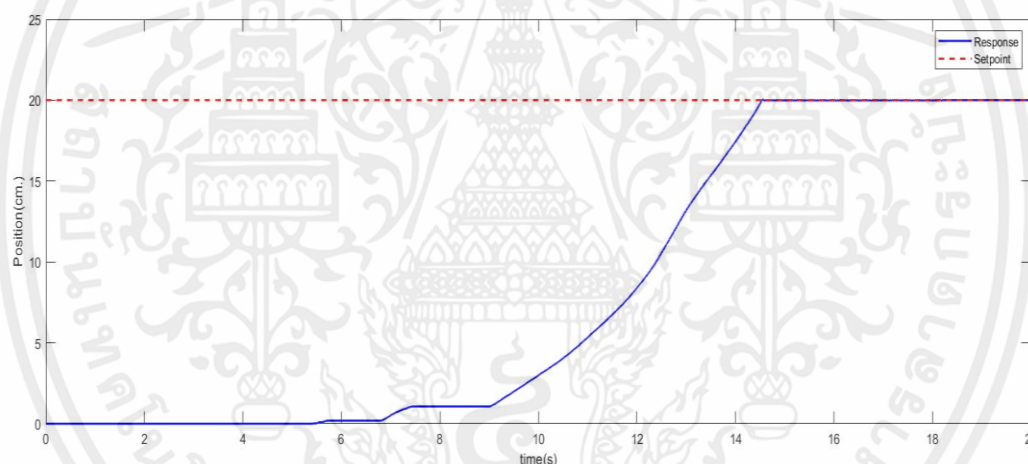
รูปที่ 4.1 แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 10 cm.

จากผลการทดลองตามรูปที่ 4.1 จะได้ค่าตัวแปรต่าง ๆ แสดงดังตารางที่ 4.2

**ตารางที่ 4.2** ตารางแสดงค่าพารามิเตอร์ของผลการตอบสนองจากการปรับ Setpoint ที่ 10 cm.

Parameters	values
ช่วงเวลาขึ้น (Rise time)	7.0890 s
เวลาเข้าที่ (Settling time )	20.2147 s
ค่าพุ่งเกิน (Over shoot)	0 %

#### 4.1.2 การเคลื่อนที่ของชุดแท่นเลื่อนไปที่ 20 cm.



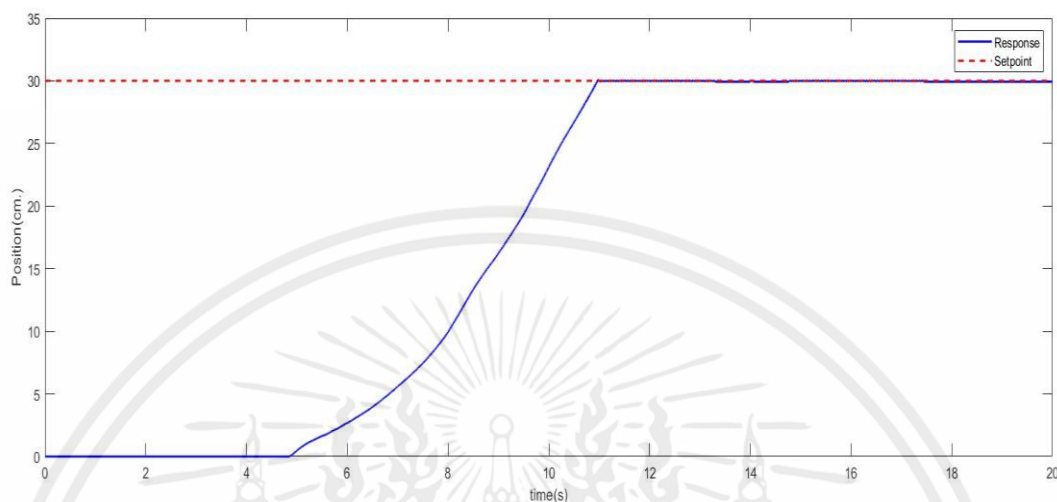
**รูปที่ 4.2** แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 20 cm.

จากผลการทดลองตามรูปที่ 4.2 จะได้ค่าตัวแปรต่าง ๆ แสดงดังตารางที่ 4.3

**ตารางที่ 4.3** ตารางแสดงค่าพารามิเตอร์ของผลการตอบสนองจากการปรับ Setpoint ที่ 20 cm.

Parameters	values
ช่วงเวลาขึ้น (Rise time)	4.6275 s
เวลาเข้าที่ (Settling time )	14.4663 s
ค่าพุ่งเกิน (Over shoot)	0 %

### 4.1.3 การเคลื่อนที่ของชุดแทนเลื่อนไปที่ 30 cm.



รูปที่ 4.3 แสดงผลการตอบสนองจากการปรับ Setpoint ที่ 30 cm.

จากผลการทดลองตามรูปที่ 4.3 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ แสดงดังตารางที่ 4.4

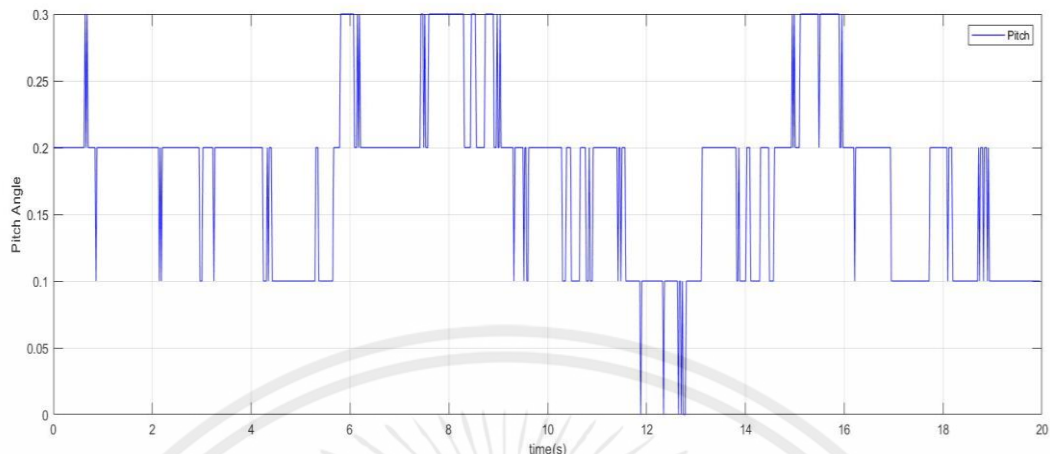
ตารางที่ 4.4 ตารางแสดงค่าพารามิเตอร์ของผลการตอบสนองจากการปรับ Setpoint ที่ 30 cm.

Parameters	values
ช่วงเวลาขึ้น (Rise time)	4.4119 s
เวลาเข้าที่ (Settling time)	10.8875 s
ค่าพุ่งเกิน (Over shoot)	0 %

## 4.2 ผลการทดลองเกี่ยวกับการควบคุมสมดุลของชุดแทนเลื่อน

### 4.2.1 ผลการวัดค่าความผิดพลาดในแนวแกน พิทช์ และ ยอร์ ขณะอยู่ที่ตำแหน่ง 10 cm.

ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนพิทช์ (Y)



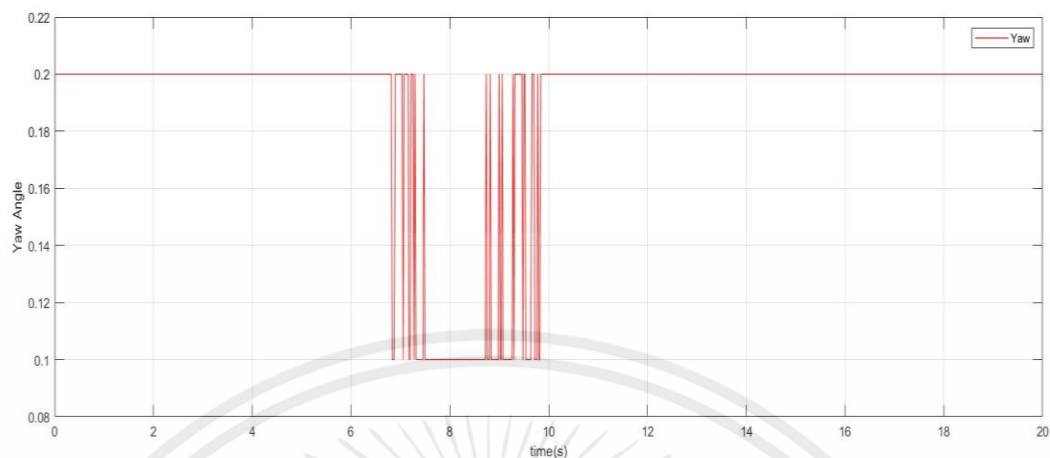
รูปที่ 4.4 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทช์ที่ระยะ 10 cm.

