

ระบบควบคุมทิศทางและการเคลื่อนที่รถจำลองด้วยการเคลื่อนไหวศีรษะ  
Car Control Direction and Movement System  
by Head Movement



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมทิศทางและการเคลื่อนที่รถจำลองด้วยการเคลื่อนไหวศีรษะ  
Car Control Direction and Movement System  
by Head Movement



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Car Control Direction and Movement System  
by Head Movement




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 LADKRABANG  
ACADEMIC YEAR 2021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ ระบบควบคุมทิศทางและการเคลื่อนที่รถจำลองด้วย  
การเคลื่อนไหวศีรษะ  
Thesis Title Car Control Direction and Movement System by Head  
Movement  
นักศึกษาผู้จัดทำ นายกิตติพงษ์ เสาร์หิรัญเมธี รหัสนักศึกษา 62015008  
นายสิรภพ เหล่ากาวิ รหัสนักศึกษา 62015118  
นายอภิชาติ แสนบุญ รหัสนักศึกษา 62015127  
ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2564

อาจารย์ที่ปรึกษาปริญญาานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ ทรงชัย วีระทวีมาศ	

หัวข้อวิทยานิพนธ์	ระบบควบคุมทิศทางและการเคลื่อนที่รถจำลองด้วยการเคลื่อนไหวศีรษะ		
Thesis Title	Car Control Direction and Movement System by Head Movement		
นักศึกษาผู้จัดทำ	นายกิตติพงษ์	เสาร์หิรัญเมธี	รหัสนักศึกษา 62015008
	นายสิรภพ	เหล่ากาวิ	รหัสนักศึกษา 62015118
	นายอภิชาติ	แสนบุญ	รหัสนักศึกษา 62015127
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2564		
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ทรงชัย วีระทวีมาศ		

### บทคัดย่อ

โครงการนี้ได้อธิบายเกี่ยวกับรถควบคุมโดยเซ็นเซอร์ Accelerometer จะควบคุมจาก การเชื่อมต่อ บลูทูธ ระหว่างเมาสเตอร์-สเลฟ และรวมถึงการตรวจสอบโดยโมดูลกล้องที่มีการเชื่อมต่อ อินเทอร์เน็ต WIFI หลักของรถควบคุมจะควบคุมโดยเซ็นเซอร์ Accelerometer จากการเคลื่อนไหว ของศีรษะ ค่า Roll ใช้สำหรับการควบคุมเพื่อเลี้ยวซ้ายไปขวา ค่า Pitch ใช้เพื่อควบคุมทิศทางรถไป ข้างหน้าหรือถอยหลัง ส่งข้อมูลที่ได้รับจากค่า Roll และค่า Pitch ส่งข้อมูลจากบลูทูธหลักเพื่อรับค่า ข้อมูลจากบลูทูธเพื่อควบคุมทิศทางและตำแหน่งของรถ โมดูลกล้องจะถูกติดตั้งในด้านหน้าของรถเพื่อ ตรวจสอบตำแหน่งของรถในปัจจุบัน

<b>Thesis Title</b>	Car Control Direction and Movement System by Head Movement
<b>Student</b>	Mr.Kittipong Saohirunmatee Student ID. 62015008 Mr.Sirapob Laokawee Student ID. 62015118 Mr.Apichart Sanboon Student ID. 62015127
<b>Degree</b>	Bachelor of Engineering
<b>Program</b>	Instrumentation Engineering
<b>Year</b>	2021
<b>Thesis Advisor</b>	Assoc. Prof. Songchai Weerathaweemas

### ABSTRACT

This project have explain of control car by accelerometer sensor, will control from Bluetooth master-slave and include monitor by the camera module with internet WIFI connection main of control car will control by the accelerometer sensor from motion of the head And the value of accelerometer have Roll and Pitch value, roll value uses for control to turn left to right, pitch value uses to control forward of the car or reverse car direction. This is described can get from the value of roll and pitch and sent data from master Bluetooth to slave Bluetooth for getting data value to control car direction and position. For camera module will install on the front of car for monitor position of the car currently.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับความปรึกษาและความอนุเคราะห์จาก อาจารย์ที่ปรึกษา รองศาสตราจารย์ ทรงชัย วีระทวีมาศ อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญาานิพนธ์นี้ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์สาขาวิชาวิศวกรรมการวัดและควบคุมทุกท่าน ที่ให้คำแนะนำและ ถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้าอันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้

และที่ลืมเสียไม่ได้คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ซึ่งเป็นที่รักและเคารพยิ่ง ที่ สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญาานิพนธ์ฉบับนี้ คุณค่าและประโยชน์อันพึงมีจาก ปริญญาานิพนธ์ฉบับนี้ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการศึกษา.....	2
<b>บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....</b>	<b>4</b>
2.1 บลูทูธ.....	4
2.1.1 วัตถุประสงค์ของเทคโนโลยี บลูทูธ.....	4
2.1.2 การทำงานของ บลูทูธ.....	5
2.1.3 ลักษณะการเชื่อมต่อ บลูทูธ.....	6
2.1.4 เวอร์ชันบลูทูธ.....	7
2.2 การสื่อสารแบบ SPI.....	10
2.2.1 การทำงานของ SPI.....	11
2.3 การสื่อสารแบบ I2C (I <sup>2</sup> C).....	12
2.3.1 การใช้งานการสื่อสารแบบ I2C.....	13
2.3.2 เวอร์ชันการพัฒนาของการสื่อสารอนุกรมแบบ I2C.....	13
2.4 ไมโครคอนโทรลเลอร์.....	14
2.4.1 โครงสร้างไมโครคอนโทรลเลอร์.....	15
2.4.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์.....	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

2.4.1.2	โครงสร้างภายนอกของไมโครคอนโทรลเลอร์.....	18
2.4.2	สถาปัตยกรรมของไมโครคอนโทรลเลอร์.....	19
2.4.2.1	สถาปัตยกรรมวอนนิวแมนน์ (Von-Newman Architecture).....	19
2.4.2.2	สถาปัตยกรรมฮาร์วาร์ด (Harvard Architecture).....	19
2.4.2.2.1	Arduino.....	20
2.4.2.2.2	Raspberry Pi.....	23
2.4.3	ไมโครคอนโทรลเลอร์และไมโครโพรเซสเซอร์.....	23
2.4.3.1	หลักการทํางานของไมโครคอนโทรลเลอร์.....	23
2.5	Servo Motor.....	24
2.5.1	บล็อกไดอะแกรมของ Servo Motor.....	25
2.5.2	หลักการทํางานของ RC Servo Motor.....	25
2.5.3	Continuous Rotation Servo.....	27
2.6	มอเตอร์ไฟฟ้ากระแสตรง (DC Motor).....	28
2.6.1	ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง.....	29
2.6.2	หลักการของมอเตอร์ไฟฟ้ากระแสตรง.....	32
2.6.3	ชนิดของมอเตอร์ไฟฟ้ากระแสตรง.....	32
2.7	Accelerometer Sensor.....	34
2.8	H-bridge Circuit.....	35
2.9	Camera WIFI ESR 32 module.....	38
2.10	Li-Ion 18650 battery.....	39
2.10.1	การนำแบตเตอรี่ 18650 ไปใช้งาน.....	32
2.11	18650 Lithium Battery Charging Module.....	40
2.12	Ngrok.....	41
2.13	Motor Drive Module L298N.....	43
2.14	HC-05.....	45
2.15	Step down (Buck) converter.....	46
2.16	Step up (Boost) converter.....	47

## สารบัญ (ต่อ)

บทที่ 3 การออกแบบและวิธีการดำเนินการ.....	48
3.1 การออกแบบ .....	48
3.2 ขั้นตอนการทำงานของระบบการควบคุมรถจำลอง .....	52
3.2.1 บลูทูธ HC-05 (มาสเตอร์-สเลฟ) .....	53
3.2.2 Accelerometer sensor ADXL345 .....	53
3.2.3 Microcontroller Arduino.....	53
3.2.4 L298P DC Motor Drive Shield .....	53
3.2.5 ESP32 Camera.....	53
3.2.6 ตำแหน่งการวางอุปกรณ์บนรถจำลอง .....	54
3.3 การทำงานในส่วนแสดงผล .....	54
3.3.1 Ngrok Server .....	55
บทที่ 4 ผลการทดลอง .....	56
4.1 กล่าวนำ.....	56
4.2 ขั้นตอนการทดลอง .....	56
4.2.1 การควบคุมรถ.....	56
4.2.2 การแสดงผลวีดีโอ .....	58
4.3 ผลการทดลอง.....	60
4.3.1 เปรียบเทียบผลการทดลอง .....	60
4.3.2 การทดลองการตอบสนองการควบคุม .....	63
4.3.3 การทดลองมุมการทำงานของ Servo Motor .....	64
บทที่ 5 สรุปผลการทดลอง.....	66
บรรณานุกรม.....	68
ภาคผนวก ก ไมโครคอนโทรลเลอร์.....	69
ภาคผนวก ข การแสดงผลภาพ .....	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงผลการเปรียบเทียบค่าในทางทฤษฎี ค่าจาก Application Strava และ ค่าจากการวัด ระยะทางจริงที่รถเคลื่อนที่ไปถึง.....	61
4.2 ระยะการตอบสนองการควบคุมการเคลื่อนที่เทียบกับระยะทาง .....	63
4.3 ตารางเปรียบเทียบค่าจาก Servo กับค่ามุมที่วัดได้ ในการเลี้ยวไปทางขวา.....	64
4.4 ตารางเปรียบเทียบค่าจาก Servo กับค่ามุมที่วัดได้ ในการเลี้ยวไปทางซ้าย.....	65
5.1 ค่าองศาในการควบคุมทิศทาง (Roll).....	66
5.2 ค่าองศาในการควบคุมการเคลื่อนที่ (Pitch).....	66



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

รูปที่	หน้า
2.1 บลูทูธ.....	5
2.2 การสื่อสาร มาสเตอร์ กับ สเลฟ.....	10
2.3 สายสัญญาณที่ใช้สำหรับการสื่อสาร มาสเตอร์ - สเลฟ.....	10
2.4 กราฟการส่งสัญญาณการสื่อสารแบบ SPI โดยเริ่มจาก มาสเตอร์ สเลฟ.....	11
2.5 การเชื่อมต่อระบบด้วยการสื่อสารอนุกรมแบบ I <sup>2</sup> C.....	12
2.6 ลักษณะบล็อกการส่งสัญญาณของการสื่อสารแบบ I <sup>2</sup> C.....	13
2.7 โครงสร้างของไมโครคอนโทรลเลอร์.....	15
2.8 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์.....	15
2.9 การติดต่อของไมโครคอนโทรลเลอร์.....	17
2.10 โครงสร้างแบบ SOIC.....	18
2.11 โครงสร้างแบบ PDIP.....	18
2.12 โครงสร้างแบบ PLCC.....	18
2.13 โครงสร้างแบบ TQFP.....	19
2.14 สถาปัตยกรรมวอนนิวแมน.....	19
2.15 สถาปัตยกรรมฮาร์วาร์ด.....	20
2.16 บอร์ดพัฒนาโปรแกรม Arduino.....	21
2.17 โครงสร้าง Hardware ของ Arduino.....	22
2.18 ตัวอย่างการประยุกต์ใช้ Raspberry Pi เป็น Smart TV.....	23
2.19 ส่วนประกอบของเซอร์โวมอเตอร์.....	24
2.20 ส่วนประกอบภายใน RC Servo Motor.....	25
2.21 Block Diagram Servo Motor.....	25
2.22 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลส์.....	26
2.23 RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา.....	27
2.24 การหมุนไปทางซ้ายด้วยความเร็วสูงสุดของ Servo Motor.....	27
2.25 การสั่งให้ Servo Motor หยุดหมุน.....	28
2.26 การหมุนไปทางขวาด้วยความเร็วสูงสุด.....	28
2.27 ส่วนที่อยู่กับที่หรือสเตเตอร์.....	30
2.28 แกนขั้ว.....	30
2.29 ขดลวดสนามแม่เหล็ก.....	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

2.30	โรเตอร์.....	31
2.31	วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม.....	33
2.32	วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน.....	33
2.33	วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบซอร์ทซ์ชั้นท์คอมปาร์ตมอเตอร์.....	34
2.34	วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบลองชั้นท์คอมปาร์ตมอเตอร์.....	34
2.35	โครงสร้างของ Accelerometer.....	35
2.36	การควบคุมมอเตอร์ H-Bridge.....	35
2.37	Schematic ของวงจร H-Bridge.....	36
2.38	การต่อบอร์ด Arduino ควบคุม H-Bridge.....	36
2.39	การต่อไอซี L293D ควบคุมดีซีมอเตอร์.....	37
2.40	H-Bridge Circuit.....	37
2.41	บอร์ด ESP32-CAM.....	38
2.42	แบตเตอรี่เบอร์ 18650 แยกตัวเลขเป็น 2 ส่วน ตามมาตรฐาน.....	39
2.43	18650 Lithium Battery Charging Module.....	41
2.44	การเปิด-ปิดใช้งาน Ngrok.....	42
2.45	อินเตอร์เฟซ Ngrok.....	42
2.46	ชุดขับมอเตอร์ L298N.....	43
2.47	การแสดงตำแหน่งและหน้าที่ของชุดขับมอเตอร์.....	44
2.48	HC-05.....	45
2.49	วงจร Step Down Converter.....	46
2.50	Step Down Converter.....	46
2.51	วงจร Buck boost Converter.....	47
2.52	Step up Converter.....	47
3.1	บล็อกไดอะแกรมแสดงการทำงานเบื้องต้น.....	48
3.2	วิธีการควบคุมการทำงานและการแสดงผล.....	48
3.3	Flow chart การทำงานของระบบในส่วนของมาสเตอร์.....	49
3.4	Flow chart การทำงานของระบบในส่วนของสเลฟ.....	50
3.5	Flow chart การทำงานของระบบในส่วนของกล่องแสดงผล.....	51
3.6	การเชื่อมต่ออุปกรณ์ในส่วนของมาสเตอร์.....	52
3.7	การเชื่อมต่ออุปกรณ์ในส่วนของสเลฟ.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

3.8 การเชื่อมต่ออุปกรณ์ในส่วนของสเลฟ.....	54
3.9 แผนภาพการใช้งานของ NGROK Server และผู้ใช้งาน.....	55
3.10 แผนภาพการเชื่อมต่อของ ESP32 กับ NGROK Server.....	55
4.1 แสดงวิธีการควบคุมทิศทาง.....	56
4.2 แสดงวิธีการควบคุมการเคลื่อนที่.....	57
4.3 Application Strava.....	57
4.4 แสดงค่า ไอพีแอดเดรส ของ module ESP32 CAM.....	58
4.5 เว็บเบราว์เซอร์แสดงภาพวิดีโอแบบ Real-Time.....	59
4.6 ปรับขนาดของวิดีโอ รูปแบบ Stream และแบบ Capture.....	59
4.7 ระยะเวลาการทดลองที่ได้จาก Application strava.....	60
4.8 กราฟแท่งเปรียบเทียบค่าระยะทางที่วัดได้ในแต่ละครั้ง.....	62
4.9 กราฟเปรียบเทียบค่าระยะทางที่วัดได้ในแต่ละครั้ง.....	62
4.10 กราฟแท่งแสดงค่าระยะทางเฉลี่ยที่วัดได้ของระยะจริงกับ Application Strava.....	63
4.11 แสดงการรับ-ส่งค่าระหว่าง มาสเตอร์ และ สเลฟ ในส่วนของค่า Roll และ ค่า Pitch.....	64

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันประเทศไทยและประเทศอื่น ๆ มีการเปลี่ยนแปลงด้านเทคโนโลยีอย่างรวดเร็วรวมทั้งด้านการขนส่งการโดยสารและมีการปรับเปลี่ยนไปในทางด้านยานยนต์ไฟฟ้าเพื่อแก้ปัญหาด้านมลพิษมลภาวะ ลดการใช้ น้ำมัน ซึ่งในภายภาคหน้าจะมีการเปลี่ยนแปลงในด้านอุตสาหกรรมทางด้านยานยนต์อย่างชัดเจนคือการปรับเปลี่ยนการใช้เชื้อเพลิงในการขับเคลื่อนของเครื่องยนต์ จากเชื้อเพลิงที่เป็นน้ำมันหรือการสันดาปภายใน จะมีการเปลี่ยนแปลงเป็นระบบยานยนต์ไฟฟ้าทั้งสิ้น ซึ่งในระบบอุตสาหกรรมในอนาคตที่จะมีบทบาทต่อการแปลงที่สำคัญอย่างหนึ่งก็คืออุตสาหกรรมทางด้านยานยนต์ไฟฟ้า

ทั้งนี้การพัฒนาจะเกิดการเปลี่ยนแปลงของโลกที่เกิดขึ้น ซึ่งในอุตสาหกรรมยานยนต์ไฟฟ้าคงต้องมีการพัฒนาในด้านอื่น ๆ เพิ่มเติม เพื่อเพิ่มประสิทธิภาพในการทำงานและในปัจจุบันมีการเปลี่ยนแปลงอย่างรวดเร็ว นวัตกรรมต่างๆที่มีการใช้ในระบบยานยนต์ไฟฟ้ามีความล้ำสมัยยิ่งขึ้น การควบคุมรถในปัจจุบันที่มีการควบคุมด้วยมนุษย์ภายในห้องโดยสาร จากการใช้งานในปัจจุบัน จึงได้เล็งเห็นแนวทางการควบคุมรถแบบใหม่ ด้วยการจำลอง โดยการใช้เซนเซอร์ ในการควบคุมการทำงานของรถยนต์เพื่ออำนวยความสะดวกและพัฒนาในการใช้งานต่อไป

ระบบจำลองควบคุมรถยนต์โดยใช้เซนเซอร์ความเร่ง โดยมีการใช้ร่วมกับอุปกรณ์ควบคุมขนาดเล็ก ของ บอร์ด Arduino มีการส่งข้อมูลในการควบคุมผ่าน Bluetooth ในการรับ-ส่งสัญญาณในรูปแบบ Master-Slave และมีการติดกล้องเพื่อแสดงจุดตำแหน่งที่รถจำลองได้เคลื่อนที่ไป ระบบนี้จะช่วยเป็นต้นแบบในการพัฒนาเพื่อนำไปใช้ในการควบคุมยานยนต์ไฟฟ้าจริง และจะเป็นประโยชน์ในการควบคุมรูปแบบใหม่ ช่วยอำนวยความสะดวกและได้วิธีการควบคุมยานยนต์แบบใหม่ในการควบคุมยานยนต์ไฟฟ้า

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 เพื่อพัฒนาระบบควบคุมยานยนต์ไฟฟ้าด้วยเซนเซอร์ Accelerometer

1.2.2 เพื่อพัฒนาการใช้งานการโดยการควบคุมผ่านการสื่อสารของ Bluetooth แบบ Master-Slave

1.2.3 เพื่อพัฒนาระบบการควบคุมนี้ไปเป็นต้นแบบในการนำไปใช้งานในอนาคต

1.2.4 เพื่อพัฒนาการแสดงผลในการเคลื่อนที่ของยานยนต์ด้วยกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 ขอบเขตของปริญญานิพนธ์

โครงการนี้เป็นการพัฒนาและประยุกต์ใช้งานในการควบคุมยานยนต์ไฟฟ้าด้วยการจำลอง ผ่านการควบคุมของเซนเซอร์ Accelerometer ในลักษณะการเคลื่อนไหวของเซนเซอร์ในแต่ละตำแหน่ง ซึ่งจำเป็นต้องศึกษาในส่วนของ เซนเซอร์ การสื่อสารของบลูทูธ การแสดงผลของกล้อง แบตเตอรี่ของรถจำลองในการควบคุม โดยการนำการศึกษาทั้งหมดมาใช้ประยุกต์มาเป็นโครงการ ดังกล่าว

1.3.1 เพื่อพัฒนาระบบการควบคุมยานยนต์จำลอง โดยใช้การทำงานของเซนเซอร์ความเร่ง จากค่า โวลต์ และ พีพีซี ในการควบคุมทิศทางต่างๆที่กำหนด

1.3.2 เพื่อพัฒนาระบบควบคุมโดยใช้การสื่อสารแบบบลูทูธ แบบมาสเตอร์กับสเลฟ ในการรับค่าและส่งค่ามาใช้ในการควบคุมการทำงานของยานยนต์จำลอง

1.3.3 เพื่อพัฒนาระบบการแสดงผล โดยแสดงผลผ่านโมดูลกล้อง สำหรับการแสดงการเคลื่อนไหวของยานยนต์จำลองในตำแหน่งและทิศทางต่างๆ

1.3.4 เพื่อพัฒนาระบบจำลองการควบคุมยานยนต์จำลอง สำหรับในการใช้งานในอนาคตเพื่อเป็นต้นแบบสำหรับยานยนต์ไฟฟ้าจริง

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้เข้าใจระบบการทำงานของยานยนต์ไฟฟ้า

1.4.2 ได้ต้นแบบในการควบคุมยานยนต์ไฟฟ้าในอนาคต

1.4.3 สามารถแสดงผลการเคลื่อนที่ของยานยนต์จำลอง ในตำแหน่งและทิศทางต่างๆ

1.4.4 ได้รับความสะดวกในการควบคุมและใช้การสื่อสารควบคุมบลูทูธได้อย่างเหมาะสม

## 1.5 ขั้นตอนการศึกษา

1.5.1 ศึกษาการควบคุมระบบยานยนต์จำลองขนาดเล็ก ชนิดมอเตอร์ ความจุแบตเตอรี่ เพื่อนำมาใช้ได้อย่างเหมาะสม

1.5.2 ศึกษาการทำงานของไมโครคอนโทรลเลอร์ Arduino ชุดคำสั่งต่างๆที่ใช้ในการควบคุมการทำงาน

1.5.3 ศึกษาการใช้งานของเซนเซอร์ Accelerometer ร่วมกับไมโครคอนโทรลเลอร์ Arduino ในการควบคุมและรับค่าพารามิเตอร์ต่างๆ

1.5.4 ศึกษาการสื่อสารของบลูทูธ รวมถึงการสื่อสารการทำงานในรูปแบบมาสเตอร์-สเลฟ ในการรับ-ส่งข้อมูลเพื่อนำไปควบคุม

1.5.5 ศึกษาการแสดงผลของโมดูลกล้อง ในการแสดงผลผ่านระบบไร้สายทางไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.6 ศึกษาการนำค่าที่ได้จากเซนเซอร์ Accelerometer มาทำการปรับให้เหมาะสมกับการเคลื่อนไหวของมอเตอร์ และ การกำหนดทิศทางให้กับเซอร์โวมอเตอร์

1.5.7 ประกอบชิ้นงาน ติดตั้งและทดสอบการทำงานโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและหลักการ

### 2.1 บลูทูธ

#### 2.1.1 วัตถุประสงค์ของเทคโนโลยี บลูทูธ

จุดประสงค์ของเทคโนโลยีบลูทูธนั้น เริ่มต้น เพื่อขายให้แก่บริษัทผู้ผลิต โทรศัพท์เคลื่อนที่ เนื่องจาก เทคโนโลยีนี้ สามารถทำให้ผลิตภัณฑ์ของกลุ่มบริษัทเหล่านี้ใช้งานได้ดีขึ้น โดยการเพิ่มขีดความสามารถของการติดต่อสื่อสารระหว่างโทรศัพท์เคลื่อนที่ กับอุปกรณ์ต่างๆ เนื่องจาก ในอดีต การสื่อสารนี้ ทำได้โดยใช้สายเคเบิล ซึ่งมีความเสี่ยงที่ข้อมูลจะเกิดความเสียหายได้ ดังนั้น จุดประสงค์ของเทคโนโลยีบลูทูธ คือ การแทนที่ การสื่อสารระหว่างอุปกรณ์เคลื่อนที่ ต่างๆ ที่ใช้สายเคเบิล มาเป็นใช้อุปกรณ์ที่ใช้เทคโนโลยีบลูทูธ

บลูทูธ คือมาตรฐานของเทคโนโลยีการติดต่อสื่อสารแบบไร้สาย ที่ใช้คลื่นวิทยุระยะ สั้น (Short-Range Radio Links) ในการติดต่อสื่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์ ในระยะทาง ใกล้ๆ ไม่เกิน 10 เมตร วัตถุประสงค์ของการใช้บลูทูธ คือ เพื่อใช้แทนสายที่ใช้ในการเชื่อมต่อ ทั้งหมดอุปกรณ์บลูทูธ สามารถรับและส่งข้อมูลที่ช่วงคลื่นความถี่ 2.4 GHz และสามารถรองรับ การเชื่อมต่ออุปกรณ์ร่วมกันได้ถึง 8 ชนิดเป็นเครือข่ายที่มีชื่อว่า Pico net ในทางทฤษฎีแล้วเทคโนโลยี นี้จะมีแบนด์วิดท์สูงสุดที่ 1 Mbits / วินาที แต่ถ้าหากเป็นการใช้งาน จริงแล้วแบนด์วิดท์แบบอะซิงโครนัส จะอยู่ที่ 721 Kbits/วินาที และมีความเร็วย้อนกลับที่ 57.6 ส่วนการสื่อสาร แบบซิงโครนัส อยู่ที่ 432.6 Kbits/วินาที เทคโนโลยีนี้เหมาะสำหรับอุปกรณ์แบบพกพาติดตัวได้เนื่องจากใช้ พลังงานต่ำมาก

บลูทูธนี้จะรองรับการรับส่งข้อมูลผ่านทางคลื่นวิทยุ โดยสามารถส่งได้ทั้งข้อมูล ปกติ และ ข้อมูลเสียง ด้วยความเร็ว 1 Mbps ตามมาตรฐาน บลูทูธ 1.x และในอนาคตอันใกล้ ก็ จะขยายขยายไป เป็นBluetooth 2.0 ซึ่งจะให้ความเร็วในการรับส่งที่เพิ่มขึ้นเป็น 10 Mbps และด้วย ความที่ว่าเป็นเทคโนโลยีไร้สายแบบระยะสั้น ซึ่งใช้อุปกรณ์ภาครับ-ส่ง ( Chip transceiver ) ขนาด เล็ก และราคาไม่แพง ทำให้เหมาะกับการใช้งานกับโทรศัพท์มือถือ, เครื่องคอมพิวเตอร์ทั้งแบบพกพา ( Notebook ) และแบบตั้งโต๊ะ ( Desktop )

### 2.1.2 การทำงานของ บลูทูธ

บลูทูธ จะใช้สัญญาณวิทยุความถี่สูง 2.4 GHz. (กิกะเฮิรตซ์) แต่จะแยกย่อยออกไปตามแต่ละประเทศ อย่างในแถบยุโรปและอเมริกา จะใช้ช่วง 2.400 ถึง 2.4835 GHz. แบ่งออกเป็น 79 ช่องสัญญาณ และจะใช้ช่องสัญญาณที่แบ่งนี้ เพื่อส่งข้อมูลสลับช่องไปมา 1,600 ครั้งต่อ 1 วินาที ส่วนที่ญี่ปุ่นจะใช้ความถี่ 2.402 ถึง 2.480 GHz. แบ่งออกเป็น 23 ช่อง ระยะทำการของ บลูทูธ จะอยู่ที่ 5-10 เมตร โดยมีระบบป้องกันโดยใช้การบ่อนรหัสก่อนการเชื่อมต่อ และ ป้องกันการดัก สัญญาณระหว่างสื่อสาร โดยระบบจะสลับช่องสัญญาณไปมา จะมีความสามารถในการเลือกเปลี่ยน ความถี่ที่ใช้ในการติดต่อเองอัตโนมัติ โดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ทำให้การดักฟังหรือ ลักลอบขโมยข้อมูลทำได้ยากขึ้นโดยหลัก ของบลูทูธจะถูกออกแบบมาเพื่อใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากใช้การขนส่งข้อมูลในจำนวนที่ไม่มาก อย่างเช่น ไฟล์ภาพ, เสียง, แอปพลิเคชันต่างๆ และสามารถเคลื่อนย้ายได้ง่าย ขอให้อยู่ในระยะที่กำหนดไว้เท่านั้น (ประมาณ 5-10 เมตร) นอกจากนี้ยังใช้ พลังงานต่ำ กินไฟน้อย และสามารถใช้งานได้นาน โดยไม่ต้องนำไปชาร์จไฟบ่อยครั้ง



รูปที่ 2.1 บลูทูธ

ส่วน ความสามารถการส่งถ่ายข้อมูลของ บลูทูธ จะอยู่ที่ 1 Mbps (1 เมกกะบิต ต่อวินาที) จะไม่มีปัญหาอะไรมากกับขนาดของไฟล์ที่ใช้กันบนโทรศัพท์มือถือ หรือ การใช้งาน แบบทั่วไป ซึ่งถือว่าใช้งานได้ดีมาก แต่ถ้าเป็นข้อมูลที่มีขนาดใหญ่ก็จะช้าเกินไป และถ้าถูกนำไป เปรียบกับ Wireless LAN (WLAN) แล้ว ความสามารถของ บลูทูธ ยังไม่สามารถเทียบเท่าได้ ซึ่งใน ส่วนของ WLAN ก็ยังมีระยะการรับ-ส่งที่ไกลกว่า แต่ขอได้เปรียบของ บลูทูธ จะอยู่ที่ขนาดเล็ก กว่า การติดตั้งทำได้ง่ายกว่า และที่สำคัญ การใช้พลังงานก็น้อยกว่ามาก อยู่ที่ 0.1 วัตต์ หากเทียบกับ คลื่นมือถือแล้ว ยังห่างกันอยู่หลายเท่าเหมือนกัน

รูปแบบของการใช้งาน บลูทูธ แบ่งออกเป็น 3 แบบคือ

1. ใช้ บลูทูธ แทนสายเคเบิลต่างๆ (Cable Replacement)
2. ใช้ บลูทูธ สร้างระบบเครือข่ายขนาดเล็กที่เรียกว่า Pico-Network หรือ PAN (Personal Area Network)
3. ใช้ บลูทูธ เป็นช่องทางในการเข้าถึงข้อมูลในระบบเครือข่ายหลัก (Access Networking)

### 2.1.3 ลักษณะการเชื่อมต่อ บลูทูธ

#### 1. Asynchronous Connectionless (ACL)

ใช้ สำหรับการสื่อสารข้อมูลทั่วไป รองรับการเชื่อมต่อทั้งแบบสมมาตร และไม่สมมาตร Multi-slot packet เมื่อใช้ ACL สามารถมี data rate ได้สูงสุด 723 Kbps ในหนึ่งทิศทาง และ 57.6 kbps ในทิศทางอื่นๆ master จะเป็นผู้ที่ควบคุม bandwidth ที่จะให้ slave ใช้งาน และ ACL ยังสนับสนุน broadcast message ด้วย

#### 2. Synchronous Connection Oriented (SCO)

ใช้ สำหรับการสื่อสารข้อมูลเสียง รองรับการเชื่อมต่อแบบสมมาตร, circuit switch และการเชื่อมต่อแบบจุดต่อจุด ในการเชื่อมต่อแบบสมมาตรมีความเร็วในการรับ/ส่งอยู่ที่ 64 kbps และสามารถเชื่อมต่อได้ 3 ช่องสัญญาณพร้อมกัน แต่โดยมากผู้ผลิต มือถือมักไม่ได้บอกรายละเอียดว่า Chip ของ บลูทูธ ที่ใส่เข้าไปเป็นแบบ ACL หรือ SCO จึงทำให้เกิดปัญหาว่าทำไมมือถือบางรุ่น ถึงมีปัญหาในการเชื่อมต่อกับอุปกรณ์ ต่างๆ แต่ก็มีวิธีแก้ปัญหามาช่วย โดยใช้ Software หรือ driver มาช่วย จึงทำให้แก้ปัญหานั้นได้

Bluetooth ได้มีการแบ่งความแรงของการส่งสัญญาณหรือ Class ด้วยกัน 4 Class คือ

Class 1 : มีกำลังส่งอยู่ที่ 100 มิลลิวัตต์ และมีระยะประมาณ 100 เมตร

Class 2 : มีกำลังส่งอยู่ที่ 2.5 มิลลิวัตต์ และมีระยะประมาณ 10 เมตร

Class 3 : มีกำลังส่งอยู่ที่ 1 มิลลิวัตต์ และมีระยะประมาณ 1 เมตร

Class 4 : มีกำลังส่งอยู่ที่ 0.5 มิลลิวัตต์ และมีระยะประมาณ 0.5 เมตร

## 2.1.4 Versions บลูทูธ

### บลูทูธ 1.0 / บลูทูธ 1.0B

เป็นเวอร์ชันเริ่มต้นที่มีปัญหาตามมามากมายหลังจากที่เปิดให้คนทั่วไปใช้งาน และยังมีตัวส่งสัญญาณ บลูทูธ Hardware Device Address (BD\_ADDR) สำหรับใช้ในกระบวนการเชื่อมต่อ และถือว่าเป็นปัจจัยหลักของความผิดพลาดในการทำงาน

### บลูทูธ 1.1

เป็นเวอร์ชันแรกที่ได้รับการจัดเรตให้อยู่ใน IEEE Standard 802.15.1-2002 ซึ่งเป็นมาตรฐานของ IEEE Standard สำหรับเทคโนโลยีสารสนเทศ โดยในเวอร์ชันนี้ได้รับการแก้ไขข้อผิดพลาดต่างๆมาจาก บลูทูธ 1.0B เรียกร้อย พร้อมทั้งเพิ่มตัวบอกระดับความแรงของสัญญาณที่ได้รับ หรือ Received Signal Strength Indicator (RSSI) ที่ได้รับเข้ามา

### บลูทูธ 1.2

พีเจอร์ที่มีการปรับปรุงและเพิ่มให้กับเวอร์ชันนี้คือ

- มีการเพิ่มความเร็วในการค้นหาและเชื่อมต่อ
- มีการต้านทานการถูกแทรกจากคลื่นสัญญาณอื่นที่แข็งแกร่งขึ้น
- มีความเร็วของการรับส่งสัญญาณเพิ่มขึ้นถึง 721 Kbit/s
- ได้รับการจัดเรตให้อยู่ใน IEEE Standard 802.15.1-2005
- เพิ่มพีเจอร์ Extended Synchronous Connections (eSCO) ทำให้คุณภาพเสียงที่ได้รับดีขึ้น
- เพิ่มระบบปฏิบัติการ Host Controller Interface (HCI) แบบ 3 สาย (UART)
- เริ่มการใช้งานโหมดควบคุมการไหลของข้อมูลและการส่งสัญญาณซ้ำหรือเรียกว่า Flow Control and Retransmission Modes สำหรับ L2CAP

### บลูทูธ 2.0 + EDR

เวอร์ชันนี้ได้มีการเปิดตัวระบบ Enhanced Data Rate (EDR) เพิ่มเพิ่มประสิทธิภาพในการรับ-ส่งข้อมูลที่รวดเร็วยิ่งขึ้น โดยมีความเร็วอยู่ที่ 3 Mbit/s

จากชื่อเวอร์ชันจะมีการเขียน + EDR ต่อท้ายเนื่องจากตัวเทคโนโลยีของ Bluetooth เวอร์ชันนี้ไม่ได้มีการเปลี่ยนแปลงอะไรมากนัก

### บลูทูธ 2.1 + EDR

เป็น บลูทูธ เวอร์ชันแรกที่ได้รับการพัฒนาจาก บลูทูธ SIG เปิดตัวในวันที่ 26 กรกฎาคม ค.ศ.2007 โดยพีเจอร์ที่เสริมเข้ามาคือการจับคู่แบบปลอดภัยหรือ Secure Simple Pairing (SSP) ที่จะช่วยให้การจับคู่ของอุปกรณ์มีความปลอดภัยมากขึ้น นอกจากนี้ยังมีการเพิ่มการขยายสัญญาณร้องขอหรือ Extended Inquiry Response ที่จะช่วยให้เกิดการคัดกรองในการเลือกการเชื่อมต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์อื่น และระบบ Sniff Sunbrating ที่ช่วยให้ลดการใช้พลังงานในโหมดของการประหยัดพลังงาน

### บลูทูธ 3.0 + HS

ฟีเจอร์ใหม่ที่เพิ่มเข้ามาคือ AMP หรือ ALternative MAC / PHY คือการเปิดให้ใช้ MAC และ PHYs เข้ามาช่วยเสริมในเรื่องของการส่งข้อมูลโปรไฟล์บลูทูธ หลักการคือ บลูทูธ จะใช้โมเดมการเชื่อมต่อพลังงานต่ำในขณะที่ระบบว่างอยู่ และจะใช้คลื่นวิทยุที่เร็วขึ้นเมื่อต้องส่งออกข้อมูลจำนวนมาก และมีการพัฒนาให้การรับส่งข้อมูลที่เร็วขึ้นกว่าเดิม

### บลูทูธ 4.0

บลูทูธในเวอร์ชันนี้มีการเปลี่ยนแปลงหลายอย่าง โดยมีการเพิ่มฟีเจอร์ใหม่ๆ ได้แก่ โพรโตคอล Classic Bluetooth, Bluetooth High Speed, และ Bluetooth Low Energy (BLE) โดย Bluetooth high speed หรือบลูทูธความเร็วสูงนั้น มีพื้นฐานมาจาก Wi-Fi, และ Classic Bluetooth ก็ประกอบไปด้วยโพรโตคอล บลูทูธ รุ่นแรกๆที่ได้มีการใช้งาน

โดย บลูทูธ เวอร์ชันนี้จะเน้นไปที่การใช้พลังงานที่ต่ำจนสามารถใช้พลังงานจากแบตเตอรี่แบบเหรียญในการรับ-ส่งข้อมูล อีกทั้งมีการออกแบบชิปให้สามารถใช้งานแบบ Dual Mode และ Single Mode กับอุปกรณ์เวอร์ชันเก่า

#### บลูทูธ 4.1

บลูทูธ SIG ประกาศรองรับเวอร์ชันนี้อย่างเป็นทางการเมื่อวันที่ 4 ธันวาคม ค.ศ. 2013 การอัปเดตในเวอร์ชันนี้ เป็นการอัปเดตซอฟต์แวร์ให้กับบลูทูธเวอร์ชัน 4.0 ที่เพิ่มฟีเจอร์เพื่อให้ผู้ใช้สามารถใช้งานได้หลากหลายยิ่งขึ้น เช่น

- ยกระดับการใช้งานให้ทันสมัยยิ่งขึ้นด้วยเทคโนโลยีมือถืออย่าง LTE
- รักษาระดับการเชื่อมต่อให้มีการแทรกแซงแบบ Manual น้อยลง
- แลกเปลี่ยนข้อมูลได้อย่างมีประสิทธิภาพยิ่งขึ้นด้วย L2CAP
- จำกัดเวลาในการค้นพบอุปกรณ์ใหม่

#### บลูทูธ 4.2

บลูทูธ เวอร์ชันนี้เปิดใช้งานเมื่อวันที่ 2 ธันวาคม ค.ศ.2014 มีการเพิ่มฟีเจอร์สำหรับ Internet of Thing หรือ IoT โดยการพัฒนาหรือฟีเจอร์หลักๆที่เพิ่มมาให้คือ

- การเชื่อมต่อที่ปลอดภัยและใช้พลังงานต่ำกว่าเดิม
- เพิ่มการคัดกรองเพื่อรักษาความเป็นส่วนตัวของชั้นข้อมูลให้มีความปลอดภัยมากขึ้น
- Internet Protocol Support Profile (IPSP) เวอร์ชัน 6 พร้อมใช้งานสำหรับการเชื่อมต่ออุปกรณ์ต่าง ๆ ภายในบ้าน สำหรับการทำระบบ Smart Home

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บลูทูธ 5.0

มีการเปิดใช้งานเมื่อวันที่ 6 ธันวาคม ค.ศ. 2016 โดยจะเพิ่มฟีเจอร์ที่โปกส์ไปในเรื่องของเทคโนโลยี IoT โดยตัว บลูทูธ 5 มีการเชื่อมต่อแบบ BLE หรือการเชื่อมต่อแบบใช้พลังงานต่ำ ซึ่งสามารถเพิ่มความแรงและเร็วในการรับ-ส่งข้อมูลเป็น 2 เท่า แต่ก็มีข้อจำกัดเรื่องระยะของการรับ-ส่งที่สั้นลง

### บลูทูธ 5.1

Bluetooth SIG เปิดตัวเวอร์ชันเมื่อวันที่ 21 มกราคม ค.ศ. 2019 โดยมีฟีเจอร์หลักๆ ที่ได้รับการเพิ่มมาคือ

- Angle of Arrival (AoA) และ Angle of Departure (AoD) ที่เป็นเทคโนโลยีสำหรับการบอกตำแหน่งของอุปกรณ์ที่เปิด บลูทูธ และใช้ในการติดตามอุปกรณ์เวลาสูญหาย เพิ่มดัชนีแบนด์วิดท์ช่วยให้การเชื่อมต่อในสถานที่ที่มีสัญญาณ Bluetooth อยู่เยอะได้ดียิ่งขึ้น

- การเก็บข้อมูล catch ของอุปกรณ์ BLE เรียกว่า GATT (Generic Attribute Profile) Caching

- เพิ่มประสิทธิภาพในการรับ – ส่งข้อมูลและความปลอดภัยให้มากขึ้น
- การซิงค์ข้อมูลในที่มีสัญญาณบลูทูธเป็นจำนวนมากเป็นระยะ ๆ

### บลูทูธ 5.2

เปิดใช้งานในวันที่ 31 ธันวาคม ค.ศ. 2019 โดยมีฟีเจอร์หลักๆคือ

- LE Audio : ทำให้อุปกรณ์เช่นหูฟัง สามารถเชื่อมต่อกับอุปกรณ์อื่นๆพร้อมกันได้หลายตัว หรือเชื่อมต่อหูฟังหลายชิ้นเข้ากับแหล่งกำเนิดเสียงเพียง 1 ชิ้นก็ได้ โดย LE Audio สามารถรับบนคลื่นที่ใช้พลังงานต่ำอย่าง BLE ได้

- LE Power Control : ช่วยให้เราตรวจสอบความแรงของสัญญาณที่เชื่อมต่ออยู่ได้ และช่วยให้รับความแรงของสัญญาณให้เข้ากับอุปกรณ์นั้นๆได้อีกด้วย

- LE Isochronous Channel : ทำให้หูฟังในแต่ละข้างสามารถรับสัญญาณได้เองโดยตรง ช่วยลดการดีเลย์ขณะฟัง

- เพิ่มประสิทธิภาพให้กับข้อมูลของอุปกรณ์ BLE ต่าง ๆ (Enhanced Attribute Protocol = EATT) ทำให้เชื่อมต่อได้เร็วขึ้น และลดการใช้พลังงานลง

## 2.2 การสื่อสารแบบ SPI

SPI ย่อมาจาก Serial Peripheral Interface เป็นการสื่อสารในระยะสั้น ๆ (ระยะทางของสายเชื่อมต่อไม่เกิน 10 ฟุต = ประมาณ 3 เมตร ซึ่งถูกพัฒนาโดย Motorola ในช่วงกลางยุค 80 เป็นการสื่อสารในรูปแบบ Full-Duplex โดยใช้ มาสเตอร์ - สเลฟ โดยมี มาสเตอร์ ได้เพียงตัวเดียวเท่านั้น โดย มาสเตอร์ นั้นจะเป็นตัวหลักในการรับส่งข้อมูลต่าง ๆ ซึ่งสัญญาณที่ถูกส่งจาก มาสเตอร์ นั้นจะวิ่งไปหาอุปกรณ์ สเลฟ ทุกตัว การจะให้ สเลฟ ตัวใดทำงานตามคำสั่งนั้นจะถูกเลือกด้วยสัญญาณของสาย Slave-Select (SS) หรืออีกชื่อคือ Chip-Select (CS)



รูปที่ 2.2 การสื่อสาร มาสเตอร์ กับ สเลฟ

สำหรับสายสัญญาณที่ใช้โดยปกติจะมี 4 เส้น คือ

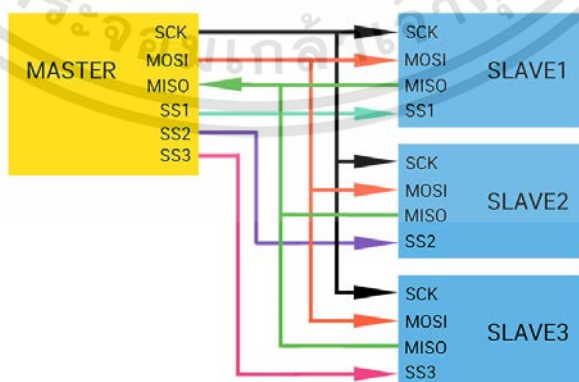
SCK - Serial Clock คือสัญญาณนาฬิกาที่ถูกส่งออกจาก มาสเตอร์ เพื่อเป็นสัญญาณอ้างอิงในการทำงาน

MOSI - Master Out Slave In คือสัญญาณข้อมูลที่ถูกส่งออกจาก มาสเตอร์ ไปเข้ายัง สเลฟ

MISO - Master In Slave Out คือสัญญาณข้อมูลที่ถูกส่งออกจาก สเลฟ ไปยัง มาสเตอร์

SS - Select Slave สัญญาณเลือก สเลฟ (แบบ Active LOW)

หากมีการใช้งาน สเลฟ หลายตัว ใน สเลฟ แต่ละตัวนั้นจะมีสายสัญญาณ SS ประจำตัว และใช้ SCK, MOSI, MISO ร่วมกัน



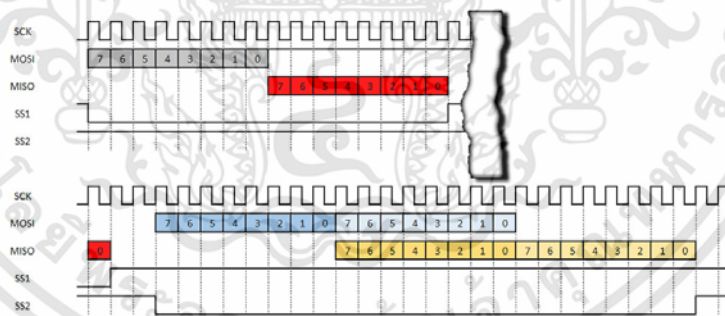
รูปที่ 2.3 สายสัญญาณที่ใช้สำหรับการสื่อสาร มาสเตอร์ - สเลฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1 การทำงาน

การสื่อสารแบบ SPI นั้นจะทำงานโดยต้องมี มาสเตอร์ 1 ตัว และ สเลฟ 1 หรือหลายตัว ตามที่ได้กล่าวไปแล้ว ซึ่งหากในการทำงานนั้นมี สเลฟ เพียงแค่ตัวเดียวก็อาจทำงานโดยให้ SS เป็น LOW ตลอดเพื่อให้ สเลฟ นั้นทำงานตลอดเวลา แต่เมื่อทำงานกับ สเลฟ หลายตัวนั้น SS ของ Slave แต่ละตัวจะถูกตั้งสัญญาณเป็น HIGH และเมื่อจะเรียกใช้ Slave ตัวใดก็จะเปลี่ยนสถานะ SS ของตัวนั้นเป็น LOW เพียงตัวเดียวเท่านั้น เมื่อรับส่งสัญญาณข้อมูลกันเสร็จ SS ก็จะถูกกลับสถานะมาเป็น HIGH ตามเดิม

หลักการส่งข้อมูลของการสื่อสารแบบ SPI เริ่มจาก มาสเตอร์ จะส่งสัญญาณนาฬิกาไป กับสายสัญญาณ SCK (โดยทั่วไปเป็นสัญญาณระดับ MHz) จากนั้นจะเลือกที่ต้องการให้ สเลฟ ตัวใด ทำงานและจะกลับสถานะ SS ของตัวนั้นเป็น LOW จากนั้น มาสเตอร์ จะส่งสัญญาณข้อมูลไปตาม MOSI และรอให้ สเลฟ ส่งสัญญาณข้อมูลกลับมาตาม MISO และเนื่องจาก SPI เป็นการสื่อสารแบบ Full-Duplex ในระหว่างที่ มาสเตอร์ กำลังส่งสัญญาณไปยัง สเลฟ อยู่ นั้น ตัว สเลฟ สามารถส่งข้อมูล กลับไปยัง มาสเตอร์ และ มาสเตอร์ ก็สามารถรับข้อมูลไปพร้อมกับการส่งข้อมูลที่เหลือได้ทันที ไม่ต้อง รอให้ข้อมูล MOSI หมดยกก่อนแล้ว MISO จึงเริ่มส่งได้ หลังจากรับส่งข้อมูลเรียบร้อยแล้ว มาสเตอร์ จึง กลับสถานะให้ SS เป็น HIGH เพื่อไปเรียกใช้ สเลฟ ตัวอื่นตามขั้นตอนที่กล่าวมานี้ไปเรื่อย ๆ ตามคำสั่ง ที่เขียนไว้นั่นเอง



รูปที่ 2.4 กราฟการส่งสัญญาณการสื่อสารแบบ SPI โดยเริ่มจาก มาสเตอร์ สเลฟ

อธิบายจากตัวอย่างกราฟสัญญาณ ด้านบน มาสเตอร์ เปิดใช้ สเลฟ 1 จากนั้นจึงส่ง สัญญาณ 8-bit ไปหา สเลฟ 1 และ สเลฟ 1 ทำการส่งสัญญาณกลับมา 8-bit จึงปิดการใช้งาน สเลฟ 1 ด้วยการกลับ SS1 เป็น HIGH และทำงานต่อมารูปล่าง หลังจากปิด SS1 ผ่านไป 2 ลูกสัญญาณ นาฬิกา มาสเตอร์ ก็เรียกใช้งาน สเลฟ 2 โดยให้ SS2 เป็น LOW จากนั้นจึงส่งสัญญาณไปหา สเลฟ 2 จำนวน 2-byte และ สเลฟ 2 ก็ส่งข้อมูลสวนกลับไปเป็นจำนวน 2-byte จากนั้นจึงหยุดใช้งานโดยการ กลับ SS2 เป็น HIGH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

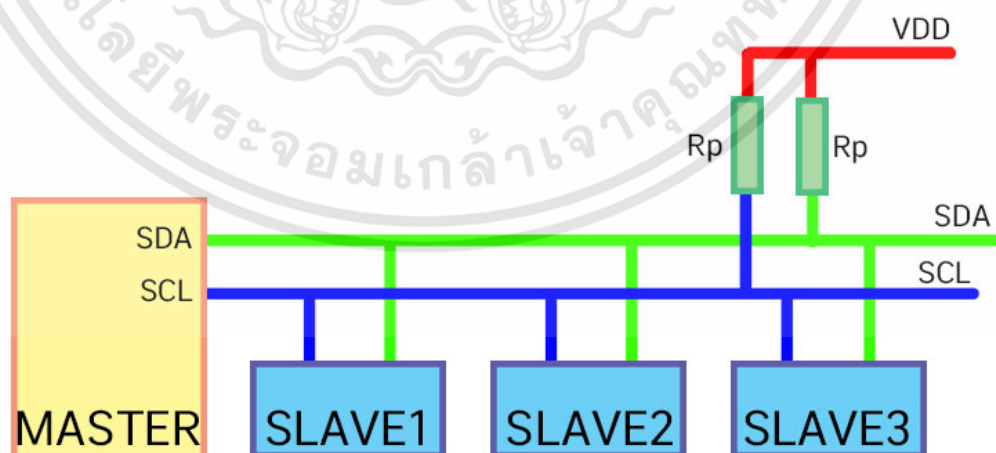
### การนำไปใช้งาน

- เซ็นเซอร์ต่าง ๆ เช่น อุณหภูมิ, ความชื้น, จอสัมผัส
- อุปกรณ์ควบคุม เช่น Digital Potentiometer, DAC
- เลนส์กล้องถ่ายรูป เช่น Canon EF Mount
- หน่วยความจำ flash, EEPROM
- Real Time Clock Module
- จอแสดงผล LCD
- SD Card, MMC Card

### 2.3 การสื่อสารแบบ I2C (I<sup>2</sup>C)

I<sup>2</sup>C ย่อมาจาก Inter-Integrated Circuit คือรูปแบบการสื่อสารข้อมูลอย่างหนึ่งที่สร้างขึ้นมาเพื่อสื่อสารข้อมูลความเร็วต่ำ นิยมใช้กับอุปกรณ์จำพวก ไมโครโปรเซสเซอร์ ไมโครคอนโทรลเลอร์ และอุปกรณ์ต่างๆที่เกี่ยวข้อง I<sup>2</sup>C ถูกคิดค้นขึ้นมาในปี ค.ศ. 1982 โดย Philip semiconductor (ปัจจุบันเปลี่ยนชื่อเป็น NXP semiconductor) ข้อดีของการสื่อสารอนุกรมแบบ I<sup>2</sup>C คือ สามารถรับ-ส่งข้อมูลได้หลายอุปกรณ์ในบัสเดียวกัน ดังรูปที่ 2.5 การเชื่อมต่อระบบด้วยการสื่อสารอนุกรมแบบ I<sup>2</sup>C และใช้สายสัญญาณเพียง 2 เส้นในการรับส่ง-ข้อมูล ทำให้สามารถลดสายสัญญาณที่ใช้ในการเชื่อมต่ออุปกรณ์ลงมาก โดยสายสัญญาณทั้ง 2 เส้นแบ่งเป็น

- SDA (Serial Data) คือ สายสัญญาณสำหรับรับ-ส่งข้อมูล
- SCL (Serial Clock) คือ สายสัญญาณนาฬิกา ใช้เป็นสำหรับควบคุมการรับ-ส่งข้อมูล



รูปที่ 2.5 การเชื่อมต่อระบบด้วยการสื่อสารอนุกรมแบบ I<sup>2</sup>C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1 การใช้งานการสื่อสารอนุกรมแบบ I2C

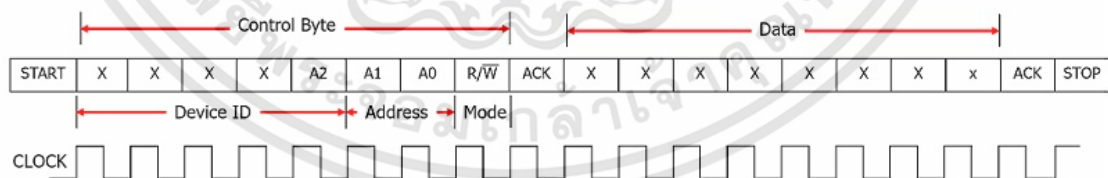
สำหรับการใช้งานการสื่อสารอนุกรมแบบ I2C จะใช้รูปแบบการสื่อสารที่เป็นมาตรฐานของการสื่อสารอนุกรมแบบ I2C มีขั้นตอนการเขียนข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C และการอ่านข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C ดังนี้

#### การเขียนข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C มีขั้นตอนดังนี้

1. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
2. Master Device เขียนไบต์ควบคุม (Control Byte) ไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave ขนาด 7 บิต และคำสั่งเขียนข้อมูล (0) จำนวน 1 บิต)
3. Master Device เขียนข้อมูลไปยัง Slave Device
4. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device

#### การอ่านข้อมูลด้วยการสื่อสารอนุกรมแบบ I<sup>2</sup>C มีขั้นตอนดังนี้

1. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
2. Master Device ส่งไบต์ควบคุม ไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave Device ขนาด 7 บิตและตามด้วยคำสั่งเขียนข้อมูล (0) จำนวน 1 บิต)
3. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device
4. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
5. Master Device ส่งไบต์ควบคุมไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave Device ขนาด 7 บิตและตามด้วยคำสั่งอ่านข้อมูล (1) จำนวน 1 บิต)
6. Master Device อ่านข้อมูลจาก Slave Device
7. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device



รูปที่ 2.6 ลักษณะบล็อกการส่งสัญญาณของการสื่อสารแบบ I<sup>2</sup>C

### 2.3.2 เวิร์กชันการพัฒนาของการสื่อสารอนุกรมแบบ I2C

- เวิร์กชันต้นฉบับเริ่มต้นในปี ค.ศ. 1982 เวิร์กชันนี้ถูกสร้างขึ้นเพื่อเป็นระบบบัสภายในที่ใช้งานง่ายสำหรับควบคุมและสื่อสารอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ความเร็วในการสื่อสารของเวิร์กชันนี้อยู่ที่ 100 kHz

- เวอร์ชัน 1 เริ่มต้นในปี ค.ศ. 1992 เวอร์ชันนี้เพิ่ม Fast mode (FM) ที่มีความเร็วในการสื่อสาร 400 kHz และเพิ่มจำนวน Address ที่รองรับเป็น 10 บิต
- เวอร์ชัน 2 เริ่มต้นในปี ค.ศ. 1998 เวอร์ชันนี้เพิ่ม High speed mode (HS) ที่มีความเร็วในการสื่อสาร 3.4 MHz และลดการใช้พลังงานลง
- เวอร์ชัน 2.1 เริ่มต้นในปี ค.ศ. 2000 เวอร์ชันนี้อธิบายเวอร์ชัน 2 โดยไม่มีการเปลี่ยนแปลงหน้าที่สำคัญ
- เวอร์ชัน 3 เริ่มต้นในปี ค.ศ. 2000 เวอร์ชันนี้เพิ่ม Fast mode plus (Fm+) ที่มีความเร็วในการสื่อสาร 1 MHz และเพิ่มการเปลี่ยนแปลง Address ของอุปกรณ์
- เวอร์ชัน 4 เริ่มต้นในปี ค.ศ. 2012 เวอร์ชันนี้เพิ่ม Ultra Fast mode (UFm) ที่มีความเร็วในการสื่อสาร 5 MHz ใช้ลอจิกแบบ push-pull แทน pull-up resistor และเพิ่ม manufacturer ID
- เวอร์ชัน 5 เริ่มต้นในปี ค.ศ. 2012 เวอร์ชันนี้แก้ไขความผิดพลาดของเวอร์ชันก่อนหน้า
- เวอร์ชัน 6 เริ่มต้นในปี ค.ศ. 2014 เวอร์ชันนี้แก้ไขความผิดพลาดของสองเวอร์ชันก่อนหน้า และใช้เวอร์ชันนี้เป็นมาตรฐาน

#### ข้อดีของ I2C

- ใช้ขาสัญญาณน้อย เพียง 2 เส้น
- ถึงแม้ว่า สเลฟ จะมีความต้องการแตกต่างกัน ในกรณีบางอุปกรณ์เป็น Fm บางอุปกรณ์เป็น Standard ก็สามารถต่อใช้งานร่วมกันได้
- รองรับการใช้งาน มาสเตอร์ ในหลายๆตัว
- มีการใช้ ACK/NACK เพื่อจัดการ Error ต่าง ๆ

#### ข้อเสียของ I2C

- การเขียนโปรแกรมจะมีความซับซ้อน
- ทำงานที่ความเร็วต่ำ
- ทำงานด้วยระบบ Half-Duplex

## 2.4 ไมโครคอนโทรลเลอร์

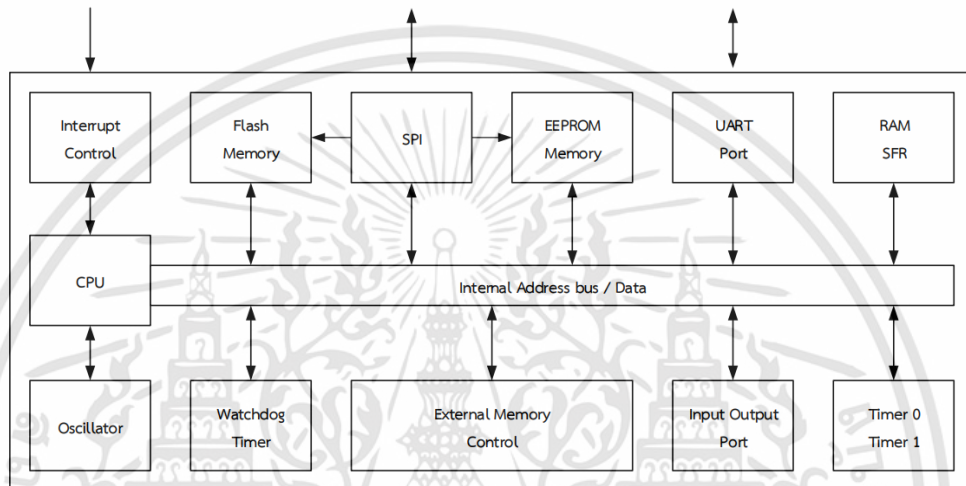
ปัจจุบันไมโครคอนโทรลเลอร์ถูกนำมาใช้อย่างกว้างขวางในงานด้านอิเล็กทรอนิกส์ เพราะใช้งานได้ง่าย สะดวก มีขนาดเล็ก ราคาถูก และที่สำคัญสามารถสั่งงานให้ทำงานตามเงื่อนไขใหม่ตามผู้ใช้งานโดยไม่จำเป็นต้องเปลี่ยนวงจรภายนอก มีบริษัทผู้ผลิตไอซีไมโครคอนโทรลเลอร์ออกมาหลายตระกูล เช่น Z80, PIC, MCS51, AVR, ARM7, ARM9, 68HC แต่ละตระกูลก็จะมีคุณสมบัติต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