จากผลการทดลองตามรูปที่ 4.4 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ แสดงดังตารางที่ 4.4

ตารางที่ 4.5 ตารางแสดงค่าพารามิเตอร์ของ PID Control

Parameters	values
Max Pitch error	0.3 องศา
Min Pitch error	0.0 องศา
ค่าเฉลี่ย Pitch error	0.2004 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.0643
ความแปรปรวน	0.0041

ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนยอร์ (Z)



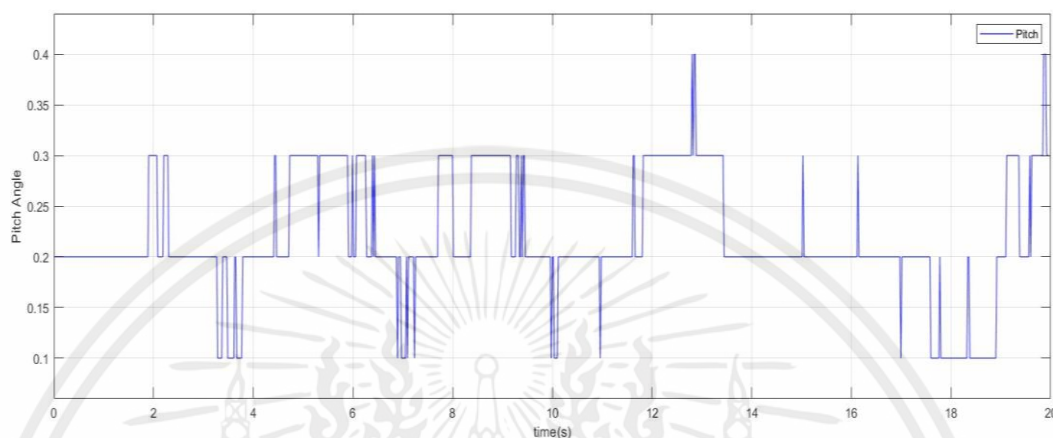
รูปที่ 4.5 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์ที่ระยะ 10 cm.

จากผลการทดลองตามรูปที่ 4.5 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ดังตารางที่ 4.5

ตารางที่ 4.6 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยอร์ที่ระยะ 10 cm.

Parameters	values
Max Yaw error	0.2 องศา
Min Yaw error	0.1 องศา
ค่าเฉลี่ย Yaw error	0.1884 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.0317
ความแปรปรวน	0.0010

4.2.2 ผลการวัดค่าความผิดพลาดในแนวแกน พิทช์ และ ยอร์্ব ขณะอยู่ที่ตำแหน่ง 20 cm.  
ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนพิทช์ (Y)



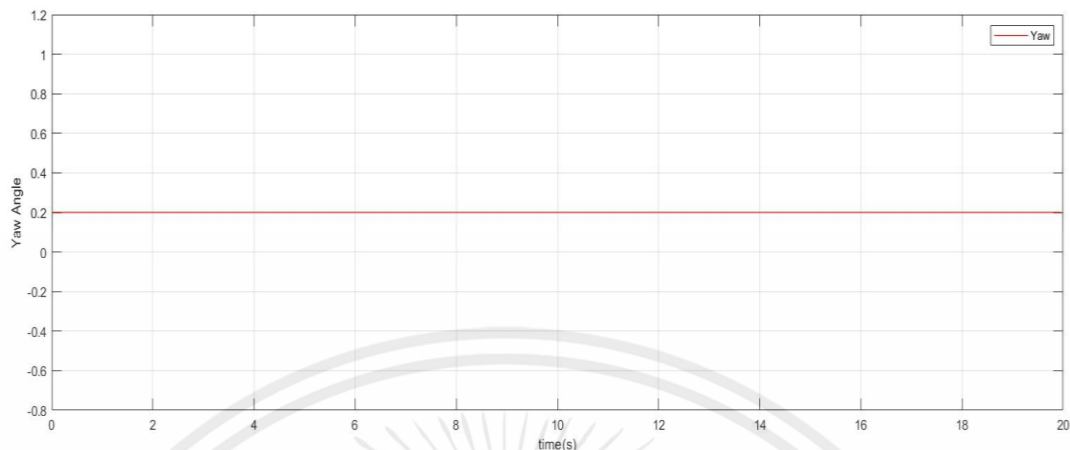
รูปที่ 4.6 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทช์ที่ระยะ 20 cm.

จากผลการทดลองตามรูปที่ 4.6 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ดังตารางที่ 4.6

ตารางที่ 4.7 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนพิทช์ที่ระยะ 20 cm.

Parameters	values
Max Pitch error	0.4 องศา
Min Pitch error	0.1 องศา
ค่าเฉลี่ย Pitch error	0.2176 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.0591
ความแปรปรวน	0.0035

ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนยอร์ว (Z)



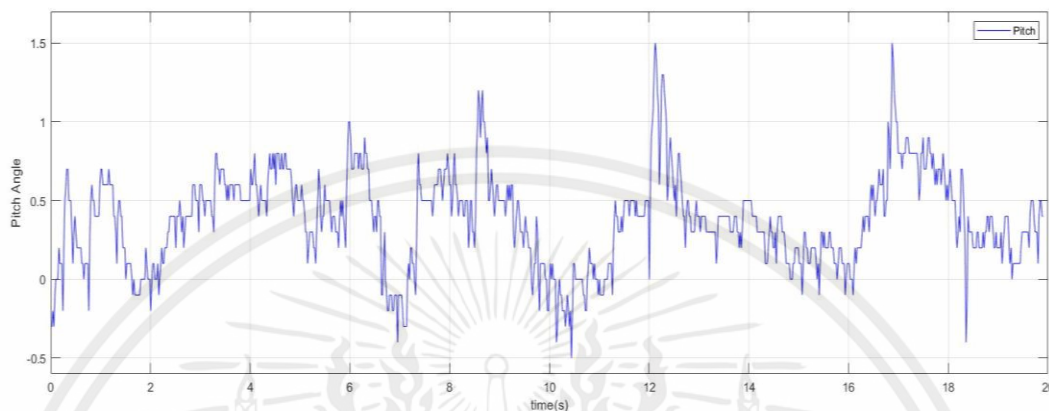
รูปที่ 4.7 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์ที่ระยะ 20 cm.

จากผลการทดลองตามรูปที่ 4.7 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ดังตารางที่ 4.7

ตารางที่ 4.8 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยอร์ที่ระยะ 20 cm.

Parameters	values
Max Yaw error	0.2 องศา
Min Yaw error	0.2 องศา
ค่าเฉลี่ย Yaw error	0.2 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.0
ความแปรปรวน	0.0

4.2.3 ผลการวัดค่าความผิดพลาดในแนวแกน พิทช์ และ ยอร์ ขณะอยู่ที่ตำแหน่ง 30 cm.  
ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนพิทช์ (Y)



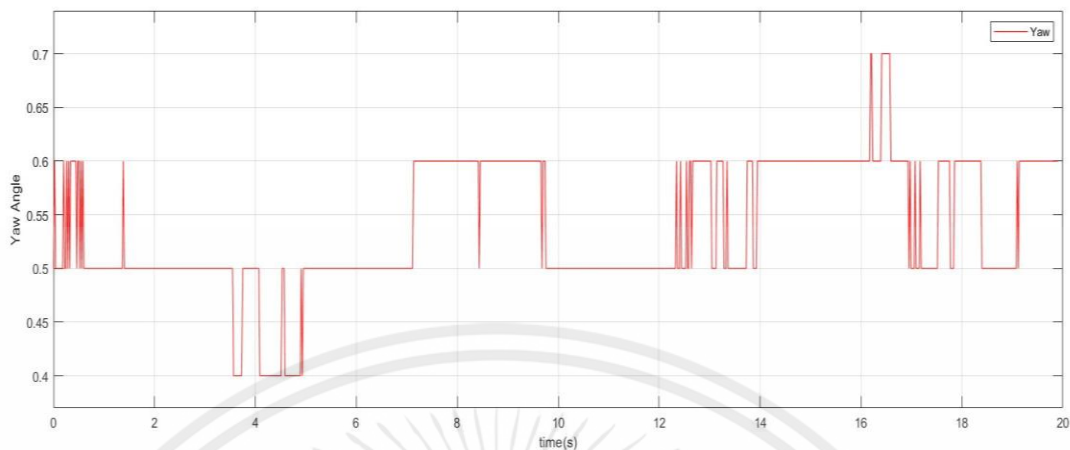
รูปที่ 4.8 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนพิทช์ที่ระยะ 30 cm.

จากผลการทดลองตามรูปที่ 4.8 จะได้อ่านค่าตัวแปรต่าง ๆ ที่ได้ดังตารางที่ 4.8

ตารางที่ 4.9 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนพิทช์ที่ระยะ 30 cm.

Parameters	values
Max Pitch error	1.5 องศา
Min Pitch error	-0.5 องศา
ค่าเฉลี่ย Pitch error	0.3772 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.3002
ความแปรปรวน	0.0901

ผลการวัดค่าความผิดพลาดที่เกิดขึ้นในแนวแกนยอร์ (Z)



รูปที่ 4.9 แสดงผลตอบสนองในการวัดค่าผิดพลาดตามแนวแกนยอร์วี่ระยะ 30 cm.