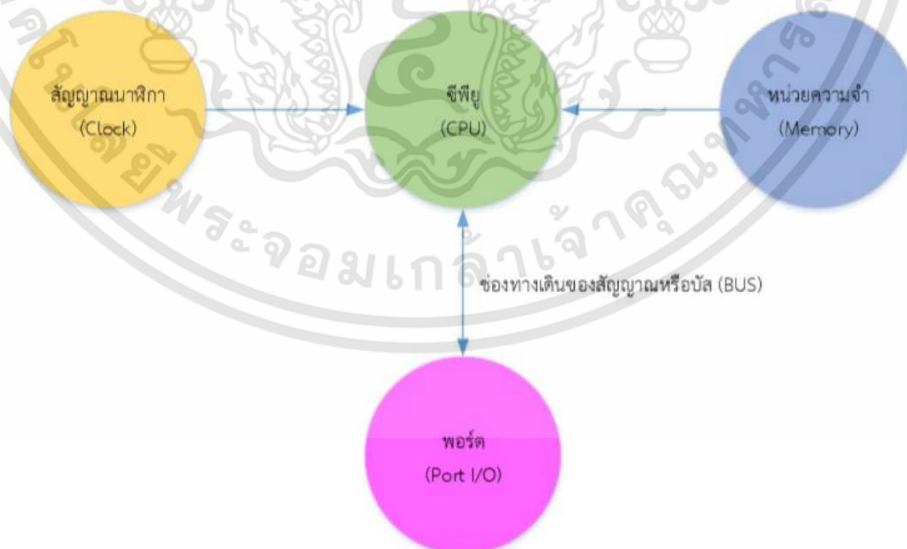
2.4.1 โครงสร้างไมโครคอนโทรลเลอร์

2.4.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์แต่ละเบอร์แต่ละตระกูลนั้นมีโครงสร้างภายในแตกต่างกันขึ้นอยู่กับ บริษัทผู้ผลิตและจุดประสงค์ของการสร้างไมโครคอนโทรลเลอร์เบอร์นั้น ๆ พื้นฐานของโครงสร้าง ภายในของไมโครคอนโทรลเลอร์มีส่วนประกอบดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างของไมโครคอนโทรลเลอร์



รูปที่ 2.8 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่เป็น ศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมด โดยนำข้อมูลจากอุปกรณ์รับข้อมูลมา ทำงาน ประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผลโดยการทำงานของซีพียูมี 2 จังหวะ

1.1 เฟตช์ (Fetch) คือ การอ่านคำสั่งโปรแกรมแล้วทำการถอดรหัสคำสั่งเป็นภาษาเครื่อง

1.2 เอ็กซีคิวต์ (Executed) คือ การทำตามคำสั่งโปรแกรมที่ถอดรหัสแล้ว

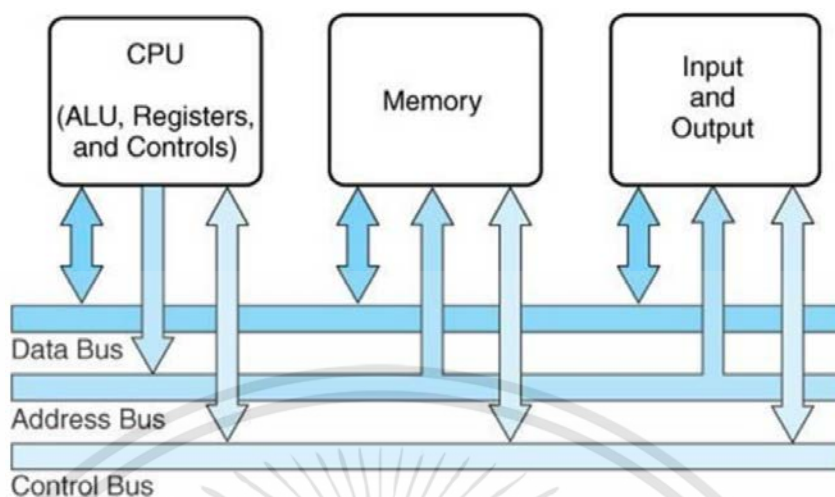
2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ

2.1 หน่วยความจำที่มีไว้สำหรับการเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือน ฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือ

2.2 หน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดาษทดในการคำนวณของ ซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วย ความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM: Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูล ได้แม้ไม่มีไฟเลี้ยง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อ กับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุตเพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่าง ของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูล ระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัว ไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัส ควบคุม (Control Bus)



รูปที่ 2.9 การติดต่อของไมโครคอนโทรลเลอร์

4.1 บัสตำแหน่ง (Address Bus) จะเป็นกลุ่มสายสัญญาณที่ใช้สำหรับกำหนดตำแหน่งของ ข้อมูลในหน่วยความจำ หรือระบุตำแหน่งที่อยู่ของอุปกรณ์ภายนอกต่างๆ โดยบัสตำแหน่งจะเป็น เส้นทางที่ใช้ส่งข้อมูลออกจากไมโครโพรเซสเซอร์เพียงทิศทางเดียว โดยใช้เพื่อระบุตำแหน่งของ อุปกรณ์ที่ต้องการจะติดต่อด้วยเท่านั้น

4.2 บัสข้อมูล (Data Bus) จะเป็นกลุ่มสายสัญญาณที่ใช้ในการรับส่งข้อมูลหรือรหัสคำสั่งต่างๆ ระหว่างไมโครโพรเซสเซอร์กับหน่วยความจำโปรแกรมหน่วยความจำข้อมูล อุปกรณ์อินพุตและ อุปกรณ์เอาต์พุตทุกตัวที่ต่อพ่วงอยู่กับระบบ โดยบัสข้อมูลนี้จะเป็นแบบสองทิศทางสามารถรับและส่ง ข้อมูลทั้งไปและกลับได้ในสายสัญญาณเดียวกัน โดยบัสข้อมูลเปรียบเสมือนช่องทางเดิน รถถ้ามีมาก จะทำให้การติดต่อทำได้รวดเร็วขึ้น เช่น บัสข้อมูลแบบ 8 บิตจะทำงานได้เร็วกว่า บัสข้อมูลแบบ 4 บิต

4.3 บัสควบคุม (Control Bus) จะเป็นกลุ่มสายสัญญาณที่ส่งสัญญาณควบคุมการทำงานต่างๆ ของระบบ โดยไมโครโพรเซสเซอร์สามารถส่งสัญญาณไปควบคุมหน่วยความจำและอุปกรณ์ อินพุต/ เอาต์พุตภายนอกเพื่ออ่านหรือเขียนข้อมูลได้ และอุปกรณ์ภายนอกสามารถส่งสัญญาณมาควบคุมการทำงานของไมโครโพรเซสเซอร์ได้เช่น การรีเซ็ต

4.4. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการ ทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกามี ความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วใน การประมวลผลสูงตามไปด้วย

### 2.4.1.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์

ในปัจจุบันผู้ผลิตไอซีไมโครคอนโทรลเลอร์ได้สร้างโครงสร้างของขาเพื่อต่อใช้งานในหลายลักษณะ เช่นโครงสร้างแบบ SOIC, PDIP, PLCC และ TQFP

1. โครงสร้างแบบ SOIC (Small Outline Integrated Circuit)



รูปที่ 2.10 โครงสร้างแบบ SOIC

2. โครงสร้างแบบ PDIP (Plastic Dual In-line Package)



รูปที่ 2.11 โครงสร้างแบบ PDIP

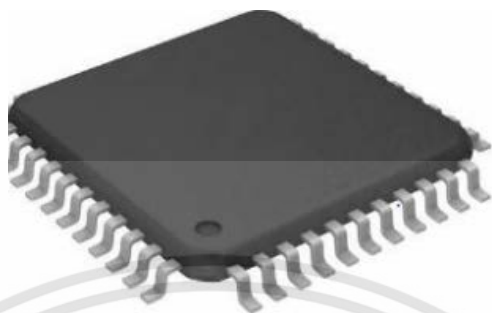
3. โครงสร้างแบบ PLCC (Plastic Leaded Chip Carrier)



รูปที่ 2.12 โครงสร้างแบบ PLCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. โครงสร้างแบบ TQFP (Thin Quad Flat Packages)



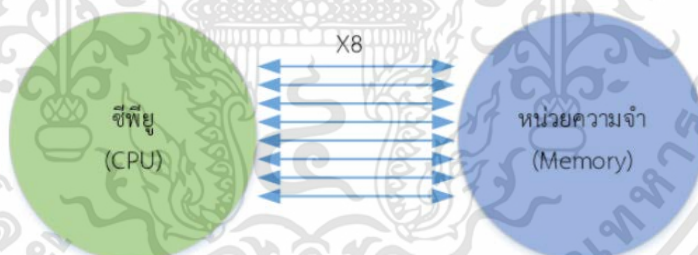
รูปที่ 2.13 โครงสร้างแบบ TQFP

#### 2.4.2 สถาปัตยกรรมของไมโครคอนโทรลเลอร์

สถาปัตยกรรมของไมโครคอนโทรลเลอร์มี 2 แบบ คือ

##### 2.4.2.1 สถาปัตยกรรมวอนนิวแมนน์ (Von-Neuman Architecture)

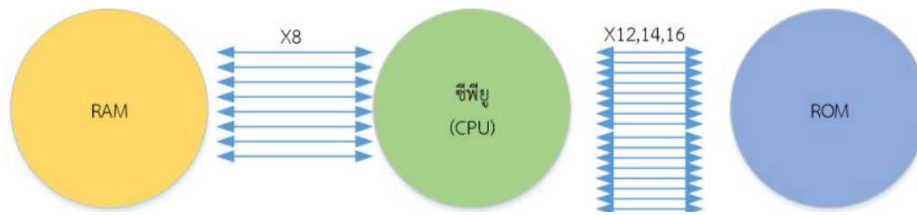
ไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมนี้ ซีพียูจะติดต่อกับหน่วยความจำผ่าน บัสข้อมูลเพียง 8 บิต ทำให้ซีพียูไม่สามารถอ่านหรือเขียนข้อมูลกับหน่วยความจำได้ในเวลาเดียวกัน การทำงานจึงค่อนข้างช้า



รูปที่ 2.14 สถาปัตยกรรมวอนนิวแมนน์

##### 2.4.2.2 สถาปัตยกรรมฮาร์วาร์ด (Harvard Architecture)

ไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมนี้จะมีบัสข้อมูลสองทาง ส่วนแรกคือ ซีพียูจะ ติดต่อกับหน่วยความจำแรมผ่านบัสข้อมูล 8 บิต และส่วนที่สองคือซีพียูจะติดต่อกับหน่วยความจำ รอมผ่านบัสข้อมูล 12, 14, 16 บิต ซีพียูจึงสามารถอ่านและเข้าถึงหน่วยความจำแรมและหน่วยความจำรวมได้ในเวลาเดียวกันทำให้ กระบวนการทำคำสั่งลดขั้นตอนลง ไมโครคอนโทรลเลอร์จึงทำงานได้เร็วขึ้น



รูปที่ 2.15 สถาปัตยกรรมฮาร์ดแวร์

ประเภทของไมโครคอนโทรลเลอร์เมื่อแบ่งสถาปัตยกรรมการประมวลผล

1. ไมโครคอนโทรลเลอร์ 8 บิต
2. ไมโครคอนโทรลเลอร์ 12 บิต
3. ไมโครคอนโทรลเลอร์ 14 บิต
4. ไมโครคอนโทรลเลอร์ 16 บิต
5. ไมโครคอนโทรลเลอร์ 32 บิต

ประเภทของไมโครคอนโทรลเลอร์เมื่อแบ่งสถาปัตยกรรมผู้ผลิต

1. ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip ไมโครชิป)
2. ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
3. ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโครคอนโทรลเลอร์ตระกูล ARM7, ARM9
5. ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
6. ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
7. ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
8. ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
9. ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
10. ไมโครคอนโทรลเลอร์ตระกูล RABBIT
11. ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

#### 2.4.2.2.1 Arduino

Arduino เป็นไมโครคอนโทรลเลอร์บอร์ดแบบสำเร็จรูปในยุคปัจจุบัน ซึ่งถูกสร้างมาจาก Controller ตระกูล AVR ของ ATMEL ข้อดีของไมโครคอนโทรลเลอร์บอร์ดคือเรื่องของ Open Source ที่สามารถนำไป พัฒนาต่อเป็นอุปกรณ์ต่างๆ ได้และความสามารถในการเพิ่ม Boot Loader เข้าไปที่ตัว AVR จึงทำให้การ Upload Code เข้าตัวบอร์ดสามารถทำได้ง่ายขึ้น และยังมีการพัฒนา Software ที่ใช้ในการควบคุมตัวบอร์ด ของ Arduino มีลักษณะเป็นภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C++ ที่โปรแกรมเมอร์มี ความคุ้นเคยในการใช้งาน ตัวบอร์ดสามารถนำโมดูลมาต่อเพิ่ม ซึ่งทาง Arduino เรียกว่าเป็น shield เพื่อเพิ่มความสามารถเพิ่มขึ้น



รูปที่ 2.16 บอร์ดพัฒนาโปรแกรม Arduino

" Arduino เป็นระบบที่ใช้ในการพัฒนาอุปกรณ์อิเล็กทรอนิกส์ต้นแบบ ซึ่งออกแบบให้ใช้งานง่ายทั้ง ฮาร์ดแวร์ และซอฟต์แวร์ กลุ่มเป้าหมายผู้ใช้งานอาद्यโน ได้แก่ ศิลปิน นักประดิษฐ์นักออกแบบ ใช้งานอดิเรก หรือ ใคร ๆ ก็ตามที่สนใจในการประดิษฐ์นวัตกรรมงานสร้างสรรค์"

- ปัจจุบัน Arduino มีบอร์ดหลายแบบให้เลือกใช้งานตามความถนัดและความเหมาะสม มากกว่า 20 รุ่น แต่ละรุ่นก็มีขนาด ความจุ ความเร็ว จำนวนขาพอร์ตอินพุตเอาต์พุต แตกต่างกันไป ออกไป มีตั้งแต่ราคาหลักสองสามร้อยบาท ไปจนกระทั่งพันกว่าบาท นอกจากนี้ยังมีอุปกรณ์ต่อพ่วง (Shield) ให้อีกมากมาย

สาเหตุที่ Arduino Uno เป็นที่นิยมเนื่องจาก

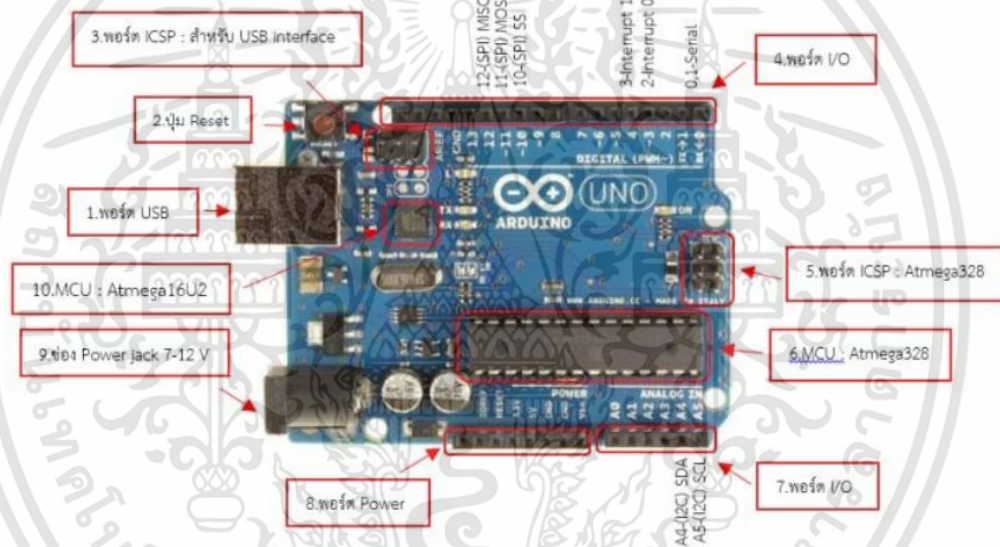
1. ระบบเป็นแบบ Open Source ไม่มีลิขสิทธิ์ในการนำไปใช้งานต่อเชิงพาณิชย์และยังแจก ไฟล์ที่ใช้ในการสร้างต้นแบบให้ฟรีๆ ทำให้ประเทศผู้ผลิตอย่างจีนสามารถนำไปผลิตได้ในราคาถูกๆ โดยไม่ต้องกังวลเรื่องค่าลิขสิทธิ์

2. ซอฟต์แวร์ หรือ Arduino IDE ที่ใช้ในการพัฒนายังให้ใช้งานฟรี ผู้ใช้ดาวน์โหลดได้อย่างถูก กฎหมาย สามารถไปใช้งานต่อ สร้างผลิตภัณฑ์แล้วขายต่อโดยไม่ต้องเสียเงินค่าลิขสิทธิ์ให้แบบ ไมโครคอนโทรลเลอร์เจ้าอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. มีซอฟต์แวร์ (แบบฟรีๆ) ที่สร้างโดยนักพัฒนาใจดีเข้ามารวมอุดมการณ์ (เช่น Fritzing และ Processing) ทำให้เกิดเป็นชุมชนขนาดใหญ่ที่มีเครื่องมือเข้ามาให้ใช้กันฟรีๆ จำนวนมากทั้ง Blog และ website

4. ชุมชนคนใช้ Arduino ในต่างประเทศมีอยู่มากมายหาได้จาก Website ต่างๆ เช่น Arduino.cc, Makezine.com, instructables.com เว็บเหล่านี้แจกแบบร่างและไฟล์ติดตั้ง (Sketch) ให้ฟรีๆ สามารถนำไปสร้างโดยหาวัดดูได้ง่าย โครงสร้าง Hardware ของ Arduino Arduino เปรียบได้กับสมองที่เชื่อมต่อการระหว่างรับรู้กับการแสดงออก ตัวอย่างเช่นดวงตา (INPUT) เห็นภาพและตัวหนังสือ สมอง (Arduino) อ่านและทำให้เกิดการแสดงผลที่ปาก (OUTPUT) ซึ่งออกเสียงตัวหนังสือที่อ่าน



รูปที่ 2.17 โครงสร้าง Hardware ของ Arduino

1. USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
4. I/OPort:Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin 3, 5, 6, 9, 10 และ 11 เป็นขา PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

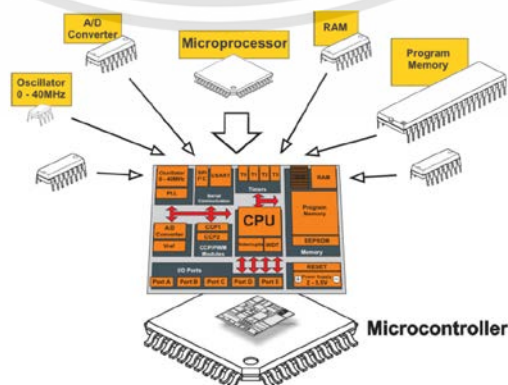
5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader43
6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อกตั้งแต่ขา A0-A5
8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขา ไฟเลี้ยง +3.3 V, +5V, GND, Vin
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

#### 2.4.2.2 Raspberry Pi

Raspberry Pi เป็นไมโครคอนโทรลเลอร์ Board ยุคนี้เหมือนกัน ใช้ Controller ตระกูล ARM เช่นกัน ที่น่าสนใจสำหรับบอร์ด Raspberry Pi ก็คือการจำลองตัวมันเองให้เป็นระบบ คอมพิวเตอร์เครื่องเล็กๆ เครื่องหนึ่ง ที่สามารถรันระบบ Linux ได้ในตัว นั่นก็หมายถึงการดึงระบบ ต่างๆ เพื่อมาใช้งานใน board ทำให้มีความสะดวกมากเพราะมี OS Linux ทำงานให้แทนอยู่แล้ว อย่างเช่นการติดต่อกับระบบ Network การติดต่อกับระบบจอภาพ การติดต่อกับระบบเสียง ตลอดจน การติดต่อกับระบบการเก็บข้อมูลผ่าน SD Card ซึ่งสามารถทำได้ครบและครอบคลุม ด้วยระบบปฏิบัติการ Linux ที่รันอยู่บนตัวบอร์ด Raspberry Pi

### 2.4.3 ไมโครคอนโทรลเลอร์และไมโครโพรเซสเซอร์

#### 2.4.3.1 หลักการทำงานของไมโครคอนโทรลเลอร์



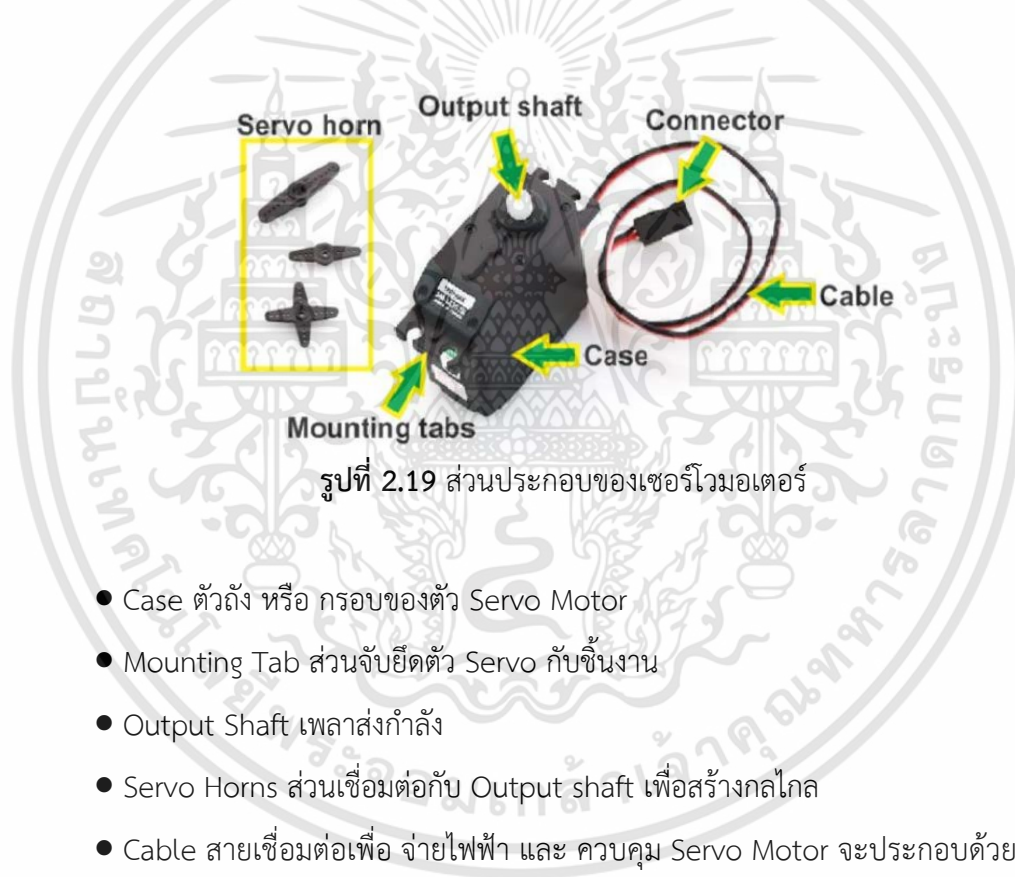
รูปที่ 2.18 ตัวอย่างการประยุกต์ใช้ Raspberry Pi เป็น Smart TV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 Servo motor

ความรู้เบื้องต้นเกี่ยวกับ Servo Motor เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ ในเชิงความหมายของ Servo Motor ก็คือ Motor สามารถสั่งงานหรือตั้งค่า แล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่สั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ในหน่วยนี้จะกล่าวถึง RC Servo Motor ซึ่งนิยมนำมาใช้ในเครื่องบินที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

Feedback Control คือระบบควบคุมที่มีการนำค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต ส่วนประกอบภายนอก RC Servo Motor



รูปที่ 2.19 ส่วนประกอบของเซอร์โวมอเตอร์

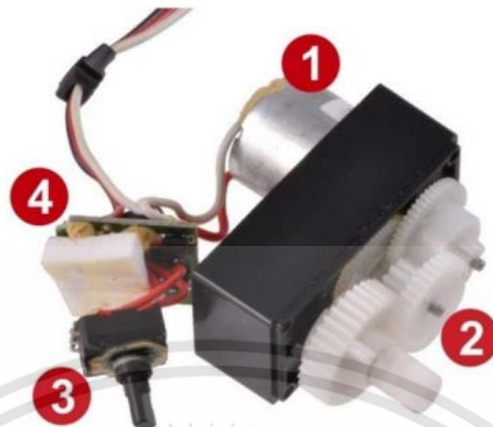
- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เพลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3

เส้น และ

ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้

- สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
- สายสีดำหรือ น้ำตาล คือ กราวด์
- สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 ส่วนประกอบภายใน RC Servo Motor

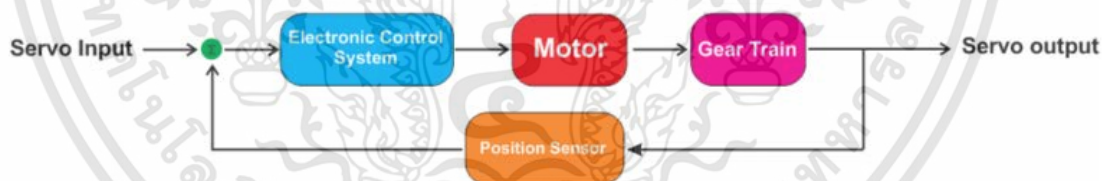
หมายเลข 1 คือ ดีซีมอเตอร์

หมายเลข 2 คือ ชุดเฟืองเกียร์ทดแรงมอเตอร์

หมายเลข 3 คือ ชุดควบคุมและปรับแต่งให้ค่าเอาต์พุตมีค่าเท่ากับ หรือใกล้เคียงกับค่าอินพุต

หมายเลข 4 คือ ส่วนที่ควบคุมและประมวลผล

### 2.5.1 บล็อกไดอะแกรมของ Servo Motor



รูปที่ 2.21 Block Diagram Servo Motor

- Motor เป็นส่วนของตัวมอเตอร์
- Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
- Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาข้อผิดพลาดในการหมุน
- Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

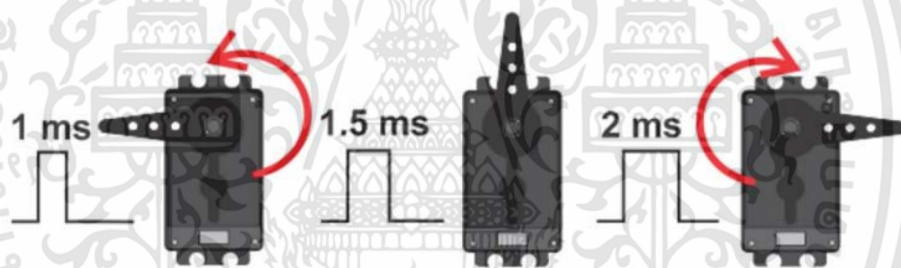
### 2.5.2 หลักการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้ามาเพื่อแปลค่าเป็นตำแหน่ง องศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยัง ตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุนเป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ สัญญาณ RC ในรูปแบบ PWM

ตัว RC Servo Motor ออกแบบมาใช้สำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่นด้วย สัญญาณวิทยุต่างๆ เช่น เครื่องบินบังคับ รถบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ที่ภาครับจะแปลง ความถี่วิทยุออกมาในรูปแบบสัญญาณ PWM (Pulse Width Modulation)

ตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางซ้ายสุด จนสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่ง ขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็ จะหมุนมาอยู่ที่ตำแหน่งตรง กลางพอดี

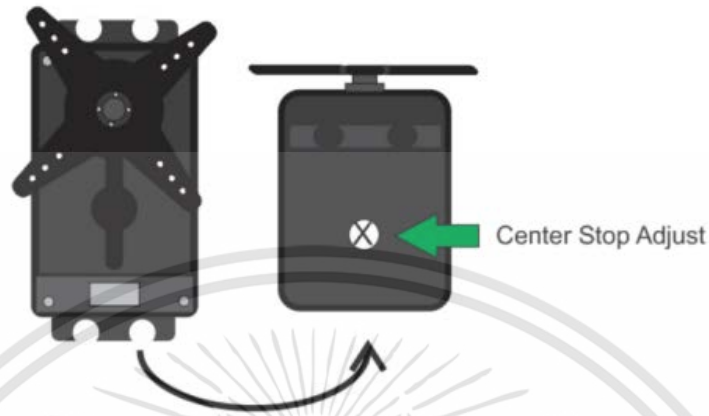


รูปที่ 2.22 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลส์

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่น RC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลส์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน  $(2000-1000)/180$  เท่ากับ 5.55 us

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาค่าพัลส์ที่ต้องการได้จาก  $5.55 \times 45$  เท่ากับ 249.75 us แต่ที่มุม 0 องศาเราเริ่มที่ความกว้างพัลส์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ  $1000 + 249.75$  เท่ากับประมาณ 1250 us

### 2.5.3 Continuous Rotation Servo

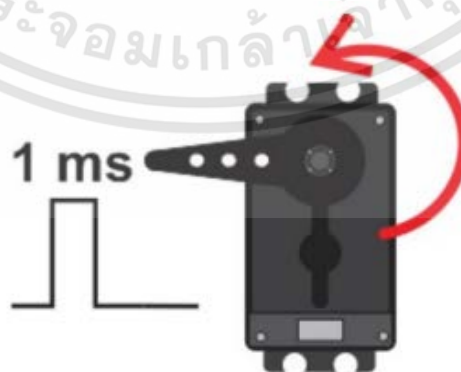


รูปที่ 2.23 RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา

Continuous Rotation Servo คือ RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา ส่วนประกอบ ภายนอกนั้นจะมีหน้าตาคล้ายกับ RC Servo Motor แบบที่หมุนได้ 180 องศา เพียงแต่จะมี Potentiometer เพื่อ ใช้สำหรับปรับ ตำแหน่ง Center Stop Adjust ของตัว Servo

ลักษณะการใช้งาน RC Servo Motor ชนิดนี้จะแตกต่างจากการใช้งาน RC Servo Motor แบบ 180 องศาตรงที่ Servo ชนิดนี้จะใช้ความกว้างของสัญญาณพัลส์ในการกำหนดความเร็ว และทิศทางในการหมุน ไม่ได้ใช้ เพื่อกำหนดมุมจึงไม่สามารถ กำหนดให้ Motor หมุนไปยังตำแหน่งมุม ต่างๆ ตามความต้องการได้ สัญญาณความ กว้างของพัลส์ที่ใช้ควบคุมจะอยู่ในช่วง 1000-2000 us แต่ จะมีความแตกต่างในความหมายของแต่ละความกว้าง ของพัลส์ดังนี้