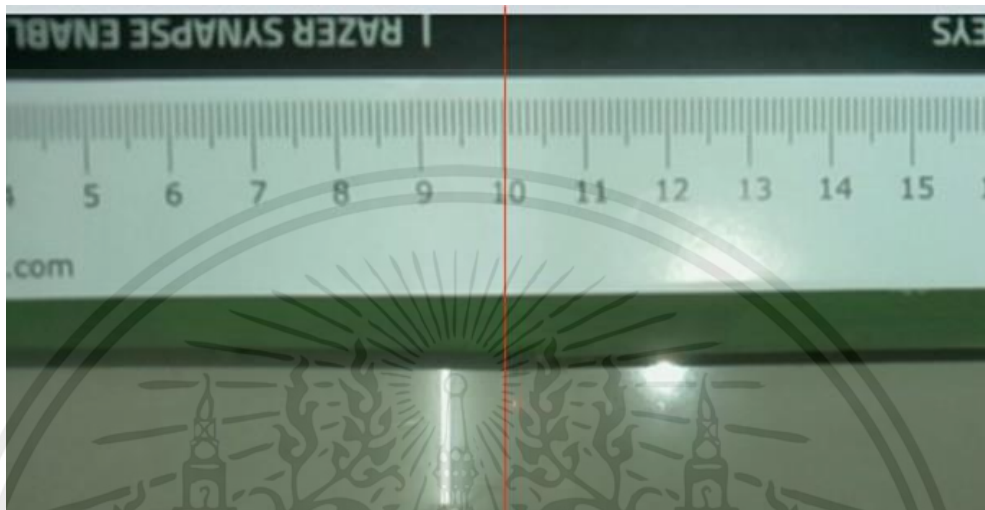
จากผลการทดลองตามรูปที่ 4.9 จะได้ค่าตัวแปรต่าง ๆ ที่ได้ดังตารางที่ 4.9

ตารางที่ 4.10 ตารางแสดงค่าพารามิเตอร์ของการวัดค่าผิดพลาดตามแนวแกนยอร์วี่ระยะ 30 cm.

Parameters	values
Max Yaw error	0.7 องศา
Min Yaw error	0.4 องศา
ค่าเฉลี่ย Yaw error	0.5379 องศา
ส่วนเบี่ยงเบนมาตรฐาน	0.5954
ความแปรปรวน	0.3545

### 4.3 ผลการทดลองของชุดแทนเลื่อนจากการถ่ายภาพที่ได้ในระบบจริง

#### 4.3.1 ผลของการถ่ายภาพจากกล้องบนชุดแทนเลื่อนที่ระยะ 10 cm.



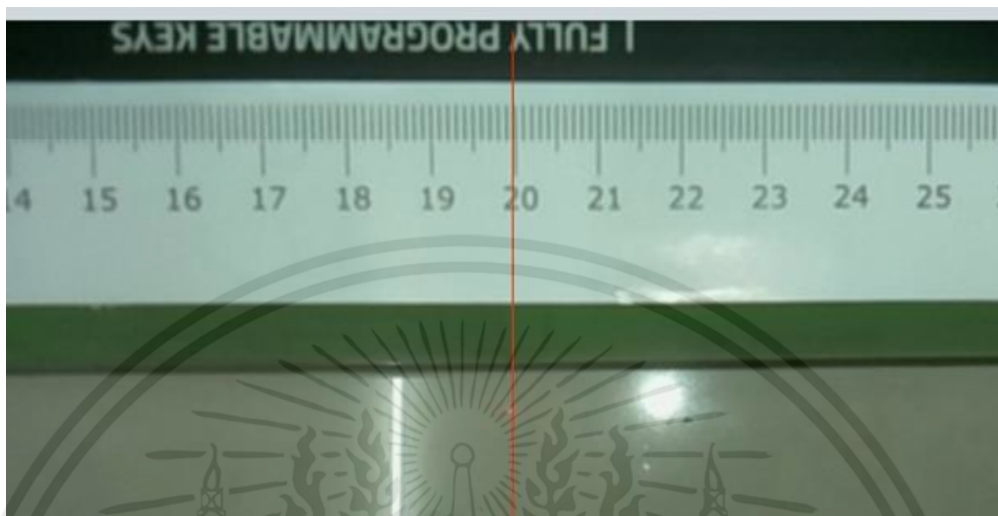
รูปที่ 4.10 แสดงผลภาพที่เลื่อนไปที่ระยะ 10 cm. จากกล้องบนชุดแทนเลื่อน



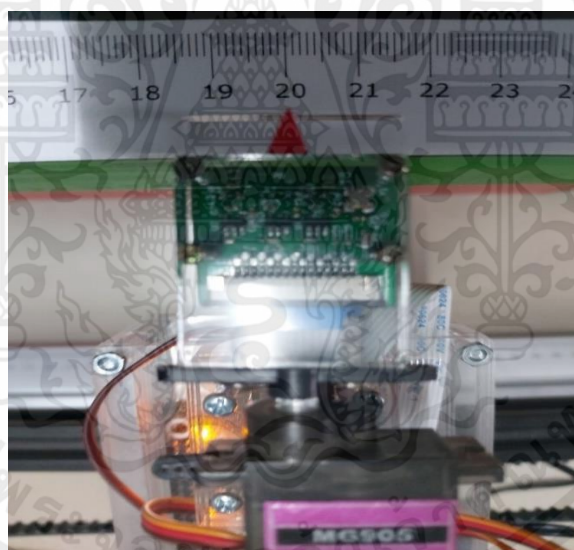
รูปที่ 4.11 แสดงผลของชุดแทนเลื่อนที่เลื่อนไปที่ระยะ 10 cm. ก่อนถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.2 ผลของการถ่ายภาพจากกล้องบนชุดแทนเลื่อนที่ระยะ 20 cm.



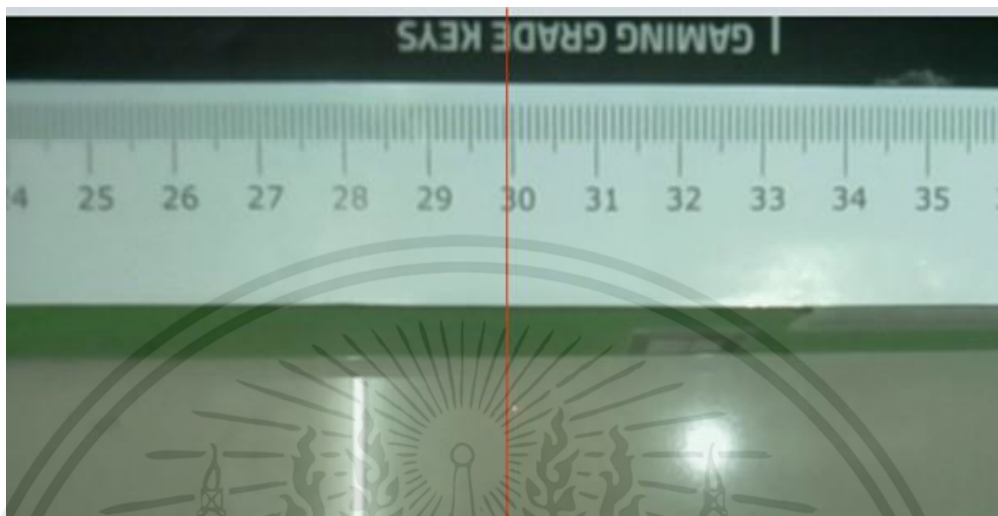
รูปที่ 4.12 แสดงผลภาพที่เลื่อนไปที่ระยะ 20 cm. จากกล้องบนชุดแทนเลื่อน



รูปที่ 4.13 แสดงผลของชุดแทนเลื่อนที่เลื่อนไปที่ระยะ 20 cm. ก่อนถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.3 ผลของการถ่ายภาพจากกล้องบนชุดแทนเลื้อนที่ระยะ 30 cm.