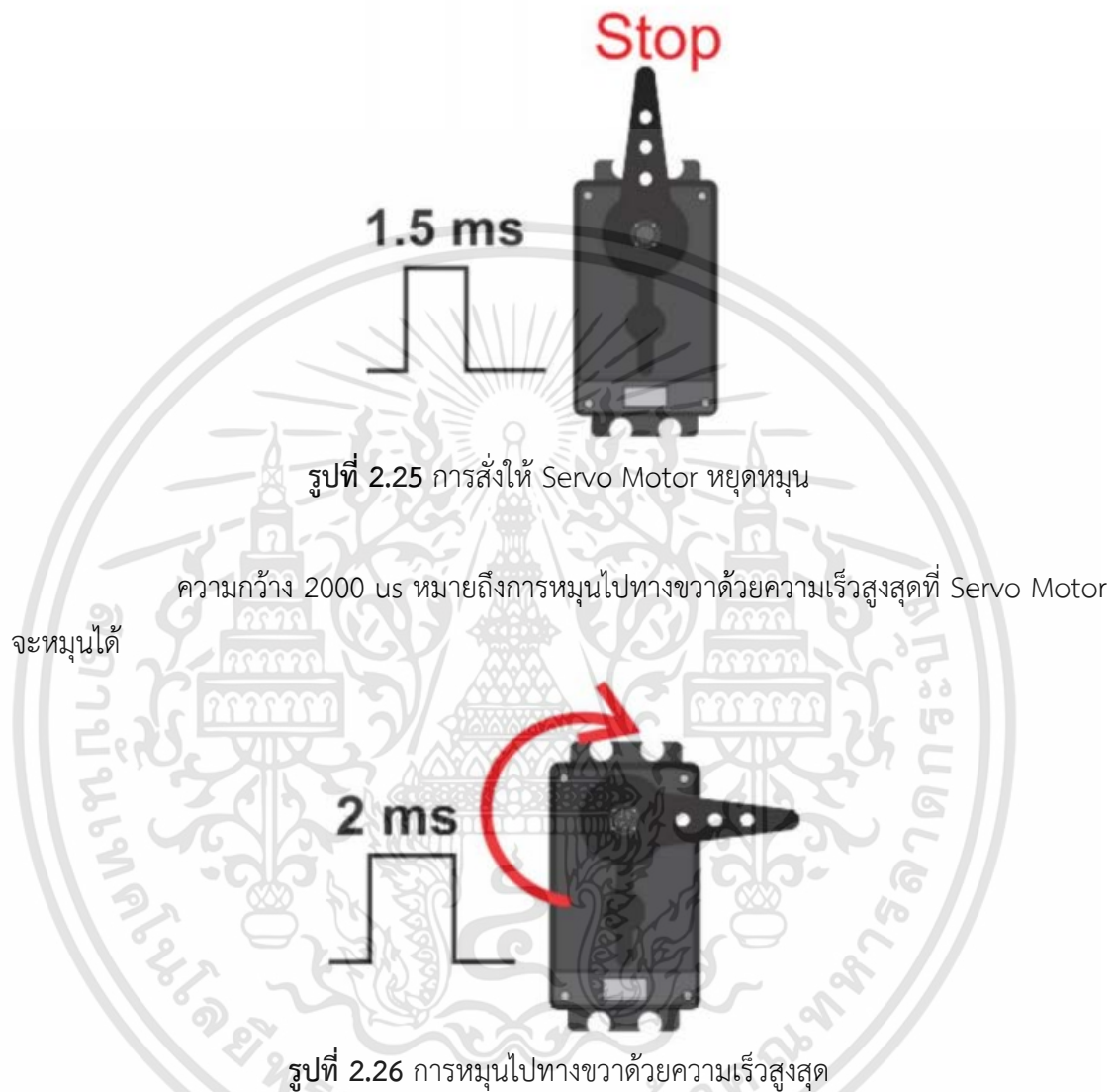
ความกว้าง 1000 us หมายถึงการหมุนไปทางซ้ายด้วยความเร็วสูงสุดที่ Servo Motor จะหมุนได้



รูปที่ 2.24 การหมุนไปทางซ้ายด้วยความเร็วสูงสุดของ Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้าง 1500 us หมายถึงการสั่งให้ Servo Motor หยุดหมุน



## 2.6 มอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

มอเตอร์ไฟฟ้า (Motor) หมายถึงเป็นเครื่องกลไฟฟ้าชนิดหนึ่ง que เปลี่ยนแปลงพลังงานไฟฟ้า มาเป็นพลังงานกล มอเตอร์ไฟฟ้าที่ใช้พลังงานไฟฟ้าเปลี่ยนเป็นพลังงานกลมีทั้งพลังงานไฟฟ้ากระแสสลับ และพลังงานไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้าแบ่งออกตามการใช้ของกระแสไฟฟ้าได้ 2 ชนิดดังนี้

มอเตอร์ไฟฟ้ากระแสสลับ (Alternating Current Motor) การแบ่งชนิดของมอเตอร์ไฟฟ้า สลับแบ่งออกได้ดังนี้

มอเตอร์ไฟฟ้ากระแสสลับแบ่งออกเป็น 3 ชนิดได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. มอเตอร์ไฟฟ้ากระแสสลับชนิด 1 เฟส
  - สปลิตเฟสมอเตอร์ (Split-Phase motor)
  - คาปาซิเตอร์มอเตอร์ (Capacitor motor)
  - รีพัลชันมอเตอร์ (Repulsion-type motor)
  - ยูนิเวอร์แซลมอเตอร์ (Universal motor)
  - เช็ดเดดโพลมอเตอร์ (Shaded-pole motor)
2. มอเตอร์ไฟฟ้ากระแสสลับชนิด 2 เฟส
3. มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟส

มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor) การแบ่งชนิดของมอเตอร์ไฟฟ้ากระแสตรงแบ่งออกได้ดังนี้

มอเตอร์ไฟฟ้ากระแสตรงแบ่งออกเป็น 3 ชนิดได้แก่

1. มอเตอร์แบบอนุกรมหรือเรียกว่าซีรีส์มอเตอร์ (Series Motor)
2. มอเตอร์แบบอนุขนานหรือเรียกว่าชันท่อมอเตอร์ (Shunt Motor)
3. มอเตอร์ไฟฟ้าแบบผสมหรือเรียกว่าคอมปาวด์มอเตอร์ (Compound Motor)

มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงาน อุตสาหกรรม เพราะมีคุณสมบัติที่ดีเด่นในด้านการปรับความเร็วได้ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม เช่น โรงงานทอผ้า โรงงานเส้นใย โพลีเอสเตอร์ โรงงานถลุง โลหะหรือให้ เป็นต้น กำลังในการขับเคลื่อนรถไฟเป็นต้นในการศึกษาเกี่ยวกับมอเตอร์ไฟฟ้า กระแสตรงจึงควรรู้จักอุปกรณ์ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรงและเข้าใจถึงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบต่าง ๆ

### 2.6.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

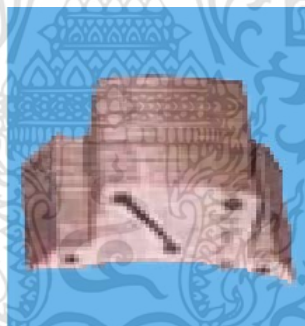
มอเตอร์ไฟฟ้ากระแสตรงที่ส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

1. ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator) ประกอบด้วย
  - เฟรมหรือโยค (Frame Or Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาเป็นรูปทรงกระบอก



รูปที่ 2.27 ส่วนที่อยู่กับที่หรือสเตเตอร์

- ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวด ส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบาง ๆ กั้นด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรมส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่า ขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุด เพื่อให้เกิดช่องอากาศได้น้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กจาก ขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากทำให้ มอเตอร์มีกำลังหมุน



รูปที่ 2.28 แกนขั้ว

ส่วนที่สองขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบ ๆ แกนขั้วแม่เหล็ก ขดลวดนี้ทำหน้าที่รับกระแสจากภายนอก เพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้นและเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น



รูปที่ 2.29 ขดลวดสนามแม่เหล็ก

2. ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกน วางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์



รูปที่ 2.30 โรเตอร์

ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานั้นจะวางอยู่บนแบร์ริง เพื่อบังคับให้หมุนอยู่ในแนวนิ่งไม่มีการสั่นสะเทือนได้
2. แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด (Torque)
3. คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ส่วนหัวซี่ของคอมมิวเตเตอร์จะมีร่องสำหรับใส่ปลายสายของขดลวดอาร์มาเจอร์ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลลาเป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยังขดลวด อาร์มาเจอร์ เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับ เส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็กดังกล่าวมาแล้วเรียกว่าปฏิกริยามอเตอร์ (Motor action)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

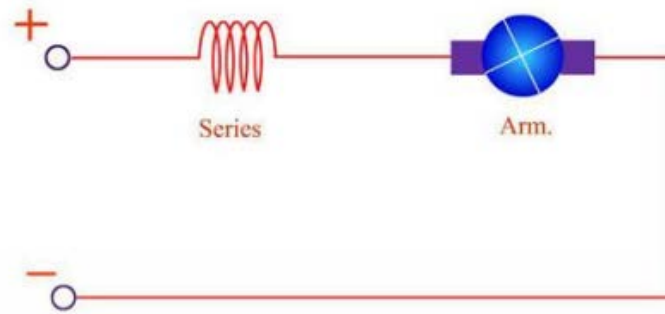
4. ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับ การออกแบบของตัวโรเตอร์ชนิดนั้น ๆ เพื่อที่จะให้เหมาะสมกับงานต่าง ๆ

### 2.6.2 หลักการของมอเตอร์กระแสไฟฟ้าตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อเป็นแรงดันกระแสไฟฟ้าตรง เข้าไปในมอเตอร์ ส่วนหนึ่งจะแปร่งผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้นและกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้นจะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกันและทิศทางเดียวจะเสริมแรงกันทำให้เกิดแรงบิด ในตัวอาร์มาเจอร์ซึ่งวางแกนเพลลาและแกนเพลลานี้สวมอยู่กับตลับลูกปืนของมอเตอร์ทำให้อาร์มาเจอร์นี้ หมุนได้ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้ เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่า ตัวหมุนการที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกริยาต่อกันทำให้ขดลวดอาร์มาเจอร์หรือ โรเตอร์หมุนไปนั้น เป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (Fleming left hand rule)

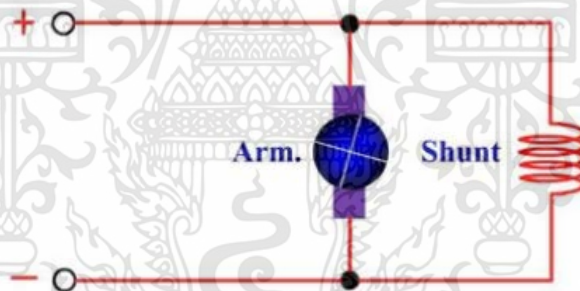
### 2.6.3 ชนิดของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์แบบอนุกรม (Series Motor) คือ มอเตอร์ที่ต่อขดลวดสนามแม่เหล็กอนุกรมกับอาร์มาเจอร์ของมอเตอร์ชนิดนี้ว่า ซีรีส์ฟิลด์ (Series Field) มีคุณลักษณะที่ดีคือให้แรงบิดสูงนิยมใช้เป็นต้นกำลังของรถไฟฟ้ารถยนต์ เครื่องไฟฟ้าความเร็วรอบของมอเตอร์อนุกรมเมื่อไม่มีโหลด ความเร็วจะสูงมากแต่ถ้ามีโหลดมาต่อ ความเร็วก็จะลดลงตามโหลด โหลดมากหรือทำงานหนัก ความเร็วลดลง แต่ขดลวดของมอเตอร์ไม่ เป็นอันตรายจากคุณสมบัตินี้จึงนิยมนำมาใช้กับเครื่องใช้ไฟฟ้าในบ้านหลายอย่าง เช่น เครื่องดูดฝุ่น เครื่องผสมอาหาร ส่วนไฟฟ้า จักรเย็บผ้าเครื่องเป่าผม มอเตอร์กระแสตรงแบบอนุกรมใช้งานหนัก ได้ดีเมื่อใช้งานหนักกระแสจะมาก ความเร็วรอบจะลดลงเมื่อไม่มีโหลดมาต่อความเร็วจะสูงมากอาจ เกิดอันตรายได้ดังนั้น เมื่อเริ่มสตาร์ทมอเตอร์แบบอนุกรมจึงต้องมีโหลดมาต่ออยู่เสมอ



รูปที่ 2.31 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม

มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt Motor) หรือเรียกว่าชันทมอเตอร์ มอเตอร์แบบขนานนี้ขดลวดสนามแม่เหล็กจะต่อ (Field Coil) จะต่อขนานกับขดลวด ชุดอาเมเจอร์ มอเตอร์แบบขนานนี้มีคุณลักษณะมีความเร็วคงที่แรงบิดเริ่ม หมุนต่ำ แต่ความเร็วรอบคงที่ชันท มอเตอร์ส่วนมากเหมาะกับงานที่เป็นพัดลมเพราะพัดลมต้องการ ความเร็วคงที่และต้องการเปลี่ยน ความเร็วได้ง่าย

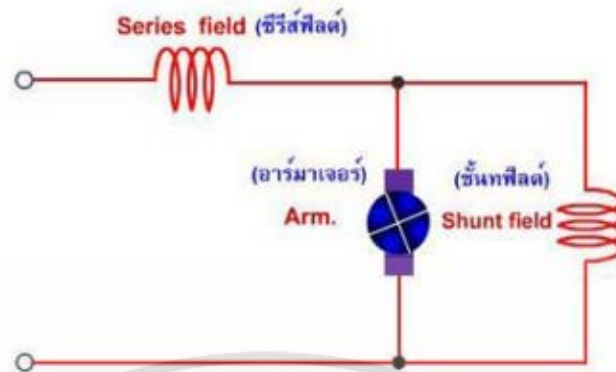


รูปที่ 2.32 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน

มอเตอร์ไฟฟ้ากระแสตรงแบบผสม (Compound Motor) หรือเรียกว่าคอมปาวด์ มอเตอร์ มอเตอร์ไฟฟ้ากระแสตรงแบบผสมนี้จะนำ คุณลักษณะที่ดี ของมอเตอร์ไฟฟ้ากระแสตรง แบบ ขนาน และแบบอนุกรมมารวมกัน มอเตอร์แบบผสมมีคุณลักษณะพิเศษคือมีแรงบิดสูง (High starting torque) แต่ความเร็วรอบคงที่ ตั้งแต่มอเตอร์ไม่มีโหลดจนกระทั่งมีโหลดเต็มที่

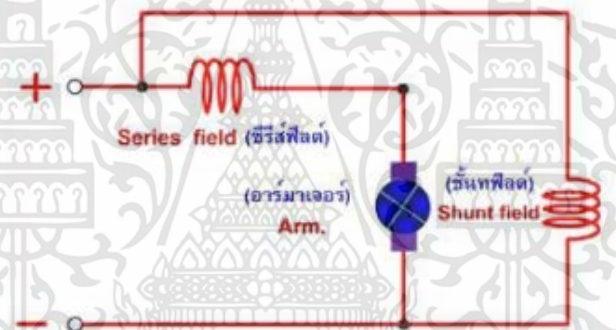
มอเตอร์แบบผสมมีวิธีการต่อขดลวดขนานหรือขดลวดชั้นที่อยู่ 2 วิธี

วิธีที่ 1 ใช้ต่อขดลวดแบบชันทขนานกับอาเมเจอร์เรียกว่า ชอทชันท (Short Shunt Compound Motor) ดังรูปวงจร



รูปที่ 2.33 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบชอร์ตชันทคอมพาร์ตมอเตอร์

วิธีที่ 2 คือ ต่อขดลวด ขนานกับขดลวดอนุกรมและขดลวดอาร์มาเจอร์เรียกว่า ลองชันทคอมพาร์ตมอเตอร์ (Long shunt motor) ดังรูปวงจร



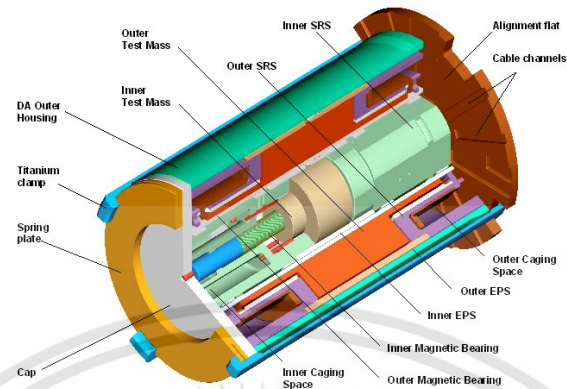
รูปที่ 2.34 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบลองชันทคอมพาร์ตมอเตอร์

## 2.7 Accelerometer Sensor

Accelerometer คือ เครื่องวัดความเร่ง ของการเคลื่อนที่ของวัตถุ พบได้ในสมาร์ทโฟนทั่วไป เช่น iPhone ตัวอย่างการใช้งาน เช่น การเขย่าเพื่อเปลี่ยนเพลง หรือการเขย่าตัวเครื่องเพื่อใช้ในการควบคุมการเล่นเกมนั้น ทั้งหมดเป็นคุณสมบัติของ accelerometer ที่ติดตามในเครื่องแปลตรงตัวคือ Acceleration + Meter หรือมิเตอร์ความเร่ง จากนิยาม หมายถึง Sensor วัดความเร่งเพิ่มขึ้น หรือลดลง (ในหน่วย  $m/s^2$ )

โครงสร้างของ accelerometer จะประกอบด้วยสปริงและลูกตุ้มน้ำหนักเมื่อมีการเคลื่อนที่ด้วยความเร่งลูกตุ้มน้ำหนักจะถูกกดไปอีกฝั่งตรงข้ามกับการเคลื่อนที่ สปริงก็ทำหน้าที่ดึงกลับเข้าที่อีกครั้งเมื่อหยุดการเคลื่อนที่ การเคลื่อนที่ด้วยความเร็วคงที่คือความเร่งเท่ากับศูนย์ ค่าที่วัดได้ก็จะไม่เปลี่ยนแปลง ส่วนตัวเซ็นเซอร์ภายใน ที่จะใช้ในการตรวจวัดความเร่งของลูกตุ้มที่อยู่ในระบบนั้นมีหลายชนิด เช่น เพียโซอิเล็กทริก, สเตรนเกจ, ชนิดใช้แสงตรวจวัด, วัดแรงเฉือน เป็นต้น

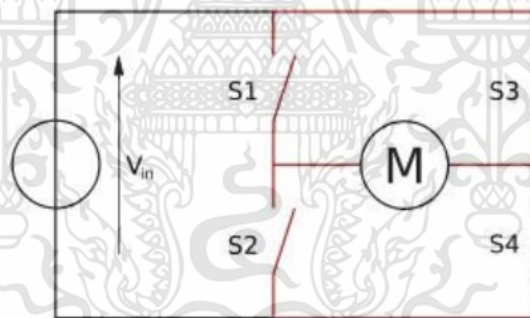
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 โครงสร้างของ Accelerometer

## 2.8 H-bridge Circuit

การควบคุมทิศทาง และความเร็วรอบของมอเตอร์ ไม่ว่าจะเป็นการขับเคลื่อนสายพาน หรือ ล้อของ หุ่นยนต์ไม่ให้หมุนแบบอิสระและสามารถควบคุมได้ ดังนั้นจึงมีวงจรที่ใช้ในการควบคุม มอเตอร์ขึ้นมา แบบที่นิยม ใช้กันเรียกว่าวงจร “H-Bridge”



รูปที่ 2.36 การควบคุมมอเตอร์H-Bridge

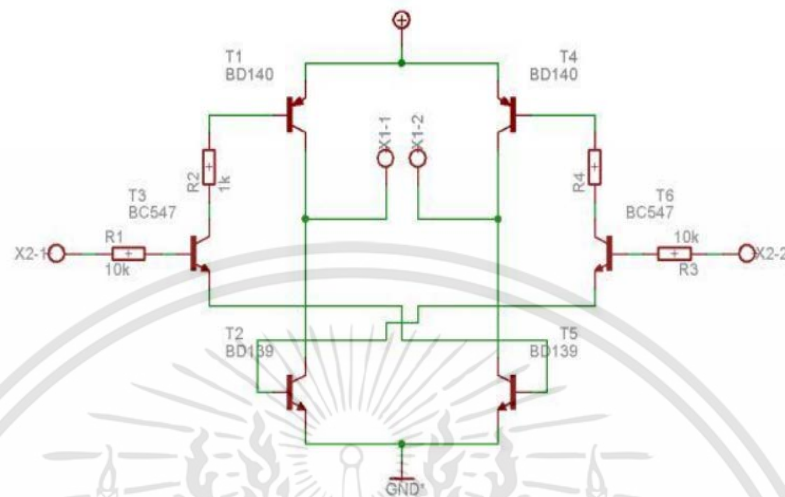
วงจรนี้ทำหน้าที่ได้ทั้งคุมทิศทางและความเร็วของมอเตอร์ โดยเริ่มจากการคุมทิศทาง การหมุน โดยปกติ หากต้องการกลับทิศการหมุนของมอเตอร์กระแสตรง วิธีหนึ่งที่ได้ก็คือกลับทิศแหล่งจ่าย วงจร H-Bridge ด้านบน

หากต้องการให้หมุนตามเข็มนาฬิกา (Clockwise :CW) ให้ S1 และ S4 ปิดวงจร และให้ S2 และ S3 เปิดวงจร หากต้องการให้หมุนทวนเข็มนาฬิกา (Counter Clockwise :CCW) ให้ S2 และ S3 ปิดวงจร และให้ S1 และ S4 เปิดวงจร

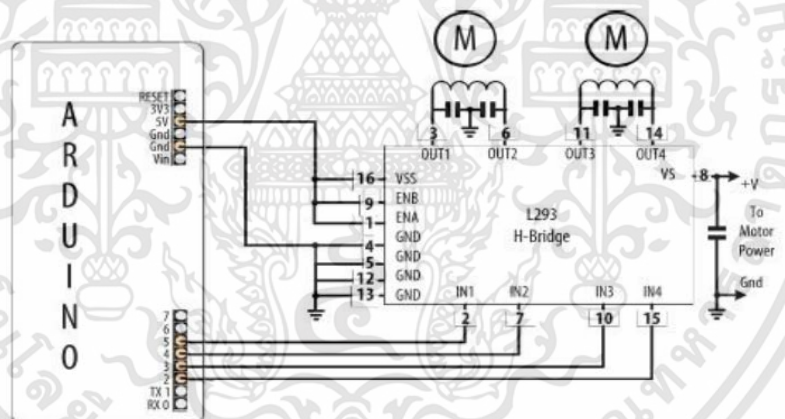
สังเกตว่าสวิตช์จะทำงานเป็นคู่ คู่แรกทำงาน คู่สองต้องเปิดวงจร และในทางตรงข้ามก็คือคู่สองทำงาน คู่แรกต้องเปิดวงจร ต่อมามีการทำให้การเปิดปิดเป็นแบบที่ง่ายกว่าเดิม โดยใช้อุปกรณ์ที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารกึ่งตัวนำเช่น MOSFET หรือ IGBT หรืออื่นๆ แล้วแต่ความเหมาะสม เช่นขนาดกระแสแรงดันที่ต้องการควบคุม

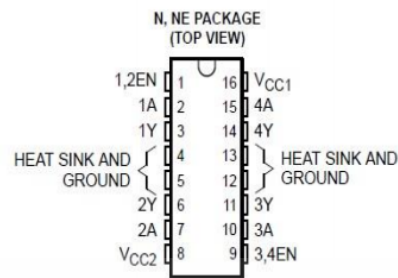


รูปที่ 2.37 Schematic ของวงจร H-Bridge

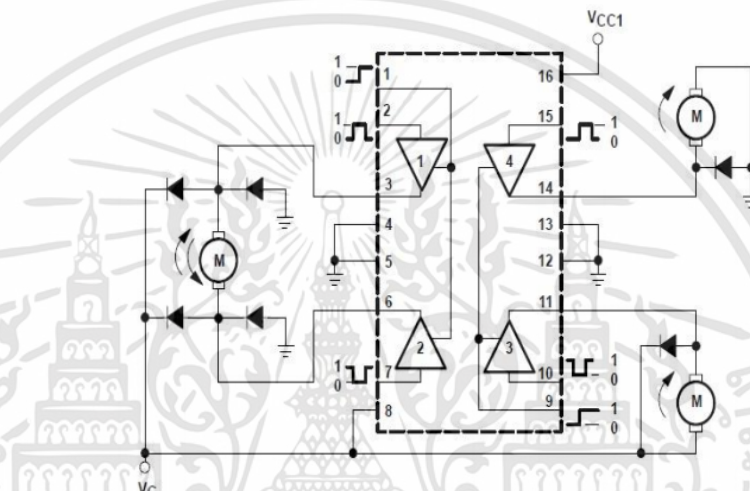


รูปที่ 2.38 การต่อบอร์ด Arduino ควบคุม H-Bridge

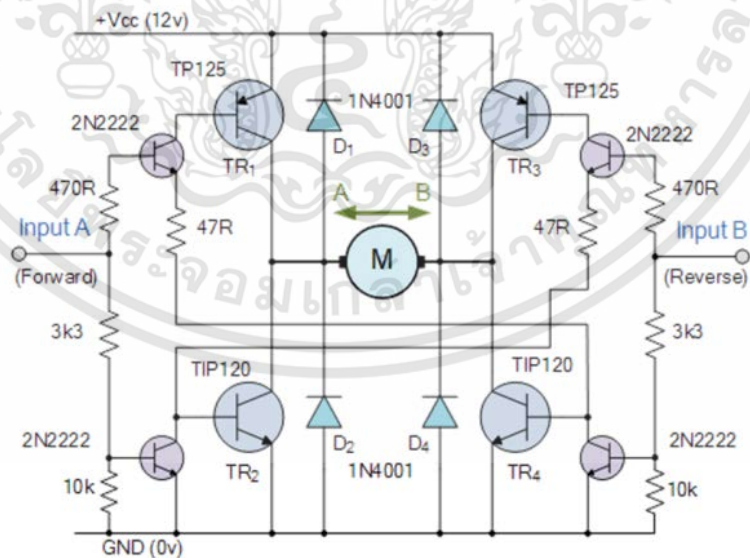
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



block diagram



รูปที่ 2.39 การต่อไอซี L293D ควบคุมดีซีมอเตอร์



รูปที่ 2.40 H-Bridge Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 Camera WIFI ESP 32 module

บอร์ด ESP32-CAM เป็นบอร์ด ESP32 ที่ออกแบบมาพร้อมขนาดเล็กรุ่น OV2640 และ TF card เป็นบอร์ดที่มีราคาถูกเหมาะสำหรับประยุกต์ใช้งานในด้านการส่งภาพไปยัง Server ส่งภาพไปยัง Line สตรีมมิ่ง กล้องวงจรปิดแบบ Online การแจ้งเตือนผ่าน Line Notify เนื่องจากบนบอร์ดไม่มีส่วนแปลงสัญญาณ USB to TTL จึงทำให้มีราคาถูกและมีขนาดเล็ก

- Ultra-small 802.11b/g/n Wi-Fi + BT/BLE SoC module
- Low-power dual-core 32-bit CPU for application processors
- Up to 240MHz, up to 600 DMIPS
- Built-in 520 KB SRAM, external 4M PSRAM
- Supports interfaces such as UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras with built-in flash
- Support for images WiFi upload
- Support TF card
- Support multiple sleep modes
- Embedded Lwip and FreeRTOS
- Support STA/AP/STA+AP working mode
- Support Smart Config/AirKiss One-click distribution network
- Support for serial local upgrade and remote firmware upgrade (FOTA)



รูปที่ 2.41 บอร์ด ESP32-CAM

## 2.10 Li-Ion 18650 battery

ถ่านชาร์จ 18650 เป็นชื่อเรียกถ่านชาร์จชนิด ลิเทียม-ไอออน ( Li-ion ) มีลักษณะเป็นถ่านทรงกระบอก มีเส้นผ่านศูนย์กลาง 18 มิลลิเมตร(mm) มีความสูงหรือความยาวของถ่าน 65 มิลลิเมตร(mm) มีแรงดันไฟ 3.7V มีทั้งแบบหัวนูนและหัวเรียบ ส่วนในเรื่องของความจุแต่ละถ่านนั้นก็ขึ้นอยู่กับว่าในแต่ละยี่ห้อจะผลิตออกมา

ถ่าน Li-ion 18650 นั้นก็เหมือนกับการชาร์จถ่านชนิด Ni-MH คือชาร์จได้ตลอดไม่จำเป็นต้องล้างประจุไฟก่อนทำการชาร์จ และถ้าเป็นไปได้ไม่ควรใช้งานจนไฟหมดแล้วค่อยนำมาชาร์จเพื่อป้องกันการเกิดการคายประจุในอัตราที่สูง ควรจะชาร์จบ่อยๆและถ้าหากอุปกรณ์กินไฟเยอะควรมีการใช้งานในระยะเวลาที่ไม่นาน เพื่อป้องกันไม่ให้ใช้ไฟจนหมดและที่สำคัญความร้อนก็มีผลกับถ่านชนิดนี้เช่นกัน ถ้าหากใช้งานแล้วถ่านเกิดความร้อนมากก็มีผลทำให้ถ่านเสื่อม ส่วนในการชาร์จครั้งแรกควรชาร์จที่ชั่วโมงนั้นขึ้นอยู่กับเครื่องชาร์จว่าเครื่องชาร์จนั้นมีคุณสมบัติในการชาร์จเป็นอย่างไร แต่ถ้าเป็นเครื่องชาร์จที่ตัดอัตโนมัติก็จะดีระดับหนึ่ง เพราะเมื่อมีการชาร์จประจุไฟจนเต็มแล้ว เครื่องชาร์จจะทำการตัดการชาร์จใช้ และที่ดีกว่านั้นคือหากชาร์จประจุไฟจนเต็มแล้วควรที่จะนำถ่านชาร์จออกจากเครื่องชาร์จหรือหากเป็นการชาร์จในตัวอุปกรณ์ที่ใช้งานเองก็ควรจะถอดปลั๊กเมื่อมีการชาร์จประจุไฟจนเต็มแล้ว

ถ่านชาร์จทุกชนิดล้วนมีข้อดีข้อด้อยที่แตกต่างกัน ทั้งนี้ในการเลือกซื้อเลือกใช้ถ่านชาร์จควรเลือกให้เหมาะสมกับการใช้งาน ใช้งานให้ถูกวิธี ดูแลรักษาให้ถูกต้อง ทั้งนี้เพื่อความปลอดภัยและการใช้งานให้เกิดความคุ้มค่า



รูปที่ 2.42 แบตเตอรี่เบอร์ 18650 แยกตัวเลขเป็น 2 ส่วน ตามมาตรฐาน

แบตเตอรี่เบอร์ 18650 แยกตัวเลขเป็น 2 ส่วน ตามมาตรฐาน คือ 18-650

18 คือ เส้นผ่าศูนย์กลางแบตเตอรี่ == 18 มิลลิเมตร (mm)

650 คือ ความยาวแบตเตอรี่ == 65.0 มิลลิเมตร (mm)