รูปที่ 4.14 แสดงผลภาพที่เลื่อนไปที่ระยะ 30 cm. จากกล้องบนชุดแทนเลื้อน



รูปที่ 4.15 แสดงผลของชุดแทนเลื้อนที่เลื่อนไปที่ระยะ 30 cm. ก่อนถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการทดลองและข้อเสนอแนะ

### 5.1 สรุปผลการทดลอง

จากผลการทดสอบการเคลื่อนที่ของชุดแท่นเลื่อน โดยใช้ไมโครคอนโทรลเลอร์ STM32F401RE ที่มีความเร็วการคำนวณพารามิเตอร์ในระดับมิลลิวินาทีใช้ร่วมกับโปรแกรม STM32CubeIDE ในการเขียนสมการตัวควบคุมพีไอดี โดยมีการเลือกค่าเป้าหมายของการทดลองการเคลื่อนที่อยู่ที่ 10, 20 และ 30 cm. ตามลำดับ และ ใช้พารามิเตอร์ PID ที่ได้จากการออกแบบไว้ ปรากฏว่า ค่าพุงเกินที่ค่าเป้าหมายของการทดลอง ณ เป้าหมายต่าง ๆ มีค่า 0 % และ ค่าเวลาเข้าที่มีค่า 20.2147 14.4663 และ 10.8875 ตามลำดับ ซึ่งผลการทดลองแสดงให้เห็นว่าสามารถควบคุมระยะการเคลื่อนที่ของแท่นเลื่อนให้เป็นไปตามระยะที่ต้องการได้เป็นอย่างดี

และจากผลการทดลองการวัดค่าความผิดพลาด ทั้งในแนวแกนพิทช์และแนวแกนยอร์ โดยจะใช้ไมโครคอนโทรลเลอร์ Raspberry Pi 3 ในการเขียนสมการตัวควบคุมพีไอดีด้วยภาษา Python ขณะอยู่ที่ตำแหน่ง 10, 20 และ 30 cm. ตามลำดับ และใช้พารามิเตอร์ PID ที่ได้จากการออกแบบไว้ ปรากฏว่าค่าความผิดพลาดเฉลี่ยที่เกิดขึ้นในแนวแกนพิทช์ มีค่า 0.2004 0.2176 และ 0.2000 ตามลำดับ ซึ่งจะเห็นว่าค่าความผิดพลาดในแนวแกนพิทช์ของการทดลอง ณ เป้าหมายต่าง ๆ มีค่าใกล้เคียงกัน ส่วนค่าความผิดพลาดเฉลี่ยที่เกิดขึ้นในแนวแกนยอร์ มีค่า 0.1884 0.2000 และ 0.3772 ตามลำดับ จะเห็นว่าค่าความผิดพลาด ณ ตำแหน่ง 30 cm. มีค่ามากที่สุด 20 cm. รองลงมา และที่ตำแหน่ง 10 cm. มีค่าความผิดพลาดน้อยที่สุด ซึ่งผลการทดลองแสดงให้เห็นว่าสามารถควบคุมค่าความผิดพลาดในแนวแกน พิตช์ และ ยอร์ ได้ เนื่องจากค่าความผิดพลาดที่เกิดขึ้นมีค่าน้อยมาก

### 5.2 ปัญหาและอุปสรรค

1. เนื่องจากสถานการณ์ จากสถานการณ์ COVID – 19 ส่งผลให้ผู้ทำการวิจัยไม่สามารถ เข้าไปปฏิบัติงานในห้องทดลองได้ ขาดเครื่องมือหรืออุปกรณ์บางอย่าง ทำให้การดำเนินงานต่าง ๆ ในการทำการวิจัยเกิดความล่าช้า รวมทั้งอุปกรณ์ทางด้านฮาร์ดแวร์บางส่วนต้องสั่งจากต่างประเทศ จึงต้องใช้ระยะเวลาในการติดต่อ และ จัดส่งเนื่องจากสถานการณ์การระบาดของไวรัสโควิด-19 ทำให้เกิดปัญหาทางการเชื่อมต่อ และสื่อสารระหว่างอุปกรณ์ จึงส่งผลให้เกิดการล่าช้าไม่ตรงไปตามแผนงานที่วางไว้

2. การใช้กล้องในการดูตำแหน่งของระยะการเคลื่อนที่ของแท่นเลื่อน มีปัจจัยที่ส่งผลให้การดูสเกลบนสายเทปมาตรฐานอ้างอิงไม่ชัดเจน คือ แสงไฟ ซึ่งหากความสว่างภายในห้องขณะทำการทดลองมีน้อยเกินไปหรือมากเกินไป จะทำให้ภาพที่ด้านจากกล้องไม่มีความชัดเจน และ ส่งผลต่อความไม่แน่นอนในการสอบเทียบด้วย จึงต้องทำการปรับสภาพของแสงไฟขณะทำการสอบเทียบให้สามารถมองเห็นสเกลบนสายเทปชัดเจนที่สุด

3. ในระหว่างการทดลองที่มีการเชื่อมต่อ Raspberry Pi 3 กับ VNC Viewer บนคอมพิวเตอร์ โดยใช้ไอซ์แคปเตอร์ไร้สาย ซึ่งปัญหาที่เกิดขึ้นระหว่างการทดลองคือ การขาดการเชื่อมต่อกันของ Raspberry Pi3 กับ VNC Viewer จึงต้องเริ่มการเชื่อมต่อใหม่ทุกครั้ง ทำให้การทำการทดลองเกิดความล่าช้าเพิ่มขึ้น

### 5.3 ข้อเสนอแนะสำหรับการวิจัยในอนาคต

1. จากผลการทดลองแสดงให้เห็นว่าสามารถควบคุมระยะการเคลื่อนที่ของแท่นเลื่อนให้เป็นไปตามระยะที่ต้องการได้แล้ว แต่สายเทปวัดระยะอ้างอิงที่ใช้จริงไม่ได้เป็นมาตรฐานสำหรับการสอบเทียบ ดังนั้นในส่วนต่อไปจึงควรทำการสอบเทียบกับสายเทปวัดระยะมาตรฐานสำหรับการสอบเทียบ และหาค่าความไม่แน่นอนในการสอบเทียบที่เกิดขึ้น ทำการวิเคราะห์ค่าความไม่แน่นอน เพื่อให้ค่าความไม่แน่นอนที่เกิดขึ้นให้เป็นไปตามเกณฑ์ของมาตรฐาน OIML

2. ในการออกแบบพารามิเตอร์ PID ควรทำการหาฟังก์ชันถ่ายโอน ซึ่งได้จากการทำ System Identification โดยโปรแกรม Matlab รวมถึงออกแบบตัวควบคุมพีไอดีจากทฤษฎีต่าง ๆ เช่น ทฤษฎีทางเดินราก (Root Locus) เป็นต้น

3. เนื่องจากแสงและความส่องสว่าง มีผลต่อความชัดเจนของภาพที่มองเห็นขณะใช้กล้องเพื่อดูตำแหน่งของระยะการเคลื่อนที่ของแท่นเลื่อน ดังนั้นควรมีการติดตั้งไฟ LED เพิ่มเติมบนชุดแท่นเลื่อนเพื่อแก้ไขปัญหานี้

4. เนื่องจากปัญหาในการขาดการเชื่อมต่อระหว่าง Raspberry Pi 3 กับ VNC Viewer บนคอมพิวเตอร์ที่เกิดขึ้นบ่อยครั้ง ดังนั้นจึงควรใช้การเชื่อมต่อที่มีความเสถียร เช่น ต่อสาย LAN โดยตรง หรือการใช้ไอซ์แคปเตอร์ไร้สายที่มีคุณภาพสูงขึ้น

5. เนื่องจากในโครงการประกอบไปด้วย 2 ส่วนการทำงานหลัก คือ ส่วนของการควบคุมการเคลื่อนที่ของชุดแท่นเลื่อน และส่วนของการควบคุมการสั่นของชุดแท่นเลื่อนซึ่งใช้ ไมโครคอนโทรลเลอร์ ที่ต่างกัน ดังนั้นเพื่อให้ระบบมีการทำงานที่สัมพันธ์ จึงควรที่จะใช้ ไมโครคอนโทรลเลอร์ตัวเดียวกันในการควบคุม

## บรรณานุกรม

- [1] วันชัย ชินชูศักดิ์. 2561. “เทคนิคการประเมินค่าความไม่แน่นอนของระบบสอบเทียบสายวัด.” วิทยานิพนธ์วิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [2] Wilhelm Andrén. Ella Hjertberg . 2019. “Gyro Stabilization of a Positioning Unit.” Department of Automatic Control Lund University.
- [3] ทฤษฎีและหลักการ. [Online] สืบค้นเมื่อ 19 มีนาคม 2564. Available : <https://sites.google.com/site/physic122/about?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>
- [4] AKM. Principle and advantages of magnetic encoder. [Online] Retrieved April 2, 2020. Available : <https://www.akm.com/kr/ko/technology/technical-tutorial/basic-knowledge-encoder/magnetic-encoder>
- [5] JSS. 2556. การทดลอง CS06: ลักษณะสมบัติของตัวควบคุม PID. ห้องปฏิบัติการพื้นฐานระบบควบคุม ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย.
- [6] thailandindustry. 2553. วิธีการปรับค่าเกนตัวควบคุมแบบ PID กรณีไม่ทราบแบบจำลองทางคณิตศาสตร์ของกระบวนการ (ตอนที่ 1). [Online] สืบค้นเมื่อ 2 เมษายน 2564 . Available : [http://www.thailandindustry.com/indust\\_newweb/articles\\_preview.php?cid=10522](http://www.thailandindustry.com/indust_newweb/articles_preview.php?cid=10522)
- [7] Navapat Jesadapatrakul. 2562. Image Processing. [Online] สืบค้นเมื่อ 10 เมษายน 2564. Available : <https://medium.com/tni-university/image-processing>
- [8] raspberryhome.net. 2561. การติดตั้ง Raspbian บน Raspberry Pi แบบไม่ต่อจอภาพ. [Online]. Available : <https://nextsoftwares.wordpress.com/2014/05/22>
- [9] มนูญศักดิ์ วรรณประพันธ์. เอกพันธ์ อินดี. 2556. “โปรแกรมตรวจจับโมเดลส่วนท้ายรถยนต์แบบอัตโนมัติ.” ปริญญาโทวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม.