### 2.10.1 การนำแบตเตอรี่ 18650 ไปใช้งาน มีการใช้งานแตกต่างกันดังนี้

หัวแบน (Flat), หัวนูน (Tip), หัวนูนมีวงจรป้องกัน (Tip with Protected Circuit)

1. แบตเตอรี่ หัวแบน (ยาว 65.0 mm)

เหมาะสำหรับ ไฟฉายแรงสูง, พัดลมพกพา, บุหรี่ไฟฟ้า, สว่านไฟฟ้า, นำไปบัดกรี, ใส่ใน PowerBank, ใส่ Notebook รุ่นเก่า, ทำ Spot Welding เชื่อมต่อกันเป็นวงจรขนาน หรือวงจรอนุกรม และ อื่นๆ

2. แบตเตอรี่ หัวนูน (ยาว 67.0 mm)

เหมาะสำหรับ ไฟฉายแรงสูง, พัดลมพกพา (พัดลมบางรุ่นใส่แล้วแน่น แนะนำให้ใช้แบตเตอรี่แบนแทน), บุหรี่ไฟฟ้า, สปอตไลท์, ลำโพง bluetooth และอื่นๆ

3. แบตเตอรี่ หัวนูนมีวงจรป้องกัน (ยาว 69.5 mm)

เหมาะสำหรับ ไฟฉายแรงสูง และอุปกรณ์ไฟฟ้าที่มีสปริงยึดหุดได้สูง เพราะวงจรป้องกันจะยาวกว่ามาตรฐานเดิมคือ 65 mm อีก 4.5 mm รวมเป็น 69.5 mm ต้องวัดเครื่องใช้ให้ดีว่าใส่ได้หรือเปล่า

สำหรับวงจรป้องกันที่เพิ่มเข้ามา ใช้สำหรับรักษาแรงดัน หรือ Voltage (V) ของแบตเตอรี่ไม่ให้ใช้หมด จนไม่มีไฟเหลือในแบตเตอรี่ซึ่งเป็นสาเหตุของแบตเตอรี่เสื่อมเซลล์แบตเตอรี่ตาย จนเป็นสาเหตุให้แบตเตอรี่ไม่เข้า

ซึ่งจะช่วยป้องกันไม่ให้แบตเตอรี่ต่ำกว่า 2.75v ซึ่งเป็นแรงดัน Cutoff Voltage ของแบตเตอรี่ Li-Ion 3.7v เบอร์ 18650 หรือ 14500

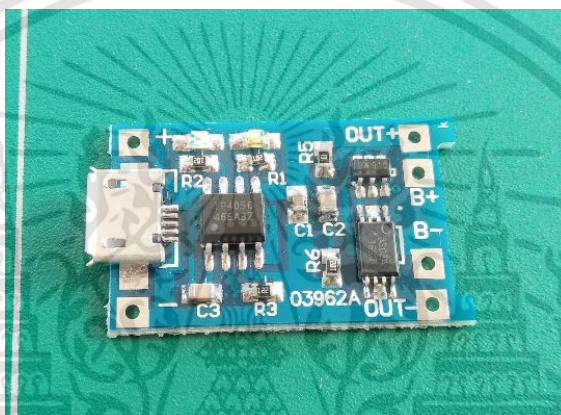
ข้อควรระวัง สำหรับแบตเตอรี่หัวนูนมีวงจรป้องกัน ที่ชาร์จแบตเตอรี่บางรุ่นอาจใส่ไม่เข้าหรือไม่ก็แน่นมากเวลาชาร์จ สำหรับเครื่องชาร์จที่เหมาะสมกับการชาร์จแบตเตอรี่หัวนูนมีวงจรป้องกัน คือ NITECORE I4, D2, D4 เป็นต้น

## 2.11 18650 Lithium Battery Charging Module

18650 Lithium Battery Charging Module + Protection โมดูลชาร์จแบตเตอรี่ลิเทียม 18650 พร้อมวงจรป้องกัน สามารถนำไปใช้ชาร์จแบตเตอรี่ลิเทียมที่มีแรงดัน 3.7V ได้ทุกขนาด โมดูลมีวงจรตัดไฟเมื่อชาร์จเต็ม และวงจรตัดไฟเมื่อแบตเตอรี่เหลือน้อย ป้องกันแบตเตอรี่พังเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้กับแบตเตอรี่ลิเธียมแรงดัน 3.7V
- จ่ายกระแสในการชาร์จ 1A
- มีวงจรป้องกันการจ่ายกระแสเกิน 3A
- ตัดการจ่ายพลังงานไฟฟ้าเมื่อแรงดันของแบตเตอรี่ลดต่ำกว่า 2.5V
- หยุดการชาร์จเมื่อแรงดันเต็อยู่ที่ 4.2V
- ใช้แรงดันอินพุต 5V เลือกว่าจ่ายผ่านช่อง MicroUSB หรือบัดกรีต่อเพิ่ม
- โมดูลมีขนาด 2.6 \* 1.7 เซนติเมตร



รูปที่ 2.43 18650 Lithium Battery Charging Module

## 2.12 Ngrok

Ngrok เป็น Tool Open Source พัฒนาโดย GitHub ซึ่งอำนวยความสะดวกให้บุคคลอื่นสามารถเข้าใช้งาน Website หรือ Application ที่กำลังทำงานอยู่บนเครื่อง Localhost นั้นเอง อีกทั้ง Ngrok ยังมีหลากหลายช่องสัญญาณที่มีความปลอดภัยในการรับและส่งข้อมูลจากเครื่องผู้ใช้ไปจนถึงเครื่อง Localhost (รูปภาพที่ 1 สถาปัตยกรรม Ngrok)

โดยบุคคลอื่นสามารถเข้าใช้งาน Website หรือ Application กำลังทำงานอยู่บนเครื่อง Localhost ผ่านทาง URL ของทาง Ngrok โดยที่ทาง Ngrok จะทำการสุ่มสร้าง URL ขึ้นมา และ URL ที่ได้มานั้น จะทำการเปลี่ยนไปทุกครั้งเมื่อมีการปิดหรือเปิดใช้งาน Ngrok

```

Administrator: C:\Users\lenovo\Downloads\ngrok.exe - ngrok http 80
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             ██████████
Version             2.3.29
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           ██████████ -> http://localhost:80
                   ██████████ -> http://localhost:80

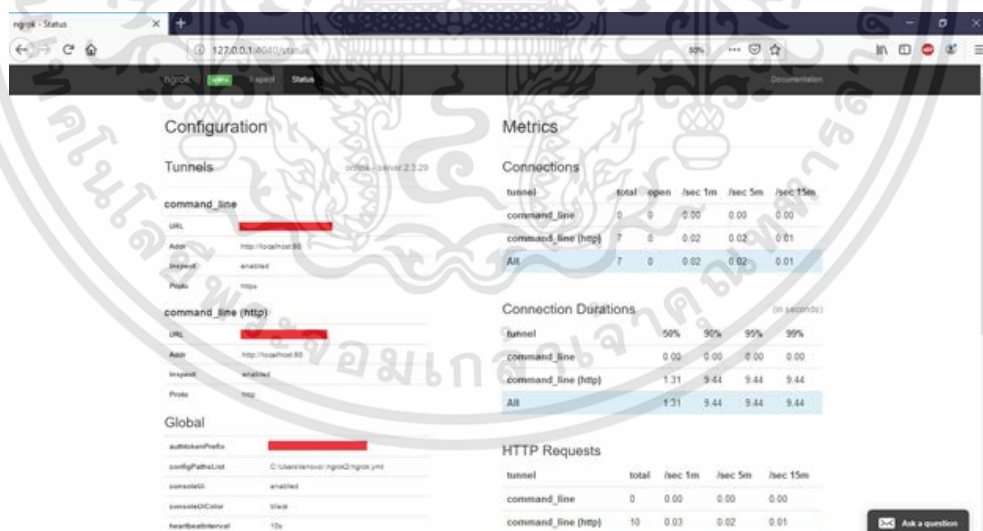
Connections
  ttl    opn    rt1    rt5    p50    p90
   7     0    0.08  0.02  1.31  9.44

HTTP Requests
GET /favicon.ico 200 OK
GET /favicon.ico 200 OK
GET /favicon.ico 200 OK
GET /favicon.ico 200 OK
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway
GET /favicon.ico 502 Bad Gateway

```

รูปที่ 2.44 การเปิด-ปิดใช้งาน Ngrok

อีกทั้ง Ngrok ยังมีเว็บอินเตอร์เฟซ ซึ่งสามารถใช้ตรวจสอบหรือ Monitor ผ่านทาง URL <http://127.0.0.1:4040> ซึ่งสามารถใช้ตรวจสอบการรับส่งข้อมูล Http ทั้งหมด ไม่ว่าจะเป็นการรับส่งข้อมูลแบบ Get หรือแบบ Post, การขอ Request การส่ง Response, Traffic รวมถึง Webhook ที่เข้ามาใช้งานบนเครื่อง Localhost



รูปที่ 2.45 อินเตอร์เฟซ Ngrok

Ngrok ยังรองรับ OS ที่หลากหลายดังนี้

1. Mac OS X
2. Linux

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Mac (32-bit)
4. Windows (32-bit)
5. Linux (ARM)
6. Linux (ARM64)
7. Linux (32-bit)
8. FreeBSD (64-Bit)
9. FreeBSD (32-bit)

### 2.13 Motor Drive Module L298N



รูปที่ 2.46 ชุดขับมอเตอร์ L298N

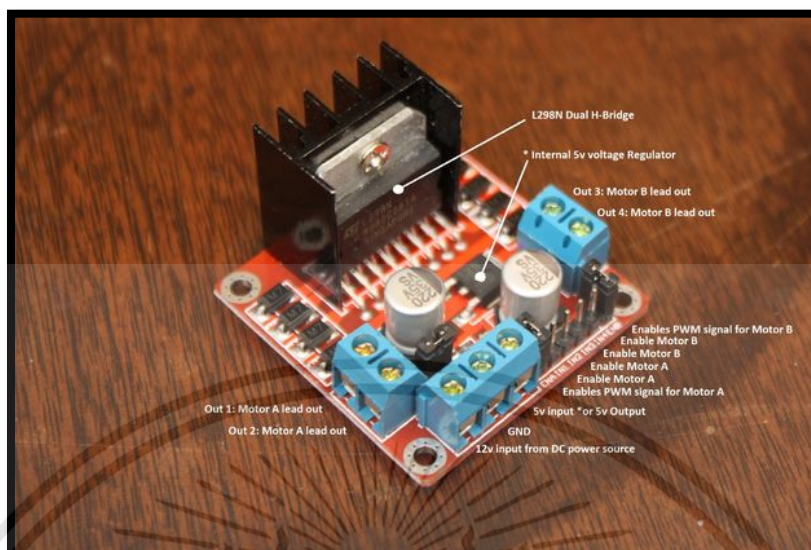
สำหรับ หุ่นยนต์ (Robot) แล้วสิ่งที่เป็นพลังขับเคลื่อนหลักให้กับหุ่นยนต์นั้นก็คงจะไม่พ้น มอเตอร์ ซึ่งต้องการการควบคุม จาก ชุดไดร์วมอเตอร์ (Motor Driver)ที่จะมาควบคุม ทั้ง ทิศทาง และ ความเร็ว ของมอเตอร์ของเรา ซึ่งในวันนี้ จะมาพูดถึง "L298N Dual H-Bridge Motor Controller"

**การใช้งาน** L298N เป็นชุดขับมอเตอร์ชนิด H-Bridge ซึ่งส่วนใหญ่จะถูกนำไปใช้ในการควบคุม ทิศทาง และความเร็วของมอเตอร์ ซึ่งสามารถควบคุมมอเตอร์ได้ทั้งหมด 2 Channel

**หลักการทำงาน** วงจร H-Bridge ของ L298N จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วย ลอจิกเพื่อควบคุมทิศทาง

ส่วนความเร็วของมอเตอร์ นั้นจะถูกควบคุมด้วย สัญญาณ (PWM Pulse Width Modulation) ซึ่งต้องมีการปรับความถี่ให้เหมาะสมกับมอเตอร์ที่จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.47 การแสดงตำแหน่งและหน้าที่ของชุดขับมอเตอร์

**ขาต่างๆ:**

Out 1: ช่องต่อขั้วไฟของมอเตอร์ A

Out 2: ช่องต่อขั้วไฟของมอเตอร์ A

Out 3: ช่องต่อขั้วไฟของมอเตอร์ B

Out 4: ช่องต่อขั้วไฟของมอเตอร์ B

12V: ช่องจ่ายไฟเลี้ยงมอเตอร์ 12V (ต่อได้ตั้งแต่ 5V ถึง 35V)

GND: ช่องต่อไฟลบ (Ground)

5V: ช่องจ่ายไฟเลี้ยงมอเตอร์ 5V (หากมีการต่อไฟเลี้ยงที่ช่อง 12V แล้วช่องนี้จะทำหน้าที่จ่ายไฟออก เป็น 5V Output สามารถต่อไฟจากช่องนี้ไปเลี้ยงบอร์ด Arduino ได้

ENA: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ A

IN1: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A

IN2: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A

IN3: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B

IN4: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B

ENB: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.14 HC-05

โมดูล HC-05 (Master/Slave) เป็นอุปกรณ์ที่มีราคาถูกและใช้สำหรับรับส่งข้อมูลอนุกรม (Serial) ไร้สายผ่าน บลูทูธ (ความถี่ประมาณ 2.4GHz) และทำงานแบบ บลูทูธ SPP (Serial Port Protocol) (เป็นอุปกรณ์ที่เรียกว่า Bluetooth-to-Serial) เหมาะสำหรับการรับส่งข้อมูลในระยะใกล้ๆ บทความนี้กล่าวถึง การใช้โมดูล HC-05 เป็นอุปกรณ์ในการอัปโหลด (upload) โปรแกรม Arduino Sketch ไปยังบอร์ด Arduino Pro Mini (5V/16MHz) ในกรณีที่ไม่มีใช้อุปกรณ์ USB-to-Serial อุปกรณ์ที่ใช้ในการทดลอง

- โมดูล HC-05 (Bluetooth-to-Serial, Master/Slave)
- โมดูล USB-to-Serial ที่มีขาไฟเลี้ยง VCC=+5V, GND และขาสัญญาณ Tx, Rx (มีระดับลอจิก 3.3V) -- ในตัวอย่างนี้ได้เลือกใช้สาย USB-to-UART cable PL-2303HX
- บอร์ด Arduino Pro Mini (ATmega328P, 5V/16MHz)



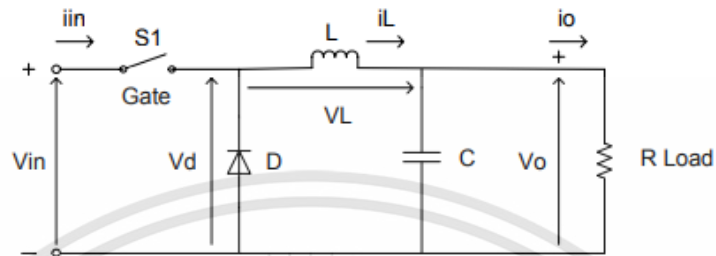
รูปที่ 2.48 HC-05

## 2.15 Step down (Buck) converter

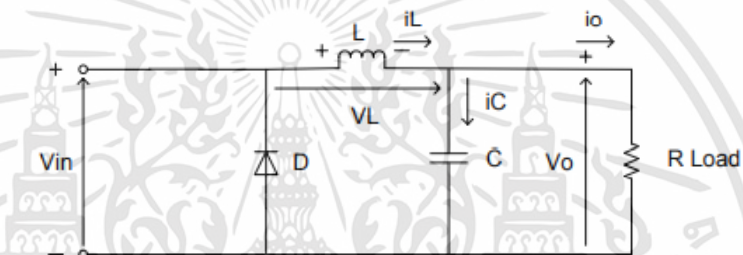
วงจรไฟฟ้า Step down converter หรือเรียกอีกอย่างหนึ่งว่า Buck converter เป็นวงจรที่ลดแรงดันไฟฟ้าให้ต่ำลงเพื่อเหมาะสมกับการใช้งาน โครงสร้างของวงจรจะประกอบด้วย สวิตช์ที่สามารถสั่งให้ “ON” หรือ “OFF” ได้ทุกขณะตามที่ต้องการ ในทางปฏิบัติสวิตช์ที่ใช้ในวงจรจริงคือ อุปกรณ์สวิตซ์ซิ่ง เช่น BJT, MOSFET, ฯลฯ นอกจากนี้ยังมีส่วนประกอบอื่นๆอีกคือ Condenser, Inductance และ Diode ดังแสดงในรูปที่ 2.49 (a) ส่วนการทำงานของ Condenser และ Inductance นั้นจะทำงานแบบ Linear แต่สำหรับ Diode นั้นจะเป็น Non – Linear

เมื่อวงจรประกอบด้วยสวิตช์ที่มีการ “ON” หรือ “OFF” เกิดขึ้นจะต้องทำการแยกวงจรออกเป็น 2 ส่วนคือส่วนที่ ON และ OFF เนื่องจากวงจรแตกต่างกันดังแสดงในรูปที่ 2.49 (b) และ (c) สิ่งที่สำคัญอย่างหนึ่งคือการกำหนดทิศทางการไหลของกระแสและแรงดันที่ตกคร่อมบนอุปกรณ์ทุกตัว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

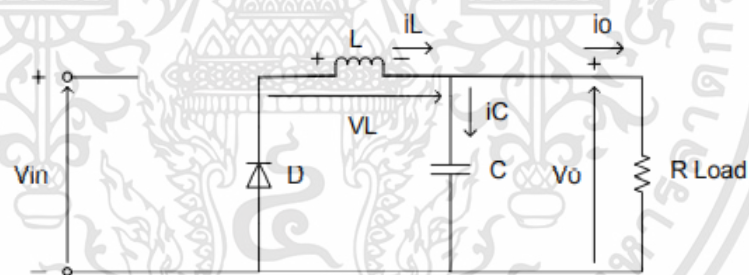
ไม่ว่าสวิตช์จะอยู่ในสถานะ ON หรือ OFF จะต้องมีทิศทางเดียวกันเสมอ มิฉะนั้นชั่วของแรงดัน/กระแสจะไม่ถูกต้อง



(a) Step down DC-DC converter

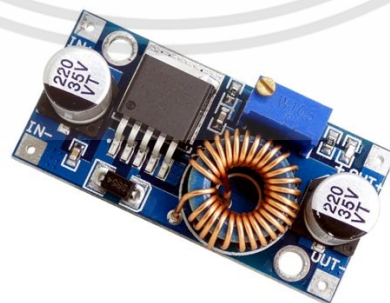


(b) Switch ON circuit



(c) Switch OFF circuit

รูปที่ 2.49 วงจร Step Down Converter

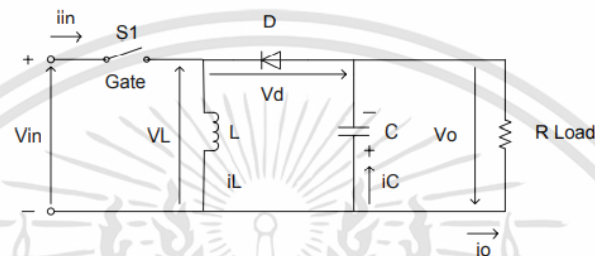


รูปที่ 2.50 Step Down Converter

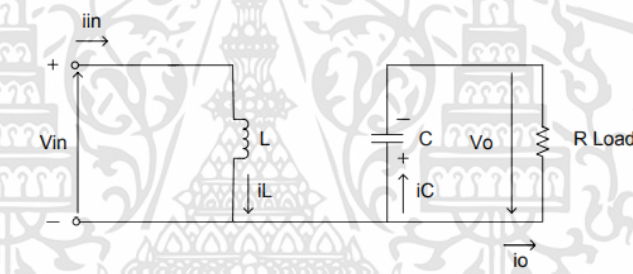
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.16 Step up (Boost) converter

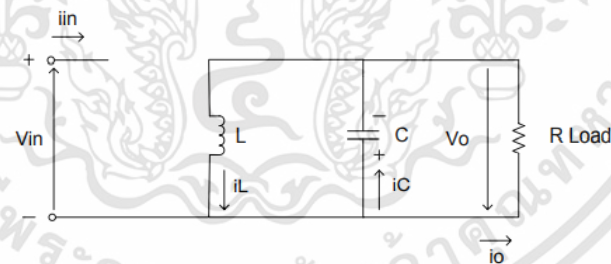
วงจรนี้ทำหน้าที่เพิ่มและลดแรงดันได้ บางครั้งเรียกว่า Step up - down converter เนื่องจากวงจรนี้สามารถใช้กับแรงดันได้กว้างจึงนำมาประยุกต์ใช้ได้กว้างขวางกว่าวงจรที่เป็น Step down หรือ Step up อย่างเดียว เทคนิคในการออกแบบโครงสร้างของวงจรนี้จะนำเอาพื้นฐานของวงจร Step down และ Step up มารวมกันแสดงดังรูปที่ 2.51



(a) Buck boost DC-DC converter



(b) Switch ON circuit



(c) Switch OFF circuit

รูปที่ 2.51 วงจร Buck boost Converter



รูปที่ 2.52 Step up Converter

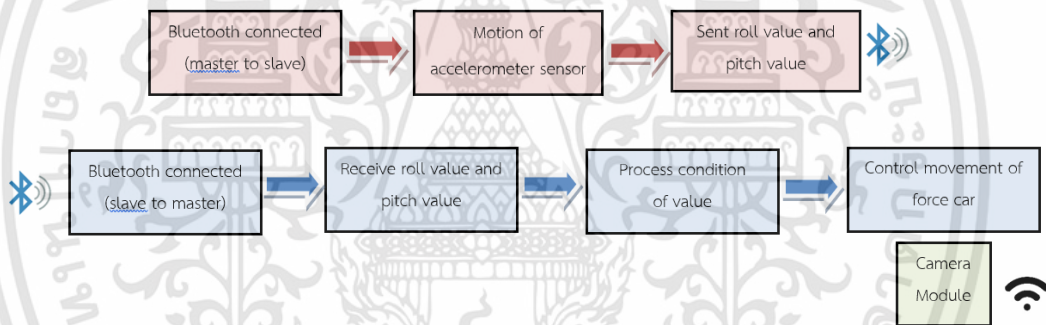
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

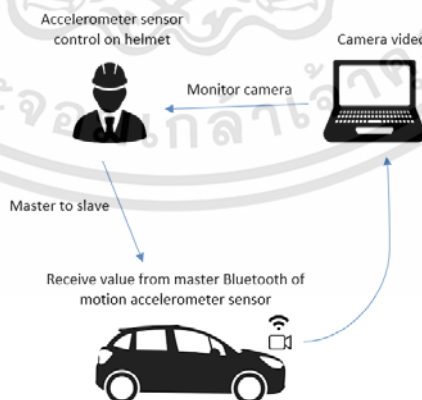
## การออกแบบและวิธีการดำเนินการ

### 3.1 การออกแบบ

การออกแบบการควบคุมรถจำลองด้วยการเคลื่อนไหวของศีรษะ โดยมีการทำงานด้วยการเคลื่อนไหวของ Accelerometer sensor และส่งค่าผ่านการเชื่อมต่อแบบบลูทูธ แบบมาสเตอร์กับสเลฟ โดยสามารถแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนในการควบคุมการทำงานของรถจำลองโดยการรับ-ส่งค่าผ่านบลูทูธ ในการควบคุมการเคลื่อนที่ของรถจำลอง และการแสดงผลของกล้องในการเคลื่อนที่ ณ ตำแหน่งนั้นๆของรถจำลอง โดยเป็นการเชื่อมต่อผ่านสัญญาณอินเทอร์เน็ตไวไฟ และแสดงผลผ่านเว็บเบราว์เซอร์ บล็อกไดอะแกรมการทำงานของระบบควบคุมโดยรวมสามารถแสดงได้ดังในรูปที่ 3.1 และ รูปที่ 3.2



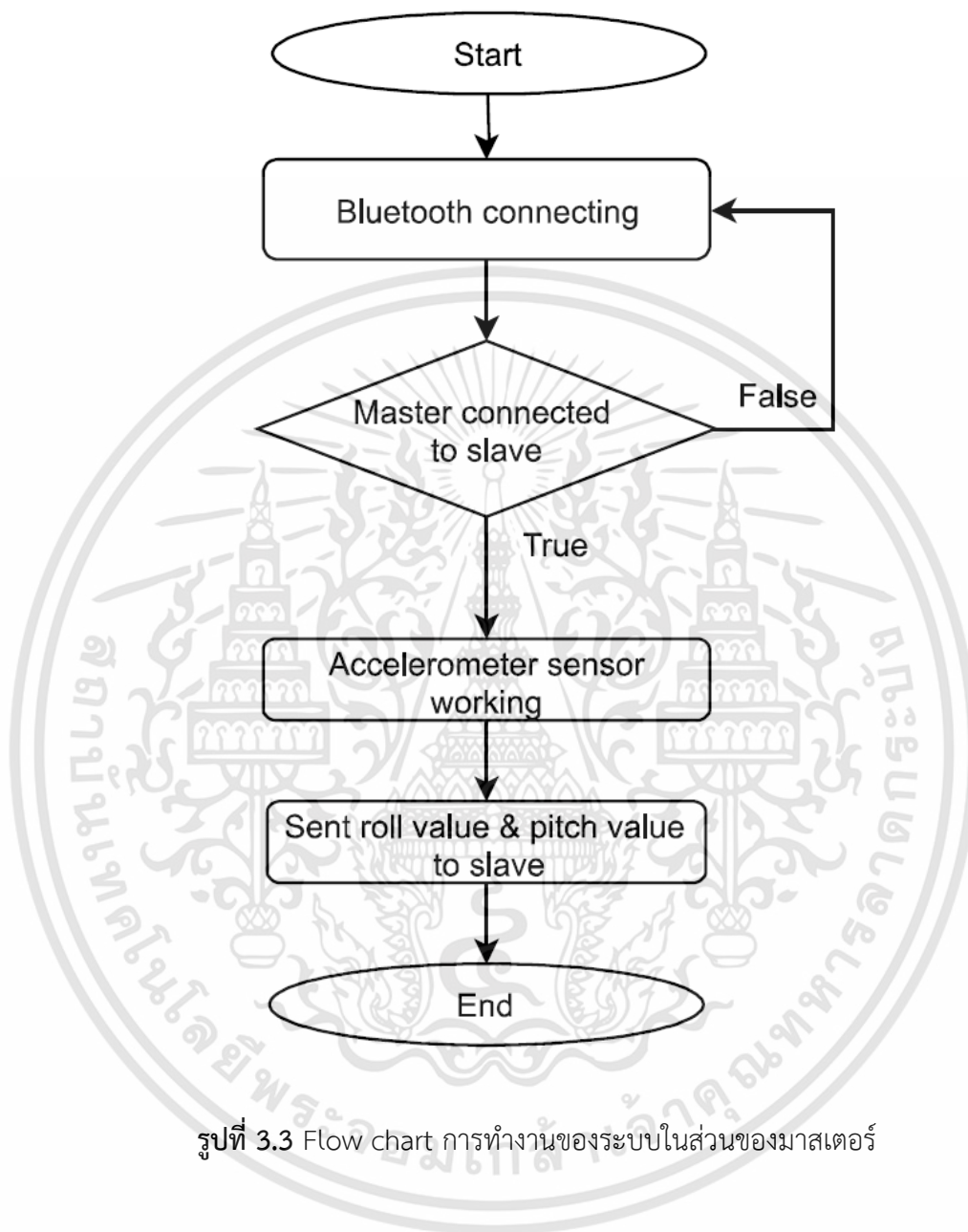
รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานเบื้องต้น



รูปที่ 3.2 วิธีการควบคุมการทำงานและการแสดงผล

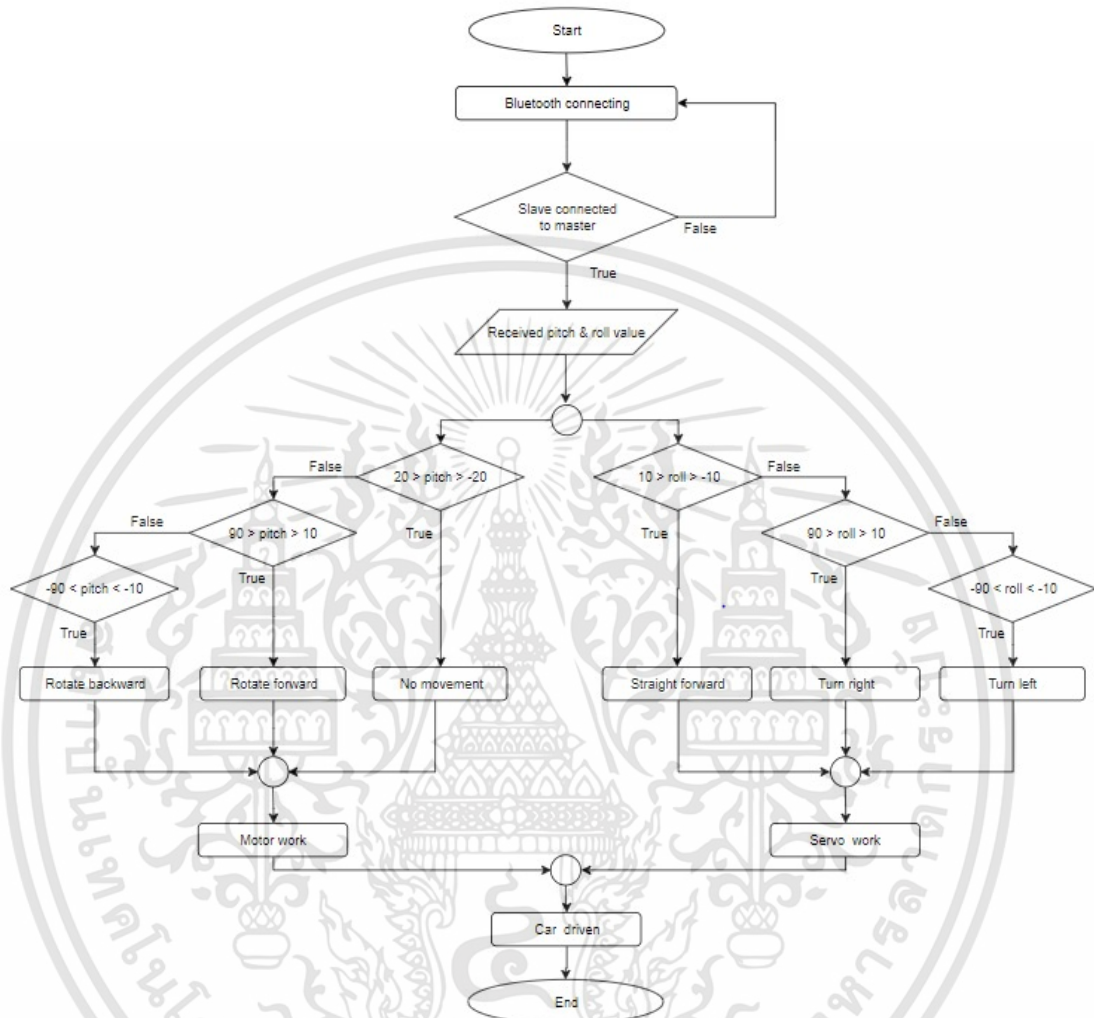
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของระบบมาสเตอร์