- [10] raspberryhome.net. 2561. การติดตั้ง Raspbian บน Raspberry Pi แบบไม่ต่อจอภาพ. [Online] สืบค้นเมื่อ 5 พฤษภาคม 2564.  
Available : <http://www.raspberryhome.net/b/1>
- [11] STMicroelectronics. 2021. **Integrated Development Environment for STM32.** [Online] Retrieved May 5, 2020.  
Available : <https://www.st.com/en/development-tools/stm32cubeide.html>
- [12] SUPPORT THAIEASYELEC. 2561. การพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt. [Online] สืบค้นเมื่อ 5 พฤษภาคม 2564.  
Available : <https://blog.thaieasyelec.com/raspberry-pi-programming-with-qt-ch1/>
- [13] Adeept. 2021. **Lesson 2 Introduction of Robot HAT.** [Online] Retrieved May 10, 2020. Available : <https://www.adeept.com/learn/tutorial-159.html>
- [14] myarduino. 2564. **TB6612FNG Motor Driver Module บอร์ดขับมอเตอร์ 2 ช่อง.** [Online] สืบค้นเมื่อ 10 พฤษภาคม 2564.  
Available : <https://www.myarduino.net/p/790>
- [15] Adeept. 2020. **Lesson 11 Reading Data from MPU6050.** [Online] Retrieved May 10, 2020.  
Available : <https://www.adeept.com/learn/tutorial-170.html>
- [16] Adeept. 2020. **Lesson 5 How to Control 180° Servo.** [Online] Retrieved May 10, 2020 .  
Available : <https://www.adeept.com/learn/tutorial-162.html>
- [17] EDIMAX. 2020. **EW-7811Un V2.** [Online] สืบค้นเมื่อ 10 พฤษภาคม 2564.  
Available : [https://www.edimax.com/edimax/merchandise/merchandise\\_detail/data/edimax/global/wireless\\_adapters\\_n150/ew-7811un\\_v2/](https://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/global/wireless_adapters_n150/ew-7811un_v2/)
- [18] raspberryhome.net. 2561. การติดตั้ง Raspbian บน Raspberry Pi แบบไม่ต่อจอภาพ. [Online] สืบค้นเมื่อ 27 พฤษภาคม 2564.  
Available : <http://www.raspberryhome.net/b/1>

- [19] kmutnb. 2560. “การใช้งานโปรแกรม GUI-MATLAB และการประยุกต์ใช้งาน”. [Online] 27 พฤษภาคม 2564.  
Available : <https://icit.kmutnb.ac.th/main/gui-matlab>
- [20] สุรวิชญ์ เหลืองทอง. 2556. ความรู้เกี่ยวกับ VNC. [Online] สืบค้นเมื่อ 27 พฤษภาคม 2564.  
Available : <http://surawit555.myreadyweb.com/page-24996.html>
- [21] stackpython. 2563. IDE คืออะไร,แนะนำ IDE. [Online] สืบค้นเมื่อ 27 พฤษภาคม 2564.  
Available : <https://stackpython.medium.com/ide>
- [22] 7svv. 2562. มาฝึกภาษา Python ด้วย Pycharm กันเถอะ ! . [Online] สืบค้นเมื่อ 28 พฤษภาคม 2564. Available : <https://medium.com/@vitsavit>





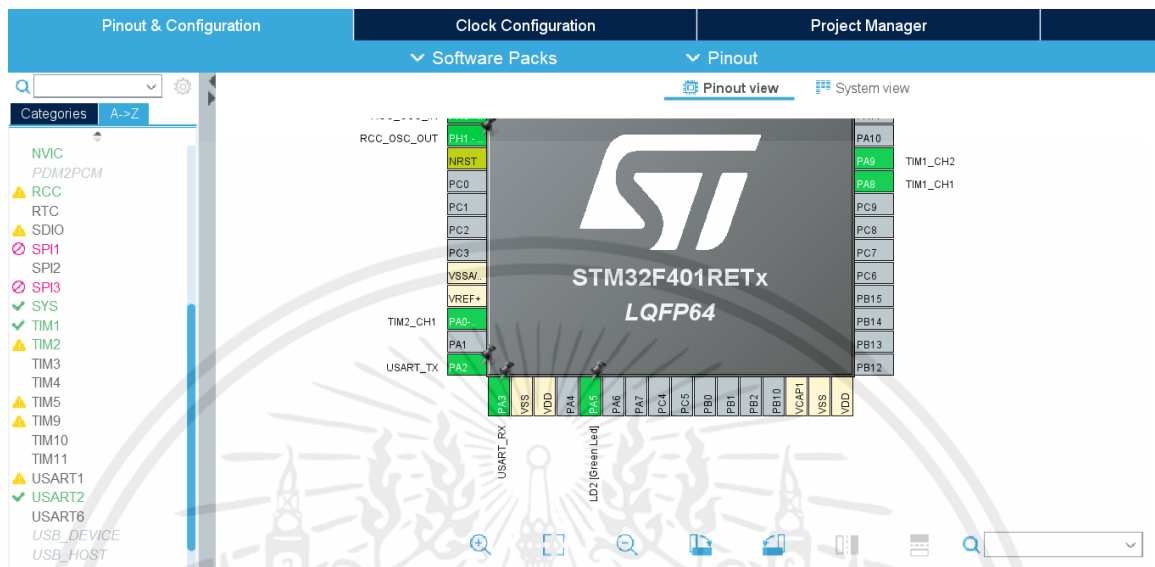
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก  
การเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก.1 โปรแกรม STM32CubeIDE Code



รูป ก.1.1 ตั้งค่า Pinout และ พารามิเตอร์ ก่อนการเขียนโปรแกรม

```

20 /* Includes -----
21 #include "main.h"
22
23 /* Private includes -----
24 /* USER CODE BEGIN Includes */
25 #include "stm32f4xx_hal.h"
26 #include "arm_math.h"
27 #include <stdio.h>
28 #include <string.h>
29 #include "math.h"
30 /* USER CODE END Includes */

```

รูปที่ ก.1.2 แสดงการกำหนดไลบรารี

```

53 /* USER CODE BEGIN PV */
54 /* Choose PID parameters */
55 #define PID_PARAM_KP      0.0001      /* Proportional */
56 #define PID_PARAM_KI      0.0001      /* Integral */
57 #define PID_PARAM_KD      1.8          /* Derivative */
58 /* USER CODE END PV */

```

รูปที่ ก.1.3 แสดงการกำหนดค่าพารามิเตอร์ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

91 /* USER CODE BEGIN 0 */
92 int puls;
93 int a=0;
94 int b=50;
95 /* USER CODE END 0 */

```

รูปที่ ก.1.4 แสดงพารามิเตอร์รับ ส่งค่า

```

103 /* USER CODE BEGIN 1 */
104 float setpoint=0;
105 float feedback=0;
106 char mgs[30];
107
108 /* PID error */
109 float pid_error;
110 /* Duty cycle for PWM */
111 float duty=0;
112 /* ARM PID Instance, float_32 format */
113 arm_pid_instance_f32 PID;
114
115 /* Set PID parameters */
116 /* Set this for your needs */
117 PID.Kp = PID_PARAM_KP; /* Proportional */
118 PID.Ki = PID_PARAM_KI; /* Integral */
119 PID.Kd = PID_PARAM_KD; /* Derivative */
120
121 /* Initialize PID system, float32_t format */
122 arm_pid_init_f32(&PID, 1);
123 /* USER CODE END 1 */

```

รูปที่ ก.1.5 แสดงตัวแปรที่รับค่า error และ duty

```

148 //PWM
149 HAL_TIM_Base_Start(&htim2);
150 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
151
152 //Encoder
153 HAL_TIM_Encoder_Start(&htim1, TIM_CHANNEL_ALL);

```

```

162 //encoder
163
164 encoder = TIM1->CNT;
165
166 if (encoder>2631)
167     encoder=0;
168 if (encoder>=0&&encoder<2630)
169     encoder=encoder;
170 else
171     encoder=0;
172
173 //Calculate
174
175 setpoint=30;
176
177 feedback = encoder*(49.27/2616);

```

รูปที่ ก.1.6 แสดงการตั้ง Setpoint และ คำนวณ feedback

```

179 //Over flow
180 if(setpoint<=0){
181     setpoint=0;
182 }
183 else if(setpoint>49){
184     setpoint=49;
185 }
186
187 /* Calculate error */
188 pid_error = setpoint-feedback;
189
190 /* Calculate PID here, argument is error */
191 /* Output data will be returned, we will use it as duty cycle parameter */
192 duty = arm_pid_f32(&PID, pid_error);
193
194 //duty =50;
195 /* Check overflow, duty cycle in percent*/
196 if (duty > 100) {
197     duty = 100;
198 } else if (duty < 0) {
199     duty = 0;
200 }

```

รูปที่ ก.1.7 แสดงการกำหนดเงื่อนไขในการเคลื่อนที่ และ คำนวณ error

```

202 //PWM
203 puls =(duty/100)*83999;
204 htim2.Instance->CCR1=puls;
205
206 if(setpoint-feedback>=0){
207     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_10, GPIO_PIN_SET);
208     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_11, GPIO_PIN_RESET);
209 }
210 if(setpoint-feedback<0){
211     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_11, GPIO_PIN_SET);
212     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_10, GPIO_PIN_RESET);
213 }

```

รูปที่ ก.1.8 แสดงการส่งค่า output เพื่อควบคุม DC Motor

```

216 /* Format string */
217 sprintf(mgs, "%.2f %.2f %d %d\r\n", setpoint, feedback, a, b);
218 //sprintf(mgs, " %d %d %d %d", 0, 100, 10, 50); encoder
219 //sprintf(mgs, "%d - %.2f \n", encoder, feedback);
220 HAL_UART_Transmit(&huart2, (uint8_t*)mgs, strlen(mgs), HAL_MAX_DELAY);
221 /* USER CODE END WHILE */

```

รูปที่ ก.1.9 การส่งข้อมูลเพื่อแสดงผล

## ก.2 โปรแกรม Thony IDE Code

```

move.py *%
1 import time
2 import Adafruit_PCA9685
3 #from mpu6050 import mpu6050
4 #import Kalman filter
5 import PID
6 import threading
7 #import RPiServo
8 import smbus
9 import math
10
11
12 '''
13 change these variable to adjust the steady function.
14 '''
15 steady_range_Min = -40
16 steady_range_Max = 130
17 range_Mid = (steady_range_Min+steady_range_Max)/2
18 P_fix_output = range_Mid
19 Y_fix_output = range_Mid
20 steady_X_set = 73
21
22 '''
23 Set PID
24 '''
25 P = 5
26 I = 0.0001
27 D = 0
28
29 '''
30 >>> instantiation <<<
31 '''

```

รูปที่ ก.2.1 แสดงไลบรารีที่ใช้งาน และการเซ็ตค่าต่าง ๆ

```

move.py *x
30 >>> instantiation <<<
31 ...
32 P_pid = PID.PID()
33 P_pid.SetKp(P)
34 P_pid.SetKd(I)
35 P_pid.SetKi(D)
36 Y_pid = PID.PID()
37 Y_pid.SetKp(P)
38 Y_pid.SetKd(I)
39 Y_pid.SetKi(D)
40 pwm = Adafruit_PCA9685.PCA9685()
41 pwm.set_pwm_freq(50)
42
43 target_P = 0
44 target_Y = 0
45
46
47
48 class MPU:
49     def __init__(self, gyro, acc, tau):
50         # Class / object / constructor setup
51         self.gx = None; self.gy = None; self.gz = None;
52         self.ax = None; self.ay = None; self.az = None;
53
54         self.gyroXcal = 0
55         self.gyroYcal = 0
56         self.gyroZcal = 0
57
58         self.gyroRoll = 0
59         self.gyroPitch = 0
60         self.gyroYaw = 0

```

รูปที่ ก.2.2 แสดงตัวแปรที่เก็บค่า PID จากไลบรารี และ Class MPU

```

move.py *x
90 .get(x,[12040.0, 0x10])
91 }
92 }.get(x,[8192.0, 0x08])
93
94 def setUp(self):
95     # Activate the MPU-6050
96     self.bus.write_byte_data(self.address, 0x6B, 0x00)
97     # Configure the accelerometer
98     self.bus.write_byte_data(self.address, 0x1C, self.accHex)
99
100     # Configure the gyro
101     self.bus.write_byte_data(self.address, 0x1B, self.gyroHex)
102
103     pwm = Adafruit_PCA9685.PCA9685()
104     pwm.set_pwm_freq(50)
105
106     #pos_init = 360
107     #pos_max = 400
108     #pos_min = 200
109
110     pwm.set_pwm(7,0,360)
111     time.sleep(1)
112     pwm.set_pwm(8,0,340)
113     time.sleep(1)
114
115     # Display message to user
116     print("MPU&SERVO set up:")
117     print('\tAccelerometer: ' + str(self.accHex) + ' ' + str(self.accScaleFactor))
118     print('\tGyro: ' + str(self.gyroHex) + ' ' + str(self.gyroScaleFactor) + "\n")
119     time.sleep(2)
120
121 def eightBit2sixteenBit(self, real):

```

รูปที่ ก.2.3 แสดงฟังก์ชันสำหรับการเซ็ตค่า MPU

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

move.py *x
138     self.ax = self.eightBit2sixteenBit(0x3B)
139     self.ay = self.eightBit2sixteenBit(0x3D)
140     self.az = self.eightBit2sixteenBit(0x3F)
141
142     def calibrateGyro(self, N):
143         # Display message
144         print("Calibrating gyro with " + str(N) + " points. Do not move!")
145
146         # Take N readings for each coordinate and add to itself
147         for ii in range(N):
148             self.getRawData()
149             self.gyroXcal += self.gx
150             self.gyroYcal += self.gy
151             self.gyroZcal += self.gz
152
153         # Find average offset value
154         self.gyroXcal /= N
155         self.gyroYcal /= N
156         self.gyroZcal /= N
157
158         # Display message and restart timer for comp filter
159         print("Calibration complete")
160         print("\tX axis offset: " + str(round(self.gyroXcal,1)))
161         print("\tY axis offset: " + str(round(self.gyroYcal,1)))
162         print("\tZ axis offset: " + str(round(self.gyroZcal,1)) + "\n")
163         time.sleep(2)
164         self.dtTimer = time.time()
165
166     def processIMUvalues(self):
167         # Update the raw data
168         self.getRawData()

```

รูปที่ ก.2.4 แสดงฟังก์ชันในการ calibrate MPU

```

move.py *x
182     self.gy /= self.accScaleFactor
183     self.az /= self.accScaleFactor
184
185     def compFilterAndST(self):
186         # Get the processed values from IMU
187         self.processIMUvalues()
188
189         # Get delta time and record time for next call
190         dt = time.time() - self.dtTimer
191         self.dtTimer = time.time()
192
193         # Acceleration vector angle
194         accPitch = math.degrees(math.atan2(self.ay, self.az))
195         accRoll = math.degrees(math.atan2(self.ax, self.az))
196
197         # Gyro integration angle
198         self.gyroRoll -= self.gx * dt
199         self.gyroPitch += self.gy * dt
200         self.gyroYaw += self.gz * dt
201         self.yaw = self.gyroYaw
202
203         # Comp filter
204         self.roll = (0.98)*(self.roll - self.gx*dt) + (1-0.98)*(accRoll)
205         self.pitch = (0.98)*(self.pitch + self.gy*dt) + (1-0.98)*(accPitch)
206
207         # Print data
208         ...
209
210         print(" R: " + str(round(self.pitch,1)) \
211               + " P: " + str(round(self.roll,1)) \
212               + " Y: " + str(round(self.yaw,1)))
213         ...

```

รูปที่ ก.2.5 แสดงฟังก์ชันสำหรับแปลงค่าจาก MPU เป็นมุมรวมถึงทำ Steady ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

move.py *x
230     P = Pitch_A
231     Y = Yaw_A
232
233     P_error = P - target_P
234     #move_Pitch = 100+(1727.44/(P_error))
235
236     Y_error = Y - target_Y
237     #move_Yaw = 100+(1727.44/(Y_error))
238
239     P_fix_output = -P_pid.GenOut(P_error)
240     P_fix_output = ctrl_range(P_fix_output, steady_range_Max, -steady_range_Max)
241
242     Y_fix_output = -Y_pid.GenOut(Y_error)
243     Y_fix_output = ctrl_range(Y_fix_output, steady_range_Max, -steady_range_Max)
244
245     move_P = (2.555*P_fix_output)/(-5)
246     move_Y = (2.555*Y_fix_output)/(-5)
247
248
249     if (P_error)>1 or (Y_error)>1 :
250         pwm.set_pwm(7,0,360-int(move_P))
251         pwm.set_pwm(8,0,340-int(move_Y))
252         print ("1")
253         #time.sleep(0.5)
254
255     elif (Y_error)<-1 or (P_error)<-1 :
256         pwm.set_pwm(7,0,360-int(move_P))
257         pwm.set_pwm(8,0,340-int(move_Y))
258         print ("2")
259         #time.sleep(0.5)
260

```

รูปที่ ก.2.6 แสดงการรับค่า error เข้าไลบรารี PID และ ส่งค่า pwm เพื่อคุมเซอร์โวมอเตอร์