รูปที่ 3.3 Flow chart การทำงานของระบบในส่วนของมาสเตอร์

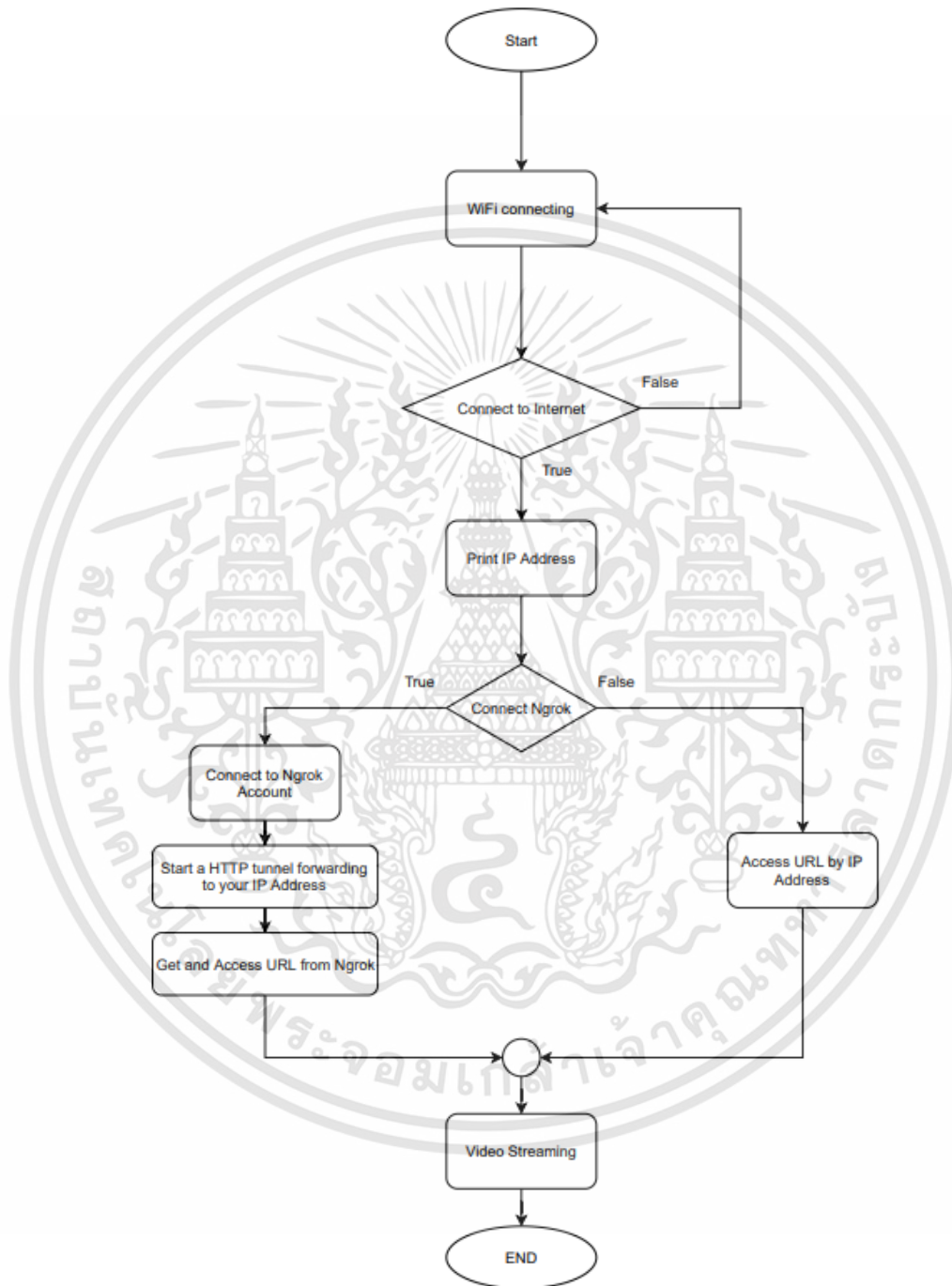
### ขั้นตอนการทำงานของระบบสเลฟ



รูปที่ 3.4 Flow chart การทำงานของระบบในส่วนของสเลฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กล่องแสดงผล

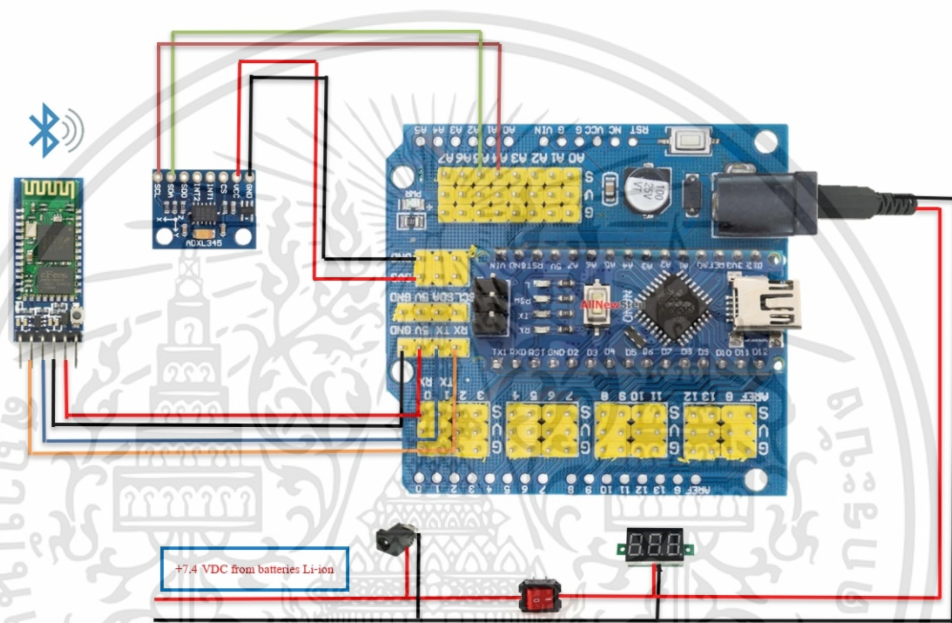


รูปที่ 3.5 Flow chart การทำงานของระบบในส่วนของกล่องแสดงผล

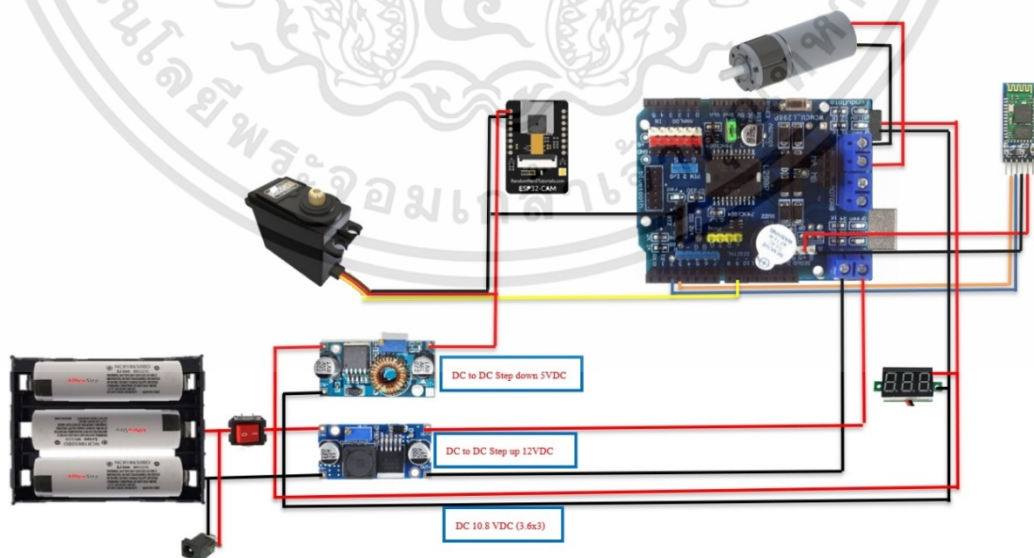
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 ขั้นตอนการทำงานของระบบการควบคุมรถจำลอง

การทำงานในส่วนของการควบคุมรถจำลองสามารถแบ่งการทำงานออกเป็น 2 ส่วนคือ การควบคุมการทำงานของรถจำลองและการแสดงผลของกล้อง โดยในส่วนการควบคุมการทำงานมีการเชื่อมต่อและส่งข้อมูลผ่านการเชื่อมต่อแบบบลูทูธ มีการเชื่อมต่อกันระหว่างมาสเตอร์และสเลฟ ใช้ในการควบคุมการทำงานของรถจำลอง โดยสามารถแสดงการเชื่อมต่อวงจรของอุปกรณ์ได้ดังรูปที่ 3.6



รูปที่ 3.6 การเชื่อมต่ออุปกรณ์ในส่วนของมาสเตอร์



รูปที่ 3.7 การเชื่อมต่ออุปกรณ์ในส่วนของสเลฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 Bluetooth HC-05 (มาสเตอร์-สเลฟ)

ใช้ในการเชื่อมต่อในการส่งค่า เพื่อนำไปใช้ในการควบคุม โดยใช้รูปแบบ มาสเตอร์และสเลฟ ข้อมูลในการส่งจะใช้ค่าจากเซนเซอร์ Accelerometer จากมาสเตอร์ส่งไปยังสเลฟ เพื่อนำค่าที่ได้มาดำเนินการควบคุมรถจำลองตามเงื่อนไขที่ได้กำหนดขึ้น ใช้ย่านความถี่ที่ 2.4 GHz

### 3.2.2 Accelerometer sensor ADXL345

Accelerometer sensor เป็นเซนเซอร์ที่ใช้ในการควบคุมการเคลื่อนที่ของรถจำลอง เมื่อเซนเซอร์ได้มีการติดตั้งที่บริเวณหมวก ในการควบคุมจะใช้การเคลื่อนไหวของหมวก จะทำให้ Accelerometer sensor มีการทำงานเกิดขึ้นค่าที่ได้จะแบ่งเป็น 2 ส่วนคือ ค่า Roll และ ค่า Pitch โดยจะส่ง 2 ค่านี้ ผ่านบลูทูธมาสเตอร์ไปยังบลูทูธ สเลฟ โดยจะนำค่าที่ได้ 2 ค่ามาทำการควบคุมรถจำลอง แบ่งโดยการนำค่า Pitch มาควบคุมการทำงานของมอเตอร์เพื่อใช้ในการเคลื่อนที่ของรถจำลอง เดินหน้า-ถอยหลัง และนำค่า Roll มาควบคุม Servo Motor ในส่วนของทิศทาง การเลี้ยวซ้าย-ขวา โดยค่าที่ได้ 2 ค่านี้จะมีค่าอยู่ในช่วง +90 ถึง -90

### 3.2.3 Microcontroller Arduino

มีการนำบอร์ด ไมโครคอนโทรลเลอร์ 2 รุ่น มาใช้ในการควบคุม โดยในส่วนของมาสเตอร์จะนำบอร์ด Arduino Nano 3.0 มาใช้ในการทำงานร่วมกับเซนเซอร์ Accelerometer และการเชื่อมต่อของบลูทูธเพื่อนำค่าที่ได้จากการทำงานของเซนเซอร์ความเร่งส่งไปยังสเลฟ และอีกส่วนหนึ่งคือบอร์ด Arduino Uno R3 ใช้ในการรับค่าจากเซนเซอร์ Accelerometer ของมาสเตอร์ที่ส่งมา เพื่อควบคุมรถจำลองในการเคลื่อนที่ตามเงื่อนไขที่ได้กำหนดไว้

### 3.2.4 L298P DC Motor Drive Shield

บอร์ดไดร์มอเตอร์ เพื่อใช้ในการควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรง เนื่องจากมอเตอร์ไม่สามารถนำไฟจากบอร์ด Arduino โดยตรงมาทำงานได้ เหตุจากแรงดันไม่เพียงพอที่จะทำให้อมอเตอร์สามารถทำงานได้ และไม่สามารถควบคุมทิศทางได้ จึงต้องนำบอร์ดไดร์ฟมอเตอร์มาใช้ในการควบคุมการทำงาน โดยจะมีวงจร H-Bridge เพื่อใช้ในการควบคุมทิศทางการทำงานและความเร็วของมอเตอร์ด้วยค่า PWM เพื่อกำหนดความเร็วในการทำงานของมอเตอร์

### 3.2.5 ESP32 Camera

โมดูลกล้อง ใช้ในการแสดงผล เพื่อแสดงผลการเคลื่อนที่ของรถจำลองในการเคลื่อนที่ ณ จุดใดๆ โดยมีการเชื่อมต่อผ่านอินเทอร์เน็ตไวไฟ ในรูปแบบ NGROK Server มีการแสดงผลผ่านเว็บเบราว์เซอร์โดยมีการกำหนด ไอพีแอดเดรสของกล้องแบบคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6 ตำแหน่งการวางอุปกรณ์บนรถจำลอง



รูปที่ 3.8 การเชื่อมต่ออุปกรณ์ในส่วนของสเลฟ

หมายเลข 1 ตำแหน่งการติดตั้งโมดูลกล่อง ESP32 เพื่อใช้ในการแสดงผลการเคลื่อนที่ของรถจำลอง ณ ตำแหน่งใดๆ

หมายเลข 2 เป็นตำแหน่งของเซอร์โวมอเตอร์ ใช้ในการควบคุมทิศทางการเลี้ยวซ้าย-ขวา

หมายเลข 3 บอร์ดควบคุมอาตุโนและบอร์ดเสริมไดร์ฟมอเตอร์

หมายเลข 4 แบตเตอรี่ และ มอเตอร์ไฟฟ้ากระแสตรง

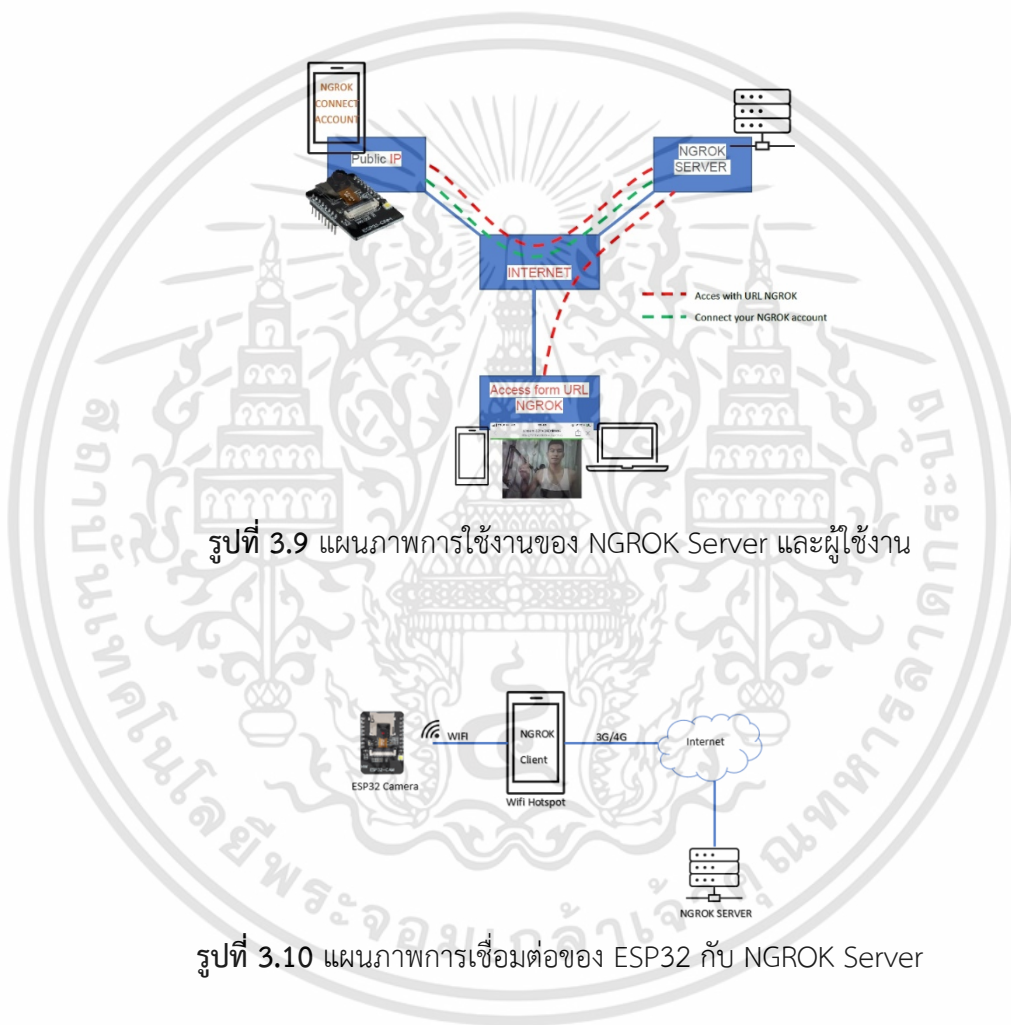
### 3.3 การทำงานในส่วนแสดงผล

การแสดงผลมีการใช้โมดูล ESP32 Camera มาใช้ในการแสดงผล โดยมีการเชื่อมต่ออินเทอร์เน็ตไวไฟที่สามารถออกอินเทอร์เน็ตได้ เพื่อให้ ESP32 Camera ทำงานและแสดงผลผ่านหน้าเว็บเบราว์เซอร์ เนื่องจากการแสดงผลมีข้อจำกัดเพราะต้องใช้ไวไฟในพื้นที่ที่มีความจำกัด เกิดความต้องการในการแสดงผลที่สามารถใช้งานผ่านอินเทอร์เน็ต จึงได้มีการประยุกต์ใช้งาน Ngrok Server เข้ามาร่วมในการเข้าไปใช้งานเว็บแอปพลิเคชันภายในเครือข่าย หรือเครือข่ายส่วนบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 Ngrok Server

เป็นเครื่องมือที่ทำให้เว็บหรือ แอปพลิเคชัน ที่ทำงานบน localhost คอมพิวเตอร์ หรือ เครื่องข่ายภายในสามารถ online เป็น Domain ผ่าน Ngrok Server ทำให้การใช้งานผ่านอินเทอร์เน็ต ง่าย และสะดวกมากยิ่งขึ้นในการทดสอบ หรือใช้งานระบบโดยไม่ต้องเข้าไปในเครือข่ายภายใน มีการรักษาความปลอดภัยในการสร้าง Tunneling เช่นเดียวกับการ VPN แต่จะเป็นการสร้างอุโมงค์หรือ Tunnel เชื่อมต่อกันระหว่าง Ngrok Server กับเครื่องที่ใช้เครือข่ายภายใน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 กล่าวนำ

การทดลองการควบคุมรถจำลองด้วยการเคลื่อนไหวของศีรษะ จากค่าเซนเซอร์ Accelerometer และแสดงผลวิดีโอผ่านระบบอินเทอร์เน็ตไวไฟ ด้วย NGROK Server จะประกอบด้วย การควบคุมรถจำลองด้วยการเคลื่อนไหวของศีรษะที่ได้จากค่าเซนเซอร์ Accelerometer ระยะเวลาที่สามารถควบคุมได้ การตอบสนองในการควบคุม การเชื่อมต่อของ บลูทูธ ในการควบคุม และการแสดงผลของวิดีโอ โดยผ่าน NGROK Server การเชื่อมต่อระบบอินเทอร์เน็ต เพื่อใช้ในการแสดงผลของวิดีโอผ่านเว็บเบราว์เซอร์

#### 4.2 ขั้นตอนการทดลอง

##### 4.2.1 การควบคุมรถ

1. เปิดการทำงานของรถและตัวหมวกที่ใช้ในการควบคุม สังเกตเชื่อมต่อของ บลูทูธ ระหว่าง มาสเตอร์ และ สลฟ เมื่อมีการเชื่อมต่อแล้วจะมีแสดงไฟกระพริบ 2 ครั้ง อย่างช้าๆ



รูปที่ 4.1 แสดงวิธีการควบคุมทิศทาง

2. ทำการสวมหมวกเพื่อทำการควบคุมการเคลื่อนที่ของรถ โดยเงื่อนไขที่กำหนดคือ จะมีการนำค่า Pitch มาใช้ในการควบคุมการเคลื่อนที่ เมื่อมีการก้มศีรษะไปด้านหน้าค่า Pitch อยู่ในระหว่าง 0-90 จะทำให้การเคลื่อนไหวของรถเดินหน้า เอนศีรษะไปทางด้านหลังค่า Pitch อยู่ในระหว่าง 0- -90 จะทำให้รถถอยหลัง และนำค่า Roll มาทำการควบคุมทิศทางในการเลี้ยวซ้ายและเลี้ยวขวา เมื่อเอียงศีรษะไปทางด้านขวาค่า Roll อยู่ระหว่าง 0-90 จะทำให้รถเลี้ยวขวา และเมื่อเอียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศีรษะไปทางด้านซ้าย ค่า Roll จะอยู่ระหว่าง 0- -90 จะทำให้รถเลี้ยวซ้าย และอยู่ในลักษณะปกติของศีรษะจะทำให้รถหยุดนิ่งอยู่กับที่



รูปที่ 4.2 แสดงวิธีการควบคุมการเคลื่อนที่

3. ติดตั้งแอป Strava ใช้ในการวัดระยะทางของรถในการควบคุมที่เคลื่อนที่ได้
4. ทำการทดลองและบันทึกผล และวัดระยะทางจริงที่ได้ในการเปรียบเทียบ

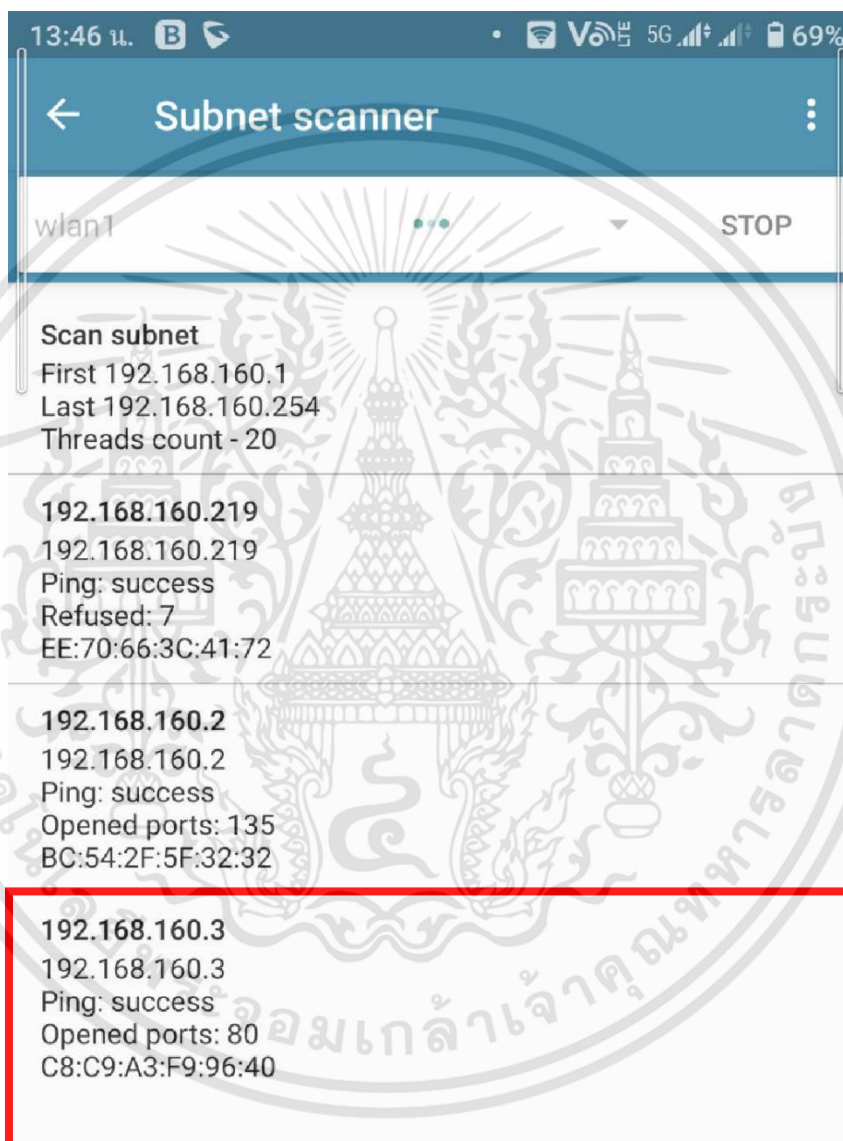


รูปที่ 4.3 Application Strava

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

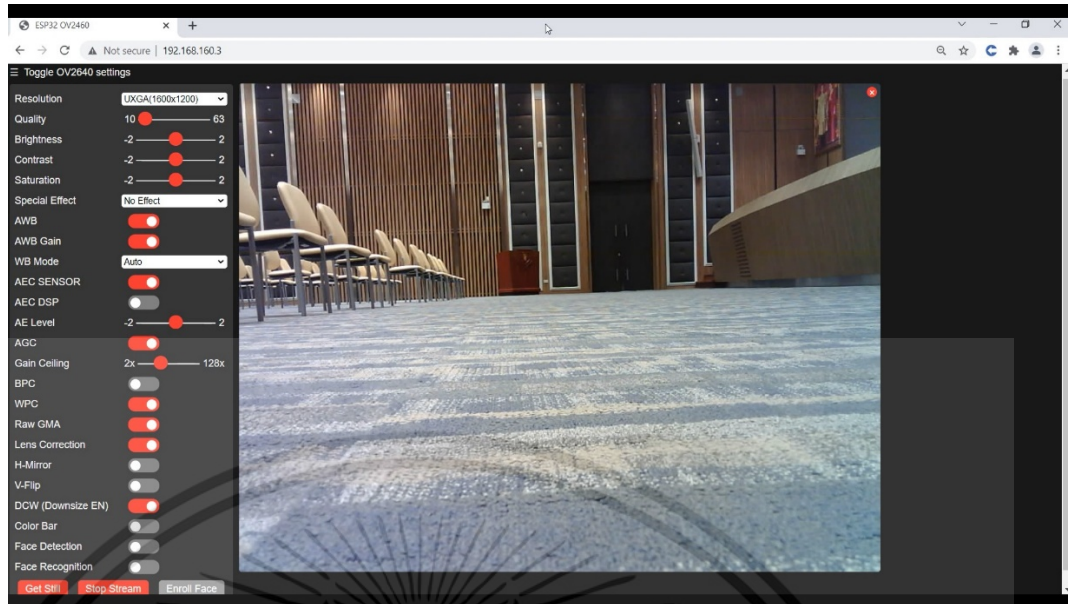
#### 4.2.2 การแสดงผลวิดีโอ

1. เปิด Application subnet scanner เพื่อแสดงค่า ไอพีแอดเดรส ของ module ESP32 CAM โดยจะทำการ Scan ค่า ไอพีแอดเดรส ใน VLAN ที่มีการเชื่อมต่อเดียวกัน ซึ่ง IP ของ ESP32 CAM คือ 192.168.160.3 opened ports: 80 และค่า mac address C8:C9:A3:96:40 แสดงได้ดังรูปที่ 4.4



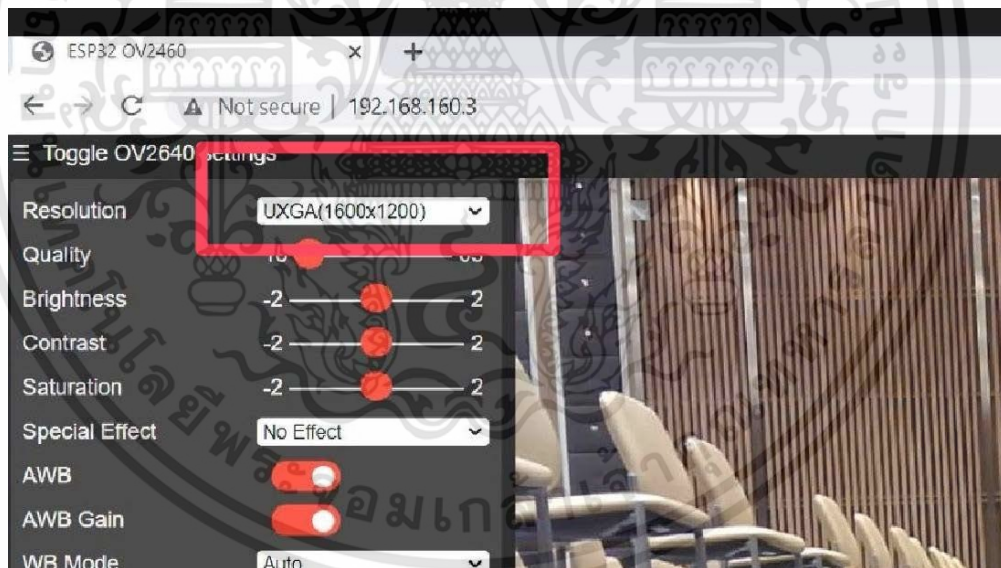
รูปที่ 4.4 แสดงค่า ไอพีแอดเดรส ของ module ESP32 CAM

2. เปิดเว็บเบราว์เซอร์ จากนั้นนำค่า ไอพีแอดเดรส ของกล้องที่ได้จากการรันโปรแกรม พิมพ์ลงแล้วกด Enter จะได้ภาพวิดีโอแบบ Real-Time แสดงดังรูปที่ 4.5



รูปที่ 4.5 เว็บเบราว์เซอร์แสดงภาพวิดีโอแบบ Real-Time

3. สามารถเลือกขนาดของวิดีโอ และสามารถเลือกวิธีการดูแบบ Stream และแบบ Capture ได้ ตามรูปที่ 4.6



รูปที่ 4.6 ปรับขนาดของวิดีโอ รูปแบบ Stream และแบบ Capture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ผลการทดลอง

จากการทดลองเมื่อ บลูทูธ มีการเชื่อมต่อกันแล้วระหว่าง มาสเตอร์ และ สเลฟ ในส่วนของการควบคุมเพื่อวัดระยะทางที่สามารถควบคุมได้นั้น จะมีการเปรียบเทียบระยะทางที่สามารถควบคุมได้โดยเปรียบเทียบกับ ค่าทางทฤษฎี ค่าจาก Application Strava และ ค่าจากการวัดจริง และบันทึกค่าต่างๆเพื่อเปรียบเทียบของการทดลอง