```

move.py *x
369     elif 'speechOff' == command_input:
370         SmoothMode = 0
371         steadyMode = 0
372         rm.pause()
373
374     def main():
375         # Set up class
376         gyro = 250         # 250, 500, 1000, 2000 [deg/s]
377         acc = 2           # 2, 4, 7, 16 [g]
378         tau = 0.98
379         mpu = MPU(gyro, acc, tau)
380
381         # Set up sensor and calibrate gyro with N points
382         mpu.setUp()
383         mpu.calibrateGyro(500)
384
385         # Run for 20 seconds
386         #startTime = time.time()
387         while True:
388             mpu.compFilterAndST()
389             time.sleep(0.02)
390
391         # End
392         print("Closing")
393
394
395     if __name__ == '__main__':
396         main()
397
398     main()
399

```

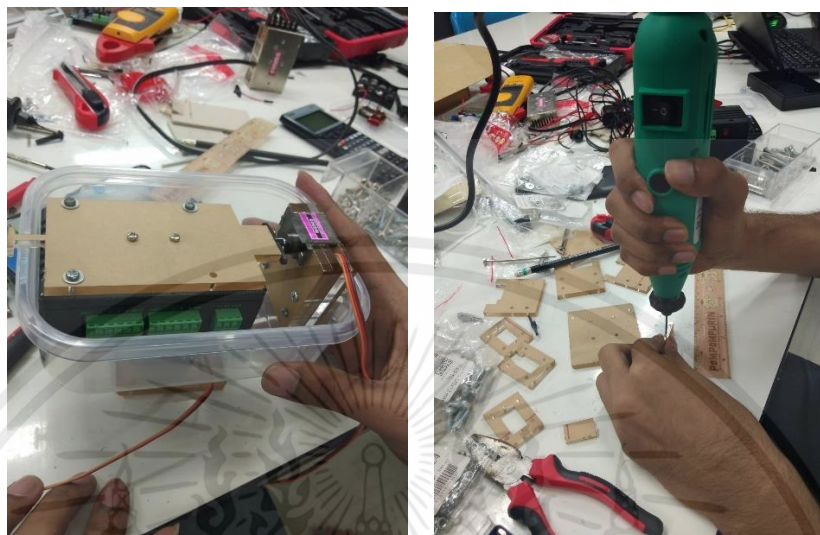
รูปที่ ก.2.7 แสดงฟังก์ชัน main

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

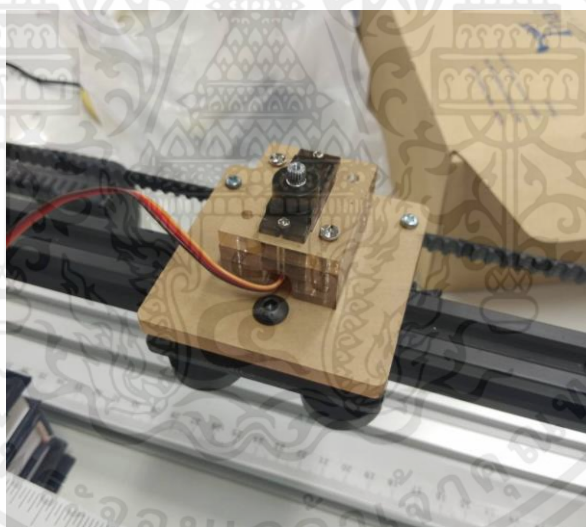


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ข.1 การออกแบบและการประกอบฮาร์ดแวร์โมเดลที่ 1

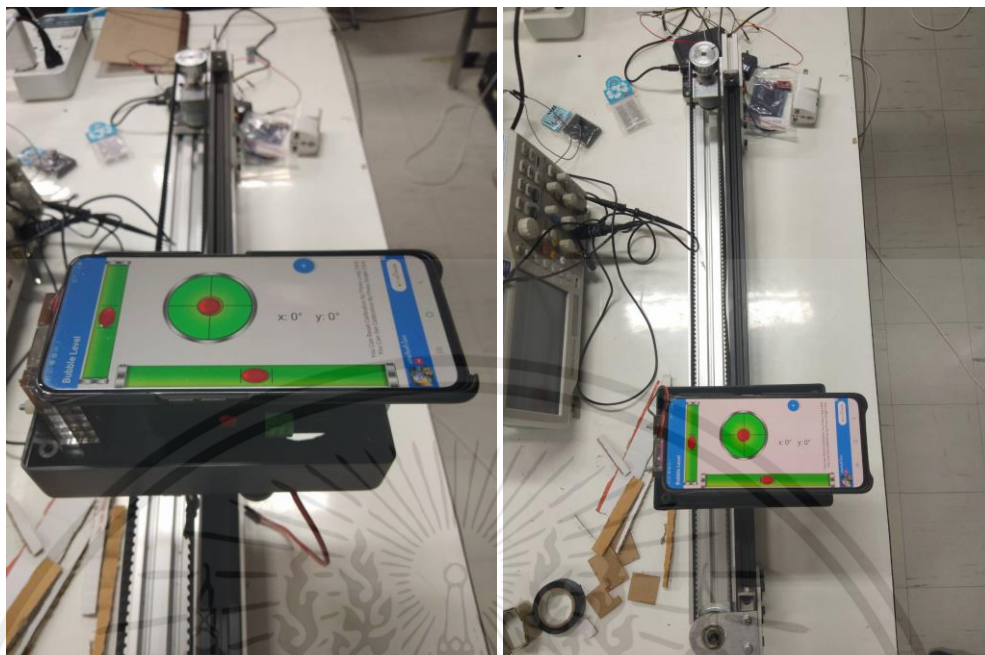


รูปที่ ข.1.1 แสดงการเจาะและประกอบโครงสร้างโมเดลที่ 1

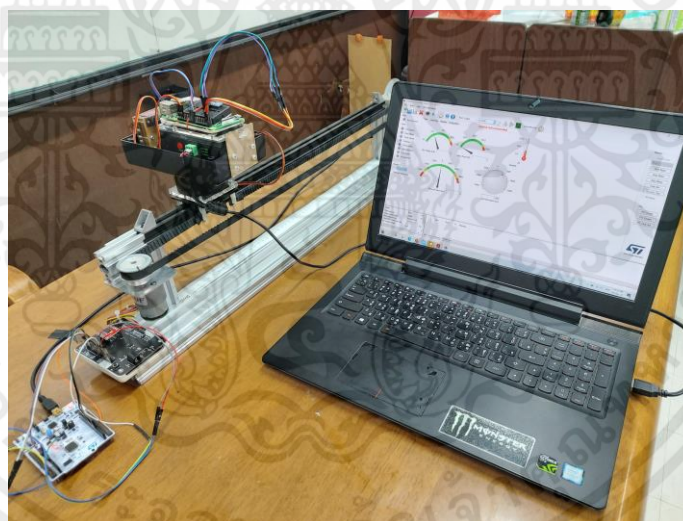


รูปที่ ข.1.2 แสดงการติดตั้งบนแท่นเลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



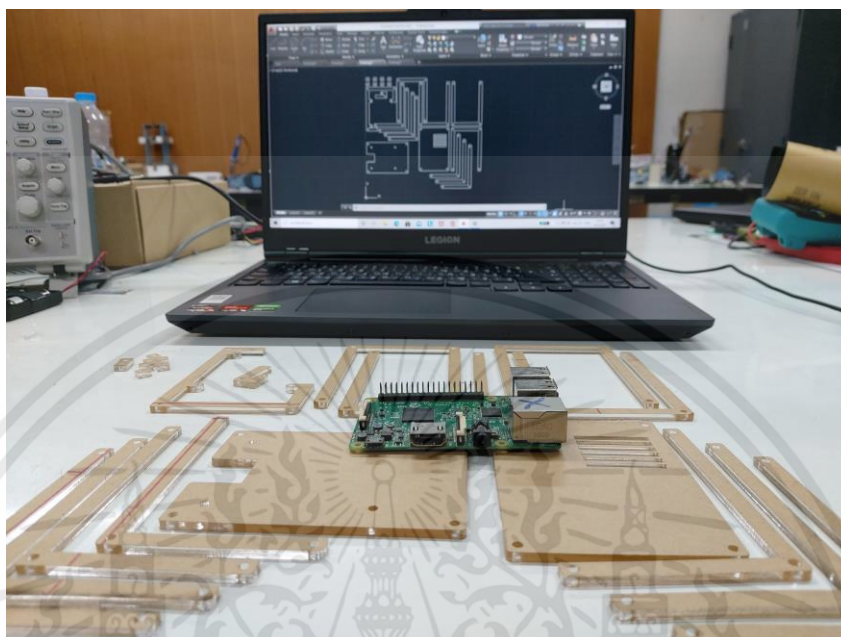
รูปที่ ข.1.3 แสดงการตรวจสอบระดับหลังการติดตั้ง



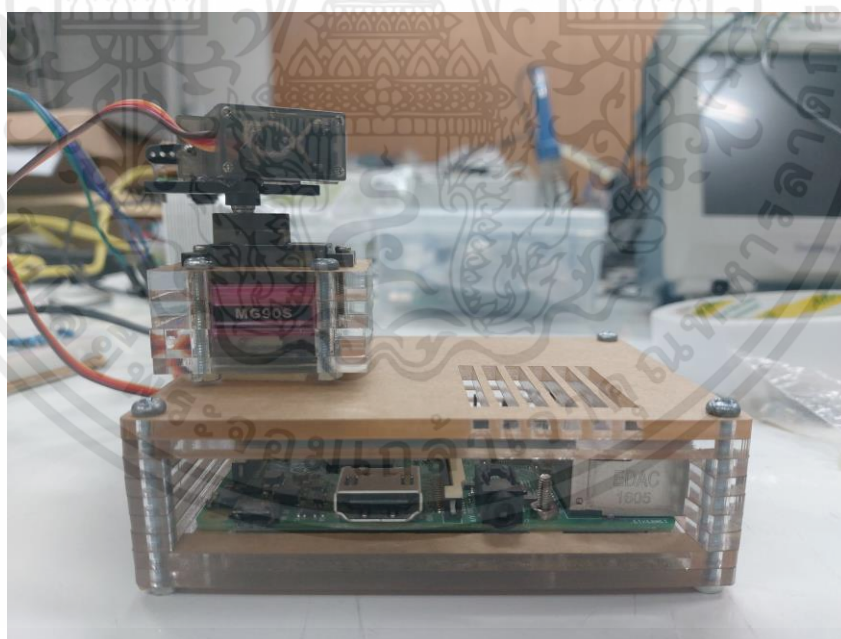
รูปที่ ข.1.4 แสดงการทดสอบการเคลื่อนที่ของโมเดลที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ข.2 การออกแบบและการประกอบฮาร์ดแวร์โมเดลที่ 2

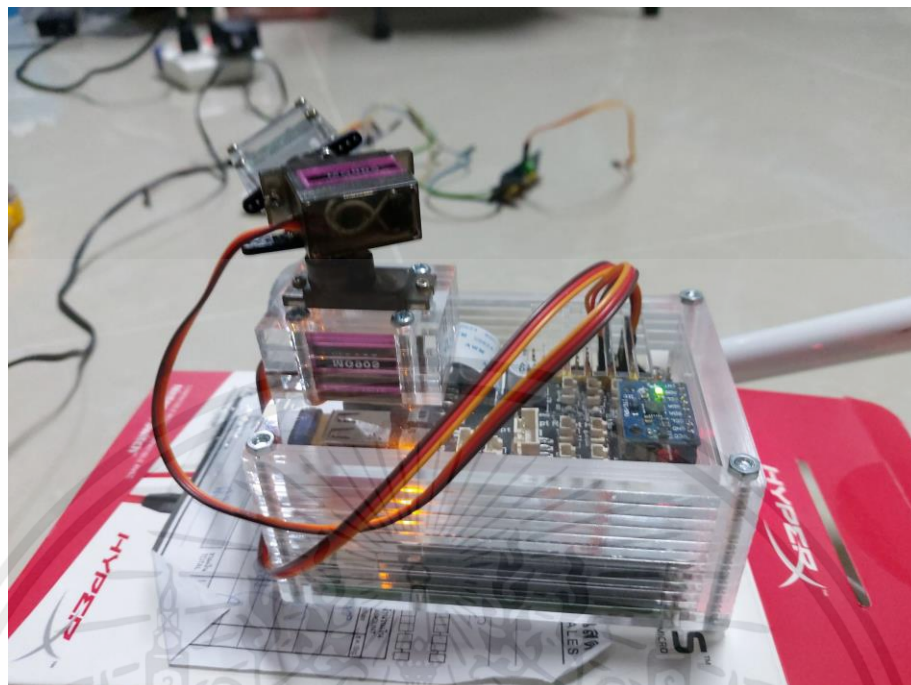


รูปที่ ข.2.1 แสดงการออกแบบโครงสร้างโมเดลที่ 2

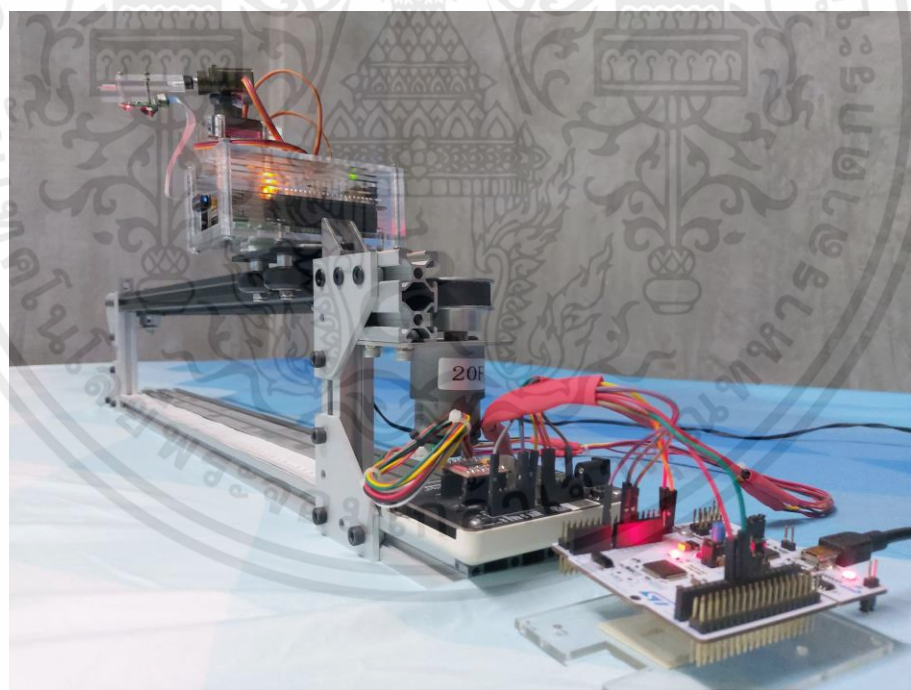


รูปที่ ข.2.2 แสดงโครงสร้างโมเดลที่ 2 ที่ประกอบเสร็จสมบูรณ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2.3 แสดงการทดสอบการควบคุมการสั้นของชุดแทนเลื่อน



รูปที่ ข.2.4 แสดงการติดตั้ง และ ทดสอบในระบบจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**Chinoros Mookrote**  
60010236@kmitl.ac.th

**ประวัติโดยย่อ** นายชินอรัส มุขโรจน์ รหัสนักศึกษา 60010236 นักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จบการศึกษาระดับชั้นมัธยมศึกษาตอนต้น โรงเรียนสาธิตเทศบาลวัดเพชรจริก ระดับชั้นมัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาภาคใต้

**สิ่งที่ได้รับจากการทำปริญญานิพนธ์ครั้งนี้** นอกจากความรู้ และประสบการณ์ในการทำโปรเจกต์ตั้งแต่การเลือกหัวข้อ กระบวนการออกแบบ กระบวนการสร้าง กระบวนการศึกษา และ ทดลองแล้ว ยังได้ฝึกการวางแผน การแก้ปัญหา การคิดอย่างมีหลักการและ เป็นระบบเพื่อให้งานสำเร็จลุล่วง



**Patcharapon Akarapongpaisan**  
60010654@kmitl.ac.th

**ประวัติโดยย่อ** นายพชรพล อัครพงษ์ไพศาล รหัสนักศึกษา 60010654 นักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จบการศึกษาระดับชั้นมัธยมศึกษาตอนต้น และ ระดับชั้นมัธยมศึกษาตอนปลายจาก โรงเรียนชลราษฎรอำรุง

**สิ่งที่ได้รับจากการทำปริญญานิพนธ์ครั้งนี้** จากความรู้ และ ประสบการณ์ที่ได้จากการทำโครงการ นอกจากนี้ยังสามารถนำทักษะที่ได้จากการทำโครงการในหลาย ๆ ด้าน ไปปรับใช้กับการทำงานในอนาคตได้ตั้งแต่ การการแก้ปัญหา การคิดอย่างเป็นระบบ และ มีหลักการ เพื่อให้โครงการได้สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## My Advisors

(Approved Profiles@ May 31, 2021)



**Arjin Numsomran**  
arjin.nu@kmitl.ac.th

**Assoc. Prof. Dr. Arjin Numsomran** received the D.Eng. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMIT'L) in 2018, the M.Eng. degree in Electrical Engineering from KMIT'L in 2001, and the B.Eng. degree in Instrumentation Engineering from KMIT'L in 1998. He is currently an Associate Professor with the Department of Instrumentation and Control Engineering, School of Engineering, KMIT'L, Bangkok. His research interests include Control System (CS) Analysis and Design, Embedded Control System (ECS), Instrumentation and Measurement System (IMS), Artificial Intelligence Internet of Things (AIoT), and Safety Instrumented System (SiS).



**Vittaya Tipsuwannaporn**  
vittaya.ti@kmitl.ac.th

**Assoc. Prof. Dr. Vittaya Tipsuwannaporn** received the D.Eng. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMIT'L) in 2007, the M.Eng. degree in Electrical Engineering from KMIT'L in 1989, the B.Eng. degree in Electrical Engineering from Pathumwan Institute of Technology in 2005, and the B.Ind. from KMIT'L in 1985. His research interests include industrial measurement and instrumentation, power electronic and control drives, Digital and Intelligence Control, Instrumentation Circuit, Industrial Measurement and Instrumentation.