รูปที่ 4.7 ระยะทางการทดลองที่ได้จาก Application strava

#### 4.3.1 เปรียบเทียบผลการทดลอง

จากการทดลองวัดระยะทางที่สามารถใช้ในการควบคุมได้ จะมีการเปรียบเทียบค่าทางทฤษฎี ค่าจาก Application strava และระยะทางจริงโดยการวัด โดยจากการบันทึกการทดลองสามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงผลการเปรียบเทียบค่าในทางทฤษฎี ค่าจาก Application Strava และ ค่าจากการวัดระยะทางจริงที่รถเคลื่อนที่ไปถึง

ครั้งที่	ระยะทางในการควบคุมผ่าน Bluetooth (m)		
	ทฤษฎี	ระยะจริง	Application Strava
1	10-100	30	30
2	10-100	35	30
3	10-100	30	20
4	10-100	25	20
5	10-100	35	30
6	10-100	35	30
7	10-100	35	40
8	10-100	25	30
9	10-100	40	30
10	10-100	35	30
ค่าเฉลี่ย		32.5	29

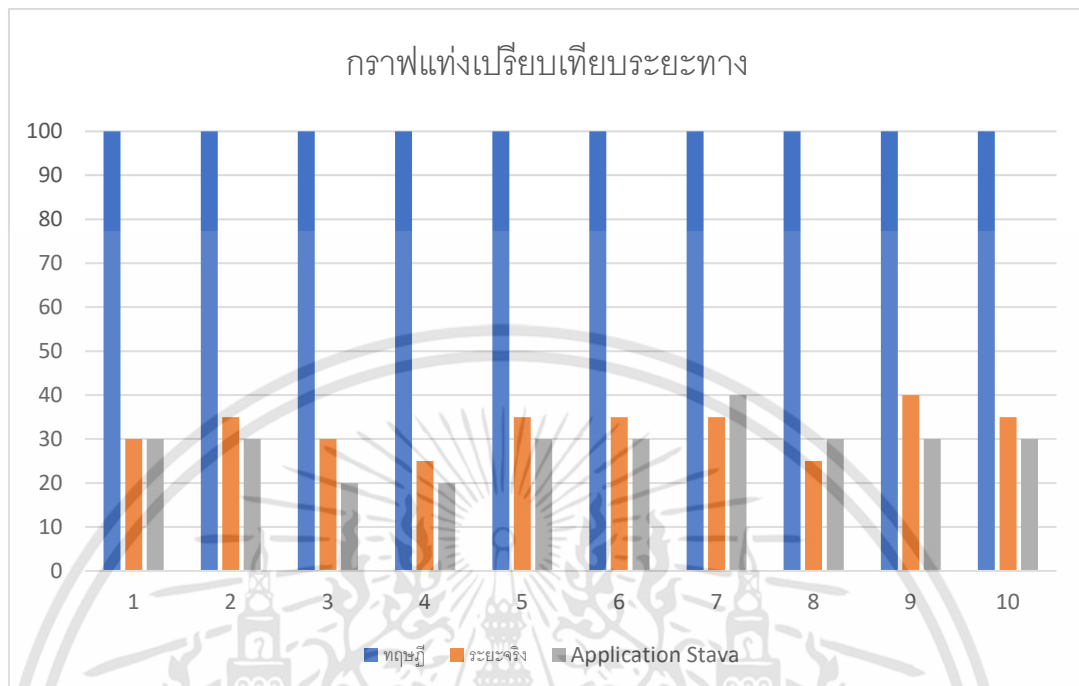
เปอร์เซ็นต์ค่าความผิดพลาด

ค่าความผิดพลาด %error =

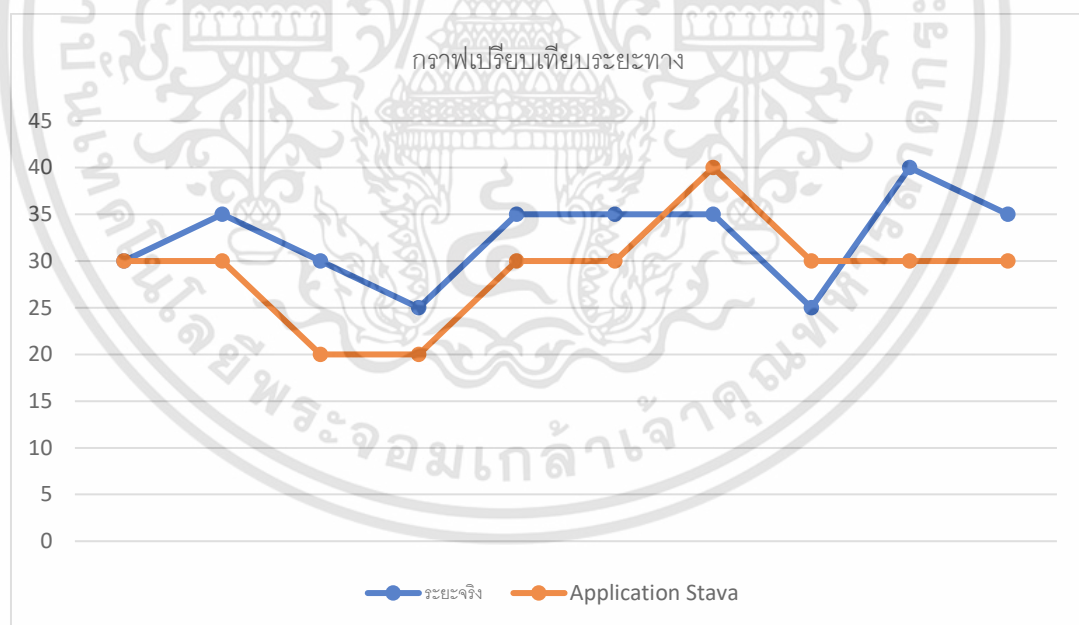
$$\begin{aligned}
 & \frac{\text{ค่าเฉลี่ยระยะทางจริง} - \text{ค่าเฉลี่ยระยะทางจากแอปพลิเคชัน}}{\text{ค่าเฉลี่ยระยะทางจริง}} \times 100 \\
 & = \frac{32.5 - 29}{32.5} \times 100 \\
 & = 10.76 \%
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำข้อมูลที่ได้แสดงเป็นกราฟได้ดังนี้

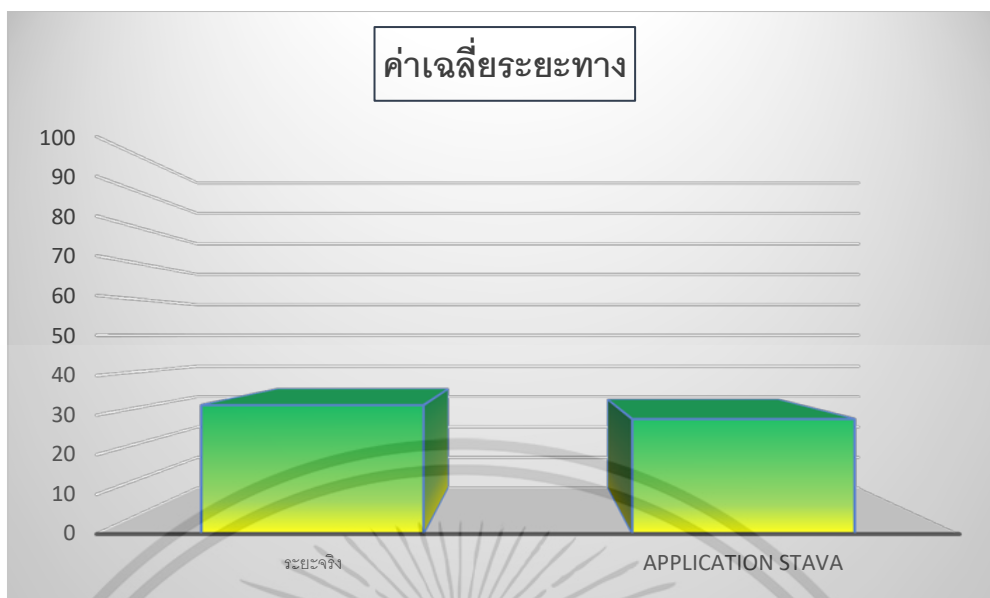


รูปที่ 4.8 กราฟแท่งเปรียบเทียบค่าระยะทางที่วัดได้ในแต่ละครั้ง



รูปที่ 4.9 กราฟเปรียบเทียบค่าระยะทางที่วัดได้ในแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟแท่งแสดงค่าระยะทางเฉลี่ยที่วัดได้ของระยะจริงกับ Application Strava

#### 4.3.2 การทดลองการตอบสนองการควบคุม

จากการทดลองการตอบสนองการควบคุมการทำงาน โดยทำการควบคุมรถจำลองใน บล็อก ขนาด 3x5 m โดยทำการควบคุมการเคลื่อนที่ในแต่ละบล็อก เพื่อดูผลการตอบสนองในการ ควบคุม ซึ่งสามารถแสดงได้ดังตารางที่ 4.2 ดังนี้

ตารางที่ 4.2 ระยะการตอบสนองการควบคุมการเคลื่อนที่เทียบกับระยะทาง

ครั้งที่	ระยะทาง (m)	ผลการตอบสนอง
1	5	สามารถตอบสนองได้ปกติ
2	10	สามารถตอบสนองได้ปกติ
3	15	สามารถตอบสนองได้ปกติ
4	20	สามารถตอบสนองได้ปกติ
5	25	ผลการตอบสนองค่อนข้างช้ากว่าการควบคุม
6	30	ผลการตอบสนองผิดพลาดมีระยะเวลาการหน่วงที่มากในการควบคุม
7	35	ผลการตอบสนองผิดพลาดมีระยะเวลาการหน่วงที่มากในการควบคุม ผลการ ควบคุมผิดพลาด
8	40	การควบคุมไม่ตอบสนอง เกิดความผิดพลาดสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตารางที่ 4.4 ตารางเปรียบเทียบค่าจาก Servo กับค่ามุมที่วัดได้ ในการเลี้ยวไปทางซ้าย

ค่าที่วัดได้จากการเลี้ยวของล้อไปทางซ้าย $\theta^\circ$					
Value	Roll Master	Roll Slave	Servo	Left	Right
-10	-10	-10	100	90	90
-20	-20	-20	110	100	100
-30	-30	-30	120	110	110
-40	-40	-40	130	115	115
น้อยกว่า -40	-60	-60	130	115	115
-40	-40	-40	130	116	116
-30	-30	-30	120	110	110
-20	-20	-20	110	100	100
-10	-10	-10	100	90	90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลอง

จากการทดลองเป็นการทดลองในการควบคุมการควบคุมรถจำลองและการแสดงผลของกล้อง ในการควบคุมจะนำค่าจากเซนเซอร์ Accelerometer มาใช้ในการควบคุมโดยมีการส่งผ่าน บลูทูธ ระหว่าง มาสเตอร์ และ สเลฟ แบ่งเป็นค่า Roll และ Pitch มีการนำค่า Roll มาใช้ในการควบคุมทิศทางในการเลี้ยวซ้ายและเลี้ยวขวาของ Servo Motor ซึ่งค่าที่อยู่ระหว่าง 0-90 จะควบคุมทิศทาง การเลี้ยวขวาและ 0- -90 จะควบคุมทิศทาง การเลี้ยวซ้าย แต่จะมีการกำหนดเงื่อนไขในการทำงาน โดยถ้าค่าอยู่ในช่วงระหว่าง 10 - -10 จะทำให้รถจำลองไม่มีการทำงานของ Servo Motor ในส่วนของ ค่า Pitch จะนำมาใช้ในการควบคุมการเคลื่อนที่โดยมีเงื่อนไขถ้าค่า อยู่ระหว่าง 0-90 จะทำให้รถ จำลองเคลื่อนที่ไปข้างหน้า และ ค่าระหว่าง 0- -90 จะทำให้รถเคลื่อนที่ถอยหลัง และมีเงื่อนไขโดยค่า Pitch ที่อยู่ระหว่าง 20 - -20 จะทำให้ไม่เกิดการเคลื่อนหรือทำงานของมอเตอร์ขึ้น

ตารางที่ 5.1 ค่าองศาในการควบคุมทิศทาง (Roll)

ค่าองศาในการควบคุมทิศทางของ Servo Motor	ความหมาย
0-90	เลี้ยวขวา
0- -90	เลี้ยวซ้าย
10- -10	หยุดนิ่ง

ตารางที่ 5.2 ค่าองศาในการควบคุมการเคลื่อนที่ (Pitch)

ค่าองศาในการควบคุมทิศทางของ DC Motor	ความหมาย
0-90	เดินหน้า
0- -90	ถอยหลัง
20- -20	หยุดนิ่ง

ในขั้นตอนแรกในการทำงานของรถจำลองจำเป็นต้องมีการเปิดให้ Module Bluetooth HC-05 มีการทำงานเสียก่อน โดยจะใช้เวลาประมาณ 5-10 วินาทีในการเชื่อมต่อกันระหว่าง มาสเตอร์ และ สเลฟ โดยสังเกตจากไฟที่กระพริบอย่างช้าๆ เมื่อมีการเชื่อมต่อกัน จะสามารถควบคุมการทำงานของรถจำลองได้ การควบคุมจะขึ้นอยู่กับการเคลื่อนไหวต่างๆตามค่าของเซนเซอร์ Accelerometer ไม่ว่าจะเป็นการเดินหน้าหรือถอยหลังจะใช้ค่า Pitch ในการควบคุมการทำงานโดยการก้มหน้าหรือเงยหน้า ขึ้น และค่า Roll มาใช้ควบคุมการทำงานการเลี้ยวซ้าย-ขวาด้วยการเอียงคอไปทางด้านซ้ายหรือ ด้านขวาในทิศทางเดียวกับต้องการเลี้ยว โดยรถจะมีการเคลื่อนที่ไปตามการเคลื่อนไหวของศีรษะ และ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงให้เห็นถึงการทำงานที่เกิดขึ้น โดยมีการทดลองการทำงานของรถจำลองเพื่อวัดระยะในการเคลื่อนที่ของรถจำลองที่สามารถเคลื่อนที่ไปได้ไกลที่สุด โดยมีการบันทึกและเปรียบเทียบค่าที่ได้จากการทดลองเพื่อแสดงให้เห็นที่ระยะทางเฉลี่ยที่สามารถทำงานได้ และยังมีการควบคุมการตอบสนองในการทำงานของรถจำลอง เพื่อดูผลการตอบสนองเทียบกับระยะทางที่ไกลขึ้นในการควบคุม และสุดท้ายคือการวัดมุมมองของล้อในการเลี้ยวซ้ายและเลี้ยวขวาเพื่อเปรียบเทียบกับค่า Roll ที่ได้จากการ map ค่าให้มีการทำงานที่เหมาะสมในการควบคุมการเลี้ยว โดยมีกำหนดค่าที่ map ไว้ในการเลี้ยวได้คือค่าอยู่ระหว่าง 50-130 ในการควบคุมการทำงานของ Servo Motor และบันทึกผลเปรียบเทียบค่าจาก Servo Motor และค่าจากการวัดจริงลงในตาราง เพื่อเปรียบเทียบค่าที่เกิดขึ้น

ในส่วนของกล้องวิดีโอ จะมีการเชื่อมต่อกับอินเทอร์เน็ตไวไฟ โดยในที่นี้จะมีการเชื่อมต่อกับมือถือที่ได้ทำการเปิดโหมด WIFI hotspot ไว้ โดยสามารถตรวจสอบการเชื่อมต่อของกล้อง ESP32 CAM ได้โดยใช้การใส่ Application Subnet scanner ในการค้นหา ไอพีแอดเดรส จะมีการแสดงค่าของ ไอพี ของ ESP32 CAM และนำค่าที่ได้เปิดดูผ่าน เว็บเบราว์เซอร์ โดยสามารถดูผลการแสดงของกล้อง ESP32 ประกอบพร้อมกับการเคลื่อนที่ของรถจำลองในเวลาเดียวกัน และสามารถปรับความละเอียดในการแสดงผล, ค่าคุณภาพของภาพ, ค่าความสว่าง, ค่าความเข้มของภาพ และค่าเอฟเฟคของภาพในการแสดงผล ตามความเหมาะสมในการแสดงในการทำงาน

ในการทดลองจุดประสงค์หลักก็คือดูผลการตอบสนองในการควบคุมการทำงานของรถจำลอง, ระยะทางที่สามารถควบคุมการทำงานได้เฉลี่ย การแสดงผลของกล้องพร้อมกับการเคลื่อนที่ของรถจำลอง การควบคุมการทำงานจำเป็นต้องมีการฝึกในการควบคุม เพื่อให้รถจำลองสามารถควบคุมไปได้ในลักษณะที่ต้องการและตอบสนองได้ตามการเคลื่อนไหว และยังให้ความสำคัญในด้านยานยนต์ไฟฟ้าและทางด้านการแพทย์โดยเป็นการคิดค้นวิธีการควบคุมยานพาหนะวิธีใหม่ในอนาคต โครงการนี้ได้เรียนรู้ในศาสตร์วิชาต่างๆในสาขาวิชาวิศวกรรมการวัดคุม หลักสูตรบัณฑิตพันธุ์ใหม่ ที่เน้นทางด้านยานยนต์ไฟฟ้า เช่น รายวิชาไมโครคอนโทรลเลอร์, Sensor and Transducers, Data Communication & Network และอีกมากมายมาประยุกต์ใช้ในการทำโครงการนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

โครงการนี้ยังคงสามารถต่อยอดได้เพิ่มเติม โดยไม่ว่าจะเป็นการพัฒนา บลูทูธ เวอร์ชันที่ใหม่ในอนาคตเพื่อความเสถียรในการรับ-ส่งค่าในการควบคุมการทำงานที่เที่ยงตรงและแม่นยำมากขึ้น และระยะที่ในการควบคุมที่ไกลขึ้น หรือจะเป็นการควบคุมการทำงานผ่านระบบอินเทอร์เน็ตไวไฟแทนการใช้ระบบ บลูทูธ ก็สามารถพัฒนาต่อยอดให้มีการทำงานที่เสถียร และระยะทางควบคุมได้ไกลยิ่งขึ้นเช่นกัน ในส่วนของกล้องแสดงผล ESP สามารถต่อยอดโดยการเชื่อมต่อกับผู้ให้บริการ Platform web server ในการเชื่อมต่อเพื่อดูผลการแสดงของกล้อง ESP32 ในที่ใดที่หนึ่งได้ โดยไม่จำเป็นต้องอยู่ในระบบเน็ตเวิร์คเดียวกัน ซึ่งเป็นแนวทางในการต่อยอดได้ในอนาคตและสามารถพัฒนาโครงการให้ดียิ่งขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

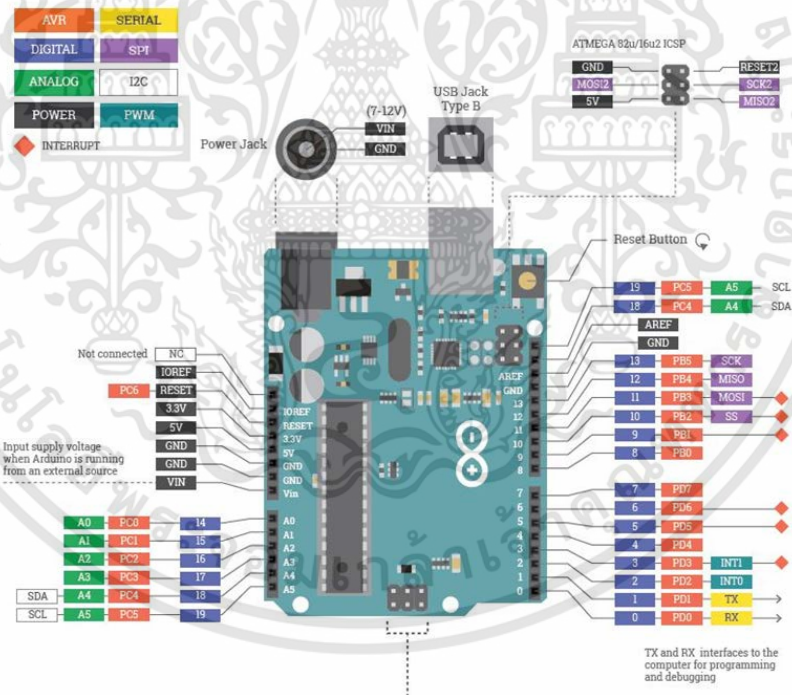
- [1] สุวิทย์ กิระวิทยา (2562) “การใช้งานไมโครคอนโทรลเลอร์และเซนเซอร์ในการสร้างนวัตกรรม” พิมพ์ครั้งที่ 1 หน้า 107 – 114 จาก <http://suwitkiravittaya.eng.chula.ac.th/download/B2i2019eBookMain.pdf>
- [2] วิทยาลัยเทคนิคสระบุรี “การควบคุมเซอร์โวมอเตอร์ด้วย ARDUINO” เอกสารประกอบการสอนวิชา ไมโครคอนโทรลเลอร์เบื้องต้น หน้า 1 – 23. ค้นวันที่ 24 พฤษภาคม 2564 จาก [http://www.sbt.ac.th/new/sites/default/files/TNP\\_Unit\\_9.pdf](http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_9.pdf)
- [3] ผศ.ดร. เดชฤทธิ์ มณีธรรม (2560) “คัมภีร์การใช้งาน ไมโครคอนโทรลเลอร์ Arduino” หน้า 1 – 31 ค้นเมื่อ 5 พฤษภาคม 2564
- [4] คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น (2547). Step down (Buck) converter, Step up (Boost) converter หน้า 91 – 108. ค้นวันที่ 27 ตุลาคม 2564
- [5] มหาวิทยาลัยมหาสารคาม (2552) “โครงการควบคุมเครื่องใช้ไฟฟ้าด้วยมือถือผ่านบลูทูธนี้เป็นโครงการที่ใช้โทรศัพท์มือถือ” หน้า 1 – 30 ค้นเมื่อ 4 พฤษภาคม 2564
- [6] noah\_arduino, Using ESP32 cam with Arduino. From Using ESP32 cam with arduino - Arduino Project Hub.
- [7] Fitrox Electronics, “การสื่อสารแบบ SPI” Retrieved 20 March 2021. From <http://fitrox.lnwshop.com>
- [8] Elecrow, The specific installation steps, Model car, Retrieved 20 March 2021, From [https://www.elecrow.com/download/4wd\\_CAR\\_Install\\_instructions.pdf](https://www.elecrow.com/download/4wd_CAR_Install_instructions.pdf)
- [9] Synergy Electronic Supply co.ltd. HC-05 Bluetooth Module User’s Manual, Retrieved 4 May 2021. From [www.synes.com](http://www.synes.com).

## ภาคผนวก ก

# ไมโครคอนโทรลเลอร์

### ก.1 คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

Arduino เป็นไมโครคอนโทรลเลอร์บอร์ดแบบสำเร็จรูปในยุคปัจจุบัน ซึ่งถูกสร้างมาจากคอนโทรลเลอร์ ตระกูล ARM ของ ATMEL ข้อดีของไมโครคอนโทรลเลอร์บอร์ดคือเรื่องของ Open Source ที่สามารถนำไป พัฒนาต่อเป็นอุปกรณ์ต่างๆได้และความสามารถในการเพิ่ม Boot Loader เข้าไปที่ตัว ARM จึงทำให้การ อัปเดตโค้ด เข้าตัวบอร์ดสามารถทำได้ง่ายขึ้น และยังมีการพัฒนา Software ที่ใช้ในการควบคุมตัวบอร์ด ของ Arduino มีลักษณะเป็นภาษา C++ ตัวบอร์ดสามารถนำโมดูลมาต่อเพิ่ม ซึ่งทาง Arduino เรียกว่าเป็น shield เพื่อเพิ่มความสามารถเพิ่มขึ้น



ก.1 Pin OUT ของ Arduino Uno R3

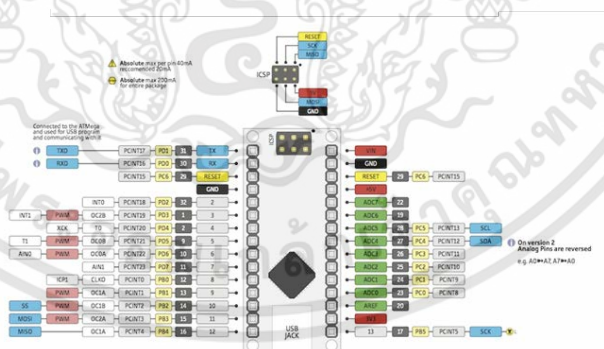
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ Arduino Uno R3

1. ใช้ไอซีไมโครคอนโทรลเลอร์ ATmega328
2. ใช้แรงดันไฟฟ้าเลี้ยงไอซีไมโครคอนโทรลเลอร์ ATmega328 มีค่า 5 โวลต์
3. แรงดันไฟฟ้าป้อนที่บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 อยู่ในช่วง 7 - 12 โวลต์
4. มีพอร์ตดิจิตอลอินพุต/เอาต์พุต (Digital VO) จำนวน 14 พอร์ต (มี PWM output จำนวน 6 พอร์ต)
5. มีพอร์ตอนาล็อกอินพุต (Analog Input) จำนวน 6 พอร์ต
6. สามารถจ่ายกระแสไฟฟ้า แต่ละพอร์ตได้ 40 มิลลิแอมป์ (mA)
7. สามารถจ่ายกระแสไฟฟ้าในพอร์ต 3.3 V จ่ายได้ 50 มิลลิแอมป์ (mA)
8. มีพื้นที่หน่วยความจำโปรแกรม 32 กิโลไบต์ (KB)
9. มีพื้นที่หน่วยความจำชั่วคราวแบบ SRAM 2 กิโลไบต์ (KB)
10. มีพื้นที่หน่วยความจำถาวรแบบ EEPROM 1 กิโลไบต์ (KB)
11. ใช้ความถี่สัญญาณนาฬิกา 16 เมกะเฮิร์ตซ์ (MHz)

## ก.2 คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano 3.0

บอร์ด Arduino Nano ออกแบบมาให้มีขนาดเล็ก และใช้กับงานทั่วไป ใช้ชิปไอซีไมโครคอนโทรลเลอร์เบอร์ ATmega328P โปรแกรมผ่านโปรโตคอล UART มีชิป USB to UART มาให้ ใช้ Mini USB เชื่อมต่อกับคอมพิวเตอร์



ก.2 Pin OUT ของ Arduino Nano 3.0

คุณสมบัติของ Arduino Nano 3.0

1. แรงดันไฟฟ้าที่ใช้งาน (ระดับตรรกะ): 5 V
2. แรงดันไฟฟ้าขาเข้า (แนะนำ): 7-12 V
3. แรงดันไฟฟ้าขาเข้า (จำกัด ): 6-20 V
4. Digital I / O Pins: 14 (ซึ่ง 6 ขาเอาต์พุต PWM)
5. ช่องใส่แบบอะนาล็อก: 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กระแสไฟกระแสตรงต่อขา I / O: 40 mA
7. หน่วยความจำแฟลช 32 KB (ATmega328) ซึ่งมี 2 กิโลไบต์ใช้สำหรับบูตโหลด
8. SRAM: 2 KB (ATmega328)
9. EEPROM: 1 KB (ATmega328)
10. ความเร็วสัญญาณนาฬิกา: 16 MHz
11. ขนาด: 0.73 "x 1.70"

### ก.3 การติดตั้งบอร์ด Arduino Uno R3 และ Arduino Nano 3.0 ลงบนโปรแกรม Arduino IDE

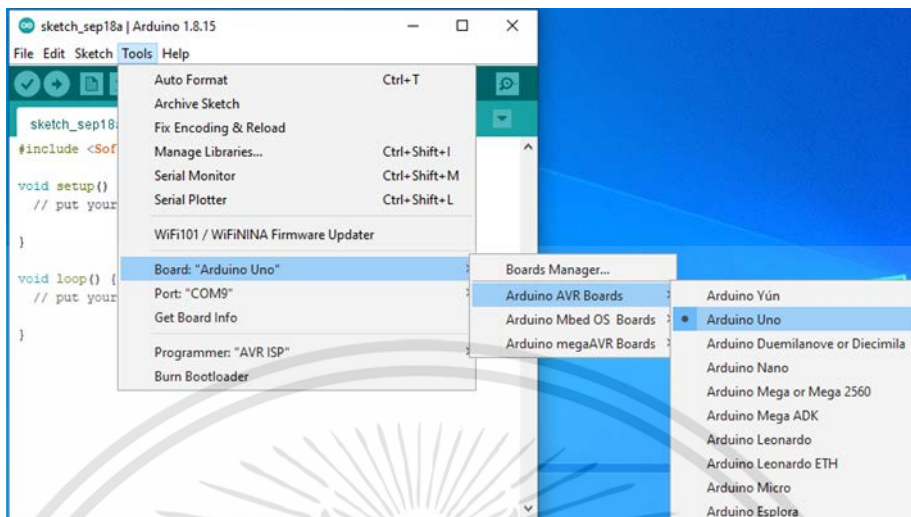
ในการใช้งานจำเป็นต้องติดตั้งโปรแกรม Arduino IDE ซึ่งสามารถโหลดได้ที่ <https://www.arduino.cc/en/software>

#### ก.3 โปรแกรม Arduino IDE

เมื่อทำการติดตั้งเสร็จแล้ว ให้เปิดโปรแกรม Arduino IDE ขึ้นมา

Arduino Uno R3

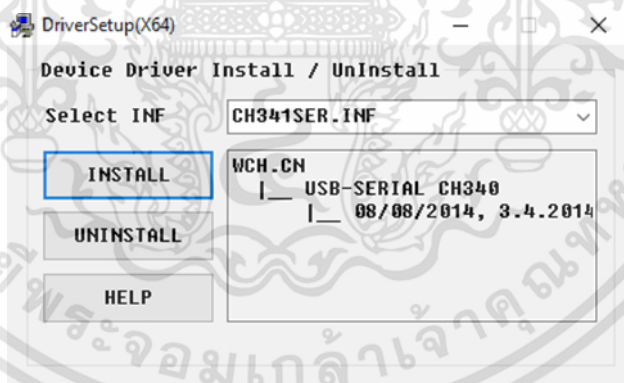
- 1) ไปที่ Tools -> Board เลือก Arduino AVR Boards ไปที่ Arduino Uno



ก.4 การเลือกการใช้งานบอร์ด Arduino Uno

Arduino Nano 3.0

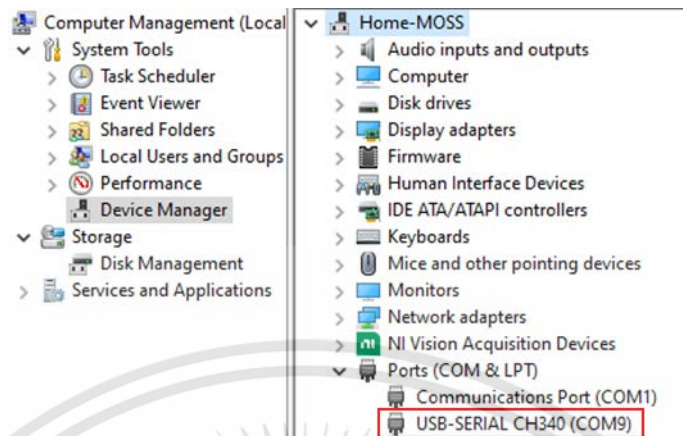
1) Download driver CH340 เพื่อใช้ทำงานของบอร์ด Arduino Nano 3.0 โดยสามารถดาวน์โหลดได้ที่ <http://download.ab.in.th/download.php?file=CH341SER.zip> เมื่อดาวน์โหลดเสร็จแล้วทำการติดตั้งโปรแกรม



ก.5 ติดตั้งไดร์เวอร์ CH340

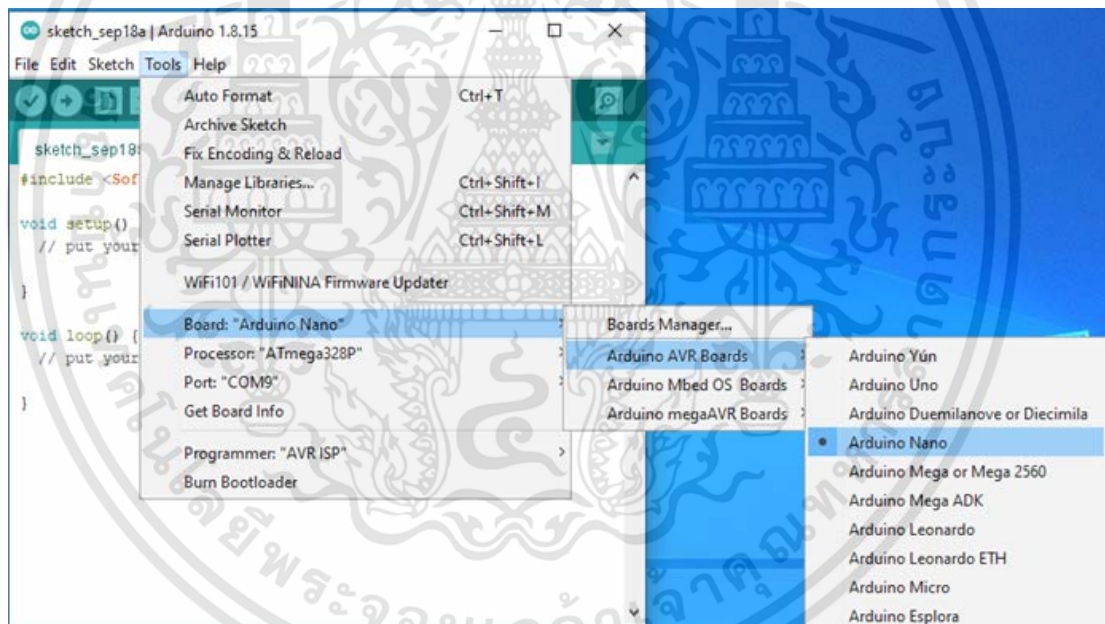
2) เมื่อติดตั้งเสร็จสิ้น เข้าไปยัง Device manager เพื่อดูความสำเร็จของการติดตั้งไดร์เวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก.6 แสดง Serial port ที่เชื่อมต่อกับบอร์ด Nano 3.0

3) ไปที่ Tools -> Board เลือก Arduino AVR Boards ไปที่ Arduino Nano

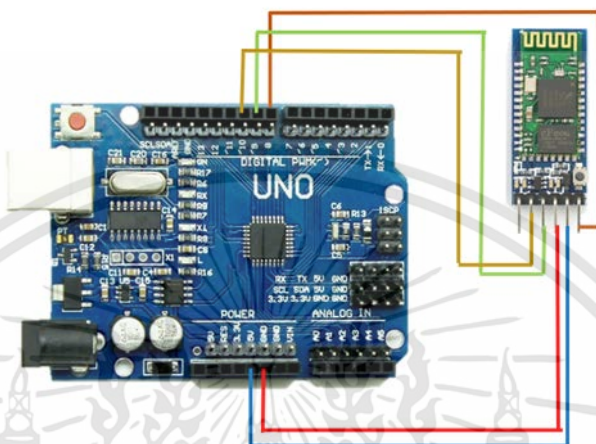


ก.7 การเลือกใช้งานบอร์ด Arduino Nano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### ก.4 การตั้งค่า บลูทูธ เชื่อมต่อแบบ มาสเตอร์ และ สเลฟ

ในการกำหนด บลูทูธ เป็น มาสเตอร์ หรือ สเลฟ ต้องมีการตั้งค่าผ่านคำสั่งของ บลูทูธ โดยมีการต่อวงจรและใช้โปรแกรมเพื่อเข้าไปตั้งค่าได้ดังนี้



ก.8 วงจรในการเชื่อมต่อเข้า AT Command ของ บลูทูธ

โปรแกรม

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(9, 10);

void setup()
{
  pinMode(8, OUTPUT);
  digitalWrite(8, HIGH);
  Serial.begin(9600);
  BTSerial.begin(9600);
  delay(1000);
  Serial.println("Enter AT commands:");
}

void loop()
{
  if (BTSerial.available())
    Serial.write(BTSerial.read());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Serial.available())
  BTSerial.write(Serial.read());
}

```

### การตั้งค่า

เมื่อ compile code และ เชื่อมต่อวงจรเสร็จสิ้น ทำการเปิดหน้าต่าง Serial monitor เลือก รูปแบบเป็น Both NL & CR โดยการตั้งค่าของ มาสเตอร์ และ สเลฟ จะมีคำสั่งในการตั้งค่าดังนี้

#### Slave

AT+UART = 9000 : ตั้งค่า baud rate ที่ 9000

AT+ROLE = 0 : ตั้งค่าใน Mode slave

AT+CMODE = 0 : เชื่อมต่อกับแอตเดรสเฉพาะ

AT+ADDR? : แสดงค่าแอตเดรสของ slave คือ 00:19:10:09:19:A4



ก.9 แสดงการตั้งค่าของ บลูทูธ สเลฟ

#### Master

AT+UART = 9000 : ตั้งค่า baud rate ที่ 9000

AT+ROLE = 1 : ตั้งค่าใน โหมดมาสเตอร์

AT+CMODE = 0 : เชื่อมต่อกับแอตเดรสเฉพาะ

AT+BIND = 0019:10:0919A4 : กำหนดค่าแอตเดรสที่เชื่อมต่อ คือ 0019:10:0919A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COM9
Enter AT commands:
OK
+UART:9600,0,0
OK
+ROLE:1
OK
+CMODE:0
OK
+BIND:0019:10:0919A4
OK
Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

```

ก.10 แสดงการตั้งค่าของ บลูทูธ มาสเตอร์

### ก.5 โปรแกรมที่ใช้ในการส่งค่า-รับค่าในการควบคุม

Master (ส่งค่า)

```

#include <Wire.h> // Wire library - used for I2C communication
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(4, 5);
int ADXL345 = 0x53; // The ADXL345 sensor I2C address
float X_out, Y_out, Z_out; // Outputs
float roll,pitch,x,y,z=0;
int data1,data2 ;

void setup() {
  Serial.begin(9600); // Initiate serial communication for printing the results on the
  Serial monitor
  BTSerial.begin(9600);

  Wire.begin(); // Initiate the Wire library
  // Set ADXL345 in measuring mode
  Wire.beginTransmission(ADXL345); // Start communicating with the device
  Wire.write(0x2D); // Access/ talk to POWER_CTL Register - 0x2D
  // Enable measurement
  Wire.write(8); // Bit D3 High for measuring enable (8dec -> 0000 1000 binary)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wire.endTransmission();
delay(10);

//Off-set Calibration
//X-axis
Wire.beginTransmission(ADXL345);
Wire.write(0x1E);
Wire.write(1);
Wire.endTransmission();
delay(10);
//Y-axis
Wire.beginTransmission(ADXL345);
Wire.write(0x1F);
Wire.write(-2);
Wire.endTransmission();
delay(10);
//Z-axis
Wire.beginTransmission(ADXL345);
Wire.write(0x20);
Wire.write(-9);
Wire.endTransmission();
delay(10);
}

void loop() {
// === Read accelerometer data === //
Wire.beginTransmission(ADXL345);
Wire.write(0x32); // Start with register 0x32 (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(ADXL345, 6, true); // Read 6 registers total, each axis value is
stored in 2 registers

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

X_out = ( Wire.read() | Wire.read() << 8); // X-axis value
X_out = X_out / 256; //For a range of +-2g, we need to divide the raw values by 256,
according to the datasheet
Y_out = ( Wire.read() | Wire.read() << 8); // Y-axis value
Y_out = Y_out / 256;
Z_out = ( Wire.read() | Wire.read() << 8); // Z-axis value
Z_out = Z_out / 256;

x = 0.5 * x + 0.5 * X_out;
y = 0.5 * y + 0.5 * Y_out;
z = 0.5 * z + 0.5 * Z_out;
// Calculate Roll and Pitch (rotation around X-axis, rotation around Y-axis)
roll = atan(y / z) * 180 / Pi;
pitch = atan(-1 * x / sqrt(pow(y, 2) + pow(z, 2))) * 180 / Pi;
data1 = char(int(roll)&0xff);
data2 = char(int(pitch)&0xff);
//Sent Data by Bluetooth
BTSerial.write(char(int(roll)&0xff));
BTSerial.write(char(int(pitch)&0xff));
delay(200);
}

Slave (ตัวรับ)
#include <SoftwareSerial.h>
#include <Servo.h>
SoftwareSerial BT(2,3);
Servo myservo;
int data1,data2,val ;
int DIRA = 12;
int PWMA = 10;
void setup()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
Serial.begin(9600);
BT.begin(9600);
myservo.attach(9);
pinMode(DIRA, OUTPUT);
pinMode(PWMA, OUTPUT);
}

void loop()
{
while(BT.available()<=0){} data1 = char(BT.read());
while(BT.available()<=0){} data2 = char(BT.read());
val = data1;
val = map(data1, -40, 40, 50, 130);
if (val > 130) val = 130;
if (val < 50) val = 50;
if (data1 > 10){
myservo.write(val);
}
else if (data1 < -10){
myservo.write(val);
}
else
myservo.write(90);

if ((data2 < 20)&&(data2 > -20)){
digitalWrite(DIRA,HIGH);
analogWrite(PWMA,0);
}

else if (data2 > 20){
digitalWrite(DIRA,HIGH);
analogWrite(PWMA,255);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
    else {  
digitalWrite(DIRA,LOW);  
analogWrite(PWMA,255);  
}  
delay(100);  
}
```

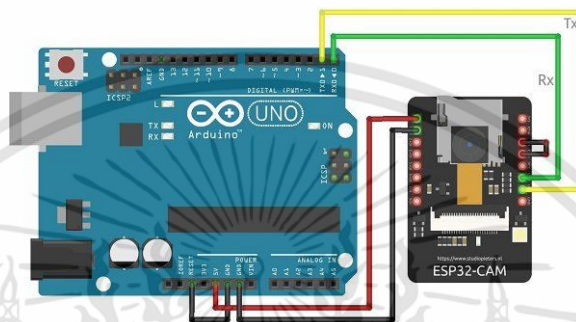


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## ข.2 การเชื่อมต่อ ESP32 Camera เพื่ออัปโหลดโค้ด

จะมีการเชื่อมต่อตามรูปภาพ ข.2 เพื่อป้อนไฟให้กับบอร์ด ESP32 Camera เพื่อใช้งาน และต่อ PIN Tx Rx ในการรับส่งข้อมูลระหว่าง คอมพิวเตอร์ผ่านบอร์ด Arduino UNO ไปยัง บอร์ด ESP32 Camera



ข.2 การเชื่อมต่อ ESP32 Camera เข้ากับ Arduino UNO

1. เมื่อทำการติดตั้งเสร็จแล้ว ให้เปิดโปรแกรม Arduino IDE ขึ้นมา

sketch\_nov13a | Arduino 1.8.14 Hourly Build 2021/01/29 11:33

File Edit Sketch Tools Help

```

Open
sketch_nov13a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

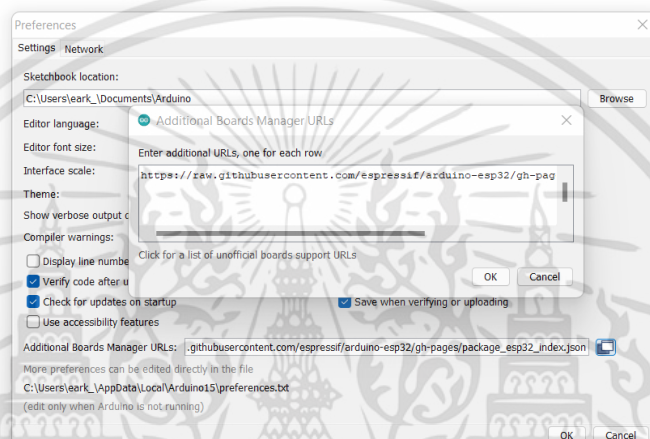
```

ข.3 ทำการเปิดโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

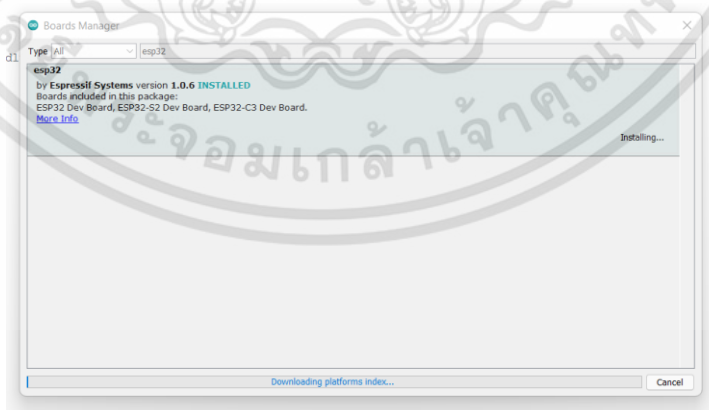
2. ไปที่ File > Preferences > Additional Boards Manager URLs แล้วเพิ่ม URL ของไฟล์ JSON ของบอร์ด ESP32 เข้าไปเพื่อใช้ในการติดตั้งการจัดการบอร์ด ESP32 Camera ในโปรแกรม Arduino IDE

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)



ข.4 การเพิ่ม URL ของไฟล์ JSON บอร์ด ESP32 Camera

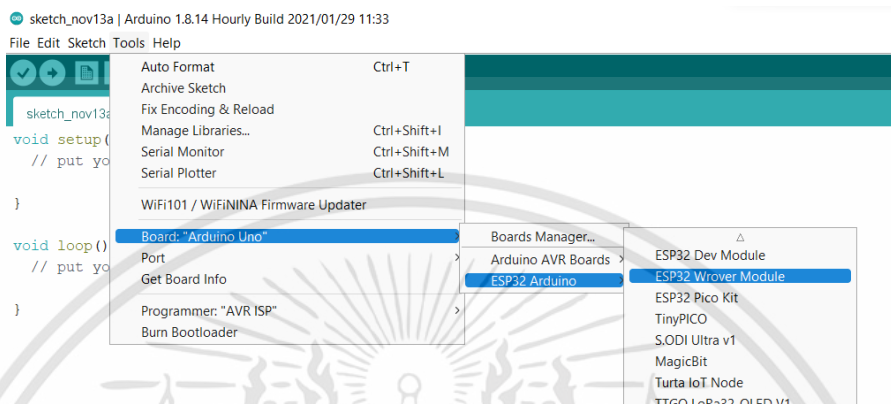
3. ทำการติดตั้งบอร์ด ESP32 ในโปรแกรม Arduino IDE ไปที่ Tools > Board > Boards Manager.



ข.5 ติดตั้งบอร์ด ESP32 ในโปรแกรม Arduino IDE

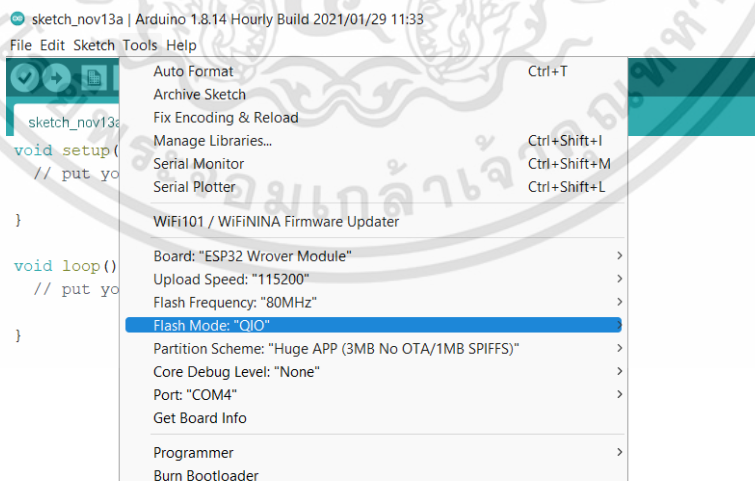
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการเลือกการตั้งค่า บอร์ดที่ต้องการอัปโหลดโค้ดไปที่ Tools > Board: > ESP32 Arduino > ESP32 Wrover Module



#### ข.6 เลือกบอร์ด ESP32 Arduino เพื่อใช้งาน

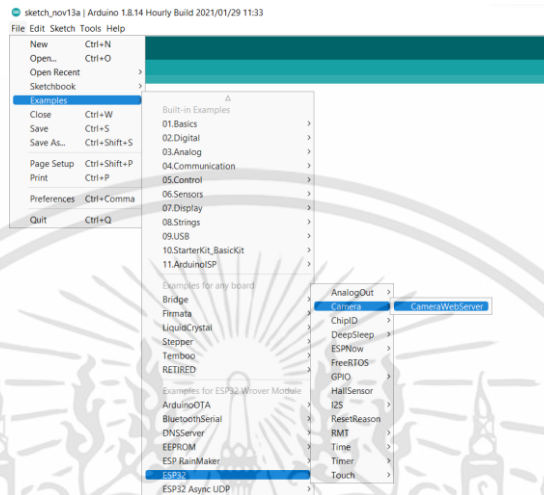
5. เมื่อเลือกบอร์ดแล้ว ทำการตั้งค่าดังนี้
- Upload Speed ที่ 115200
  - Flash Frequency ที่ 80 MHz
  - Partition Scheme เป็น Huge APP (3MB No OTA/1MB SPIFFS)
  - Core Debug Level เป็น None
  - Port (ตามการเสียบ USB ของเครื่อง) เป็น COM4



#### ข.7 การตั้งค่า ESP32 เพื่อการ Upload code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อตั้งค่าบอร์ด ESP32 เสร็จแล้ว ไปที่ File > Example > ESP32 > Camera > CameraWebServer



ข.8 เลือก โค้ด จาก Examples

7. ทำการแก้ไขโค้ด ดังนี้

- ทำการเลือกใช้ Model camera CAMERA\_MODEL\_AI\_THINKER
- ทำการแก้ไข SSID และ พาสเวิร์ด

 A screenshot of the Arduino IDE showing the code for CameraWebServer.cpp. The code includes comments and preprocessor directives for selecting a camera model. The model CAMERA\_MODEL\_AI\_THINKER is selected. There are also lines for defining SSID and password constants.
 

```

CameraWebServer.cpp | Arduino 1.8.14 Hourly Build 2021/01/29 11:33
File Edit Sketch Tools Help
CameraWebServer $
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
// Ensure ESE32 Wrover Module or other board with PSRAM is selected
// Partial images will be transmitted if image exceeds buffer size
//
// Select camera model
// #define CAMERA_MODEL_WROVER_KIT // Has PSRAM
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
// #define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version 2 Has PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
// #define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
// #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "*****";

void startCameraServer();
  
```

ข.9 แก้ไขโค้ดเพื่อให้สามารถใช้งาน Wi-Fi ตาม SSID ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8. ทำการ Compile โค้ด

```
// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#include "camera_pins.h"
#define PHOTO 14 //ESP32 CAM 1
```

```
Uploading...
Writing at 0x00140000... (60 %)
Writing at 0x00144000... (60 %)
Writing at 0x00148000... (61 %)
Writing at 0x0014c000... (62 %)
Writing at 0x00150000... (63 %)
Writing at 0x00154000... (64 %)
Writing at 0x00158000... (64 %)
Writing at 0x0015c000... (65 %)
Writing at 0x00160000... (66 %)
Writing at 0x00164000... (67 %)
Writing at 0x00168000... (68 %)
Writing at 0x0016c000... (68 %)
Writing at 0x00170000... (69 %)
Writing at 0x00174000... (70 %)
Writing at 0x00178000... (71 %)
Writing at 0x0017c000... (71 %)
Writing at 0x00180000... (72 %)
Writing at 0x00184000... (73 %)
Writing at 0x00188000... (74 %)
Writing at 0x0018c000... (75 %)
Writing at 0x00190000... (75 %)
Writing at 0x00194000... (76 %)
Writing at 0x00198000... (77 %)
Writing at 0x0019c000... (78 %)
Writing at 0x001a0000... (78 %)
Writing at 0x001a4000... (79 %)
Writing at 0x001a8000... (80 %)
Writing at 0x001ac000... (81 %)
```

ข.10 Compile โค้ด

## 9. จากนั้นจะได้ ไอพี ในการเข้าไปใช้งาน เว็บเบราว์เซอร์ ผ่าน Serial Monitor

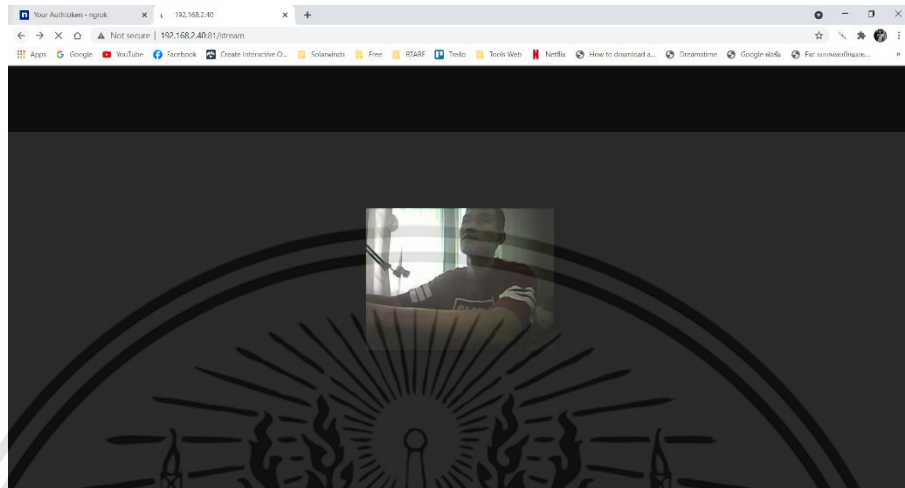
```
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.2.40' to connect
[D][WiFiGeneric.cpp:374] _eventCallback(): Event: 5 - STA_DISCONNECTED
[W][WiFiGeneric.cpp:391] _eventCallback(): Reason: 8 - ASSOC_LEAVE
E (7102) wifi:adba response cb: sta bss deleted
[D][WiFiGeneric.cpp:374] _eventCallback(): Event: 4 - STA_CONNECTED
[D][WiFiGeneric.cpp:374] _eventCallback(): Event: 7 - STA_GOT_IP
[D][WiFiGeneric.cpp:419] _eventCallback(): STA IP: 192.168.2.40, MASK: 255.255.255.0, GW: 192.168.2.1
```

Autoscroll  Show timestamp Newline  115200 baud  Clear

ข.11 ESP32 Camera เชื่อมต่อ Wi-Fi สำเร็จและแสดง ไอพีแอดเดรส ของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

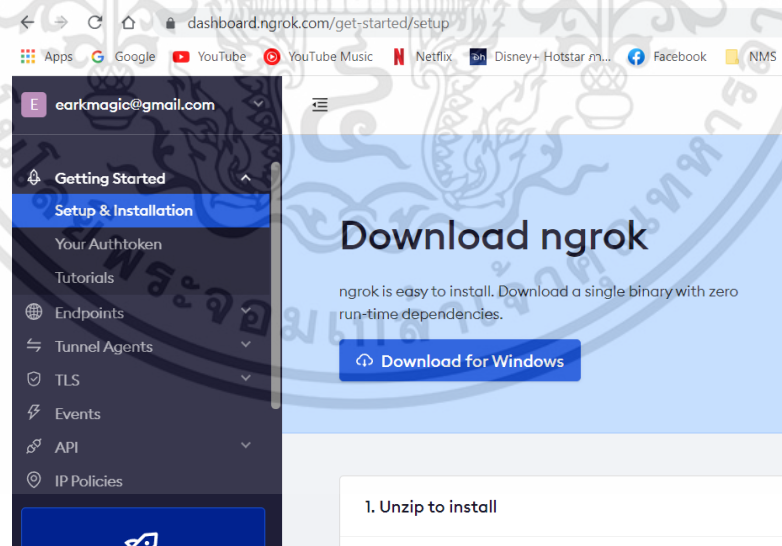
## 10. ทำการทดสอบ บอร์ด ESP32 Camera ที่พร้อมใช้งาน ผ่านเว็บเบราว์เซอร์



ข.12 ทดสอบการใช้งาน

## ข.3 การใช้งาน ESP32 Camera ร่วมกับ Ngrok Server บนคอมพิวเตอร์ระบบปฏิบัติการ Windows

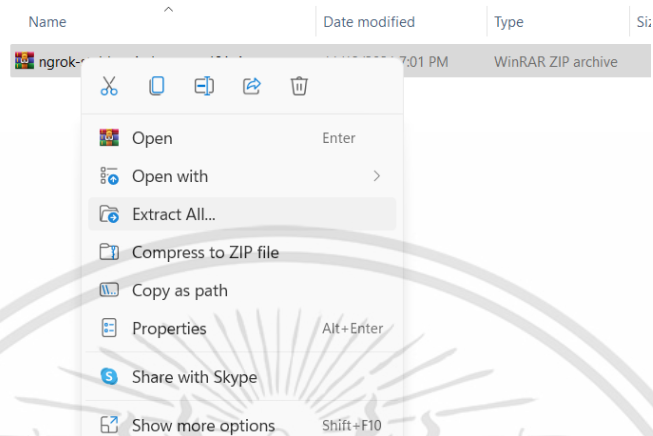
### 1. ทำการดาวน์โหลด Ngrok ผ่าน <https://dashboard.ngrok.com/>



ข.13 Download Ngrok for Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. Extract File Ngrok



### ข.14 Extract File Ngrok

3. เปิด Command Prompt แล้วไปยัง Directory ที่ทำการ Extract File ด้วย Command `cd C:` เพื่อไปยัง Directory C และ `cd Users\eark\Download\ngrok\ngrok-stable-windows-amd64` เพื่อไปยัง Folder ที่ Extract file

```

C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64\ngrok.exe
EXAMPLES:
  ngrok http 80 # secure public URL for port 80 web server
  ngrok http -subdomain=baz 8080 # port 8080 available at baz.ngrok.io
  ngrok http foo.dev:80 # tunnel to host:port instead of localhost
  ngrok http https://localhost # expose a local https server
  ngrok tcp 22 # tunnel arbitrary TCP traffic to port 22
  ngrok tls -hostname=foo.com 443 # TLS traffic for foo.com to port 443
  ngrok start foo bar baz # start tunnels from the configuration file

VERSION:
  2.3.39

AUTHOR:
  inconshreveable - <alan@ngrok.com>

COMMANDS:
  authtoken save authtoken to configuration file
  credits prints author and licensing information
  http start an HTTP tunnel
  start start tunnels by name from the configuration file
  tcp start a TCP tunnel
  tls start a TLS tunnel
  update update ngrok to the latest version
  version print the version string
  help Shows a list of commands or help for one command

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64>
  
```

4. เปิดใช้งาน Ngrok Client และ Connect Account ด้วย Command Ngrok `authtoken` ตามด้วย Auth token จากหน้าเว็บของ Ngrok ที่เราใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64\ngrok.exe
ngrok http -subdomain=baz 8080 # port 8080 available at baz.ngrok.io
ngrok http foo.dev:80 # tunnel to host:port instead of localhost
ngrok http https://localhost # expose a local https server
ngrok tcp 22 # tunnel arbitrary TCP traffic to port 22
ngrok tls -hostname=foo.com 443 # TLS traffic for foo.com to port 443
ngrok start foo bar baz # start tunnels from the configuration file

VERSION:
2.3.39

AUTHOR:
inconshreveable - <alan@ngrok.com>

COMMANDS:
authtoken save authtoken to configuration file
credits prints author and licensing information
http start an HTTP tunnel
start start tunnels by name from the configuration file
tcp start a TCP tunnel
tls start a TLS tunnel
update update ngrok to the latest version
version print the version string
help Shows a list of commands or help for one command

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64>ngrok authtoken 1s4KNDJ1lyqv0K3hGj5tE85qdHP_5K7k8mh4u8CLumQcYKDMW
Authtoken saved to configuration file: C:\Users\eark\.ngrok2\ngrok.yml
C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64>

```

### ข.16 ยืนยันตัวตนการใช้งาน Ngrok

5. ทำการเปิด Tunnel การเข้าถึง URL ของกล้อง ESP32 Camera ด้วย Command Ngrok http http://192.168.2.40:81 เพื่อทำการ Forward URL เมื่อเข้ามาจาก URL ที่จะได้รับจาก Ngrok Server

```

C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64\ngrok.exe
C:\Users\eark\Downloads\ngrok\ngrok-stable-windows-amd64>ngrok http http://192.168.2.40:81

```

### ข.17 ทำการ Forward URL Ngrok

6. จากนั้นการทำงานของ Ngrok ผ่าน Command Prompt จะแสดง URL ที่จะใช้งานดังภาพ ข.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

C:\Users\eark_\Downloads\ngrok\ngrok-stable-windows-amd64\ngrok.exe - ngrok http http://192.168.2.40:81
ngrok by @inconshreveable
Session Status      online
Account             earkmagic@gmail.com (Plan: Free)
Update              update available (version 2.3.40, Ctrl-U to update)
Version             2.3.39
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://a3c16cb8f916.ngrok.io -> http://192.168.2.40:81
                    https://a3c16cb8f916.ngrok.io -> http://192.168.2.40:81
Connections
  ttl   opn   rt1   rt5   p50   p90
   0     0    0.00  0.00  0.00  0.00

```

ข.18 Session Status Ngrok เมื่อ Forwarding สำเร็จ

## 7. ทำการทดสอบการใช้งาน URL ที่ได้รับ ผ่าน Web browser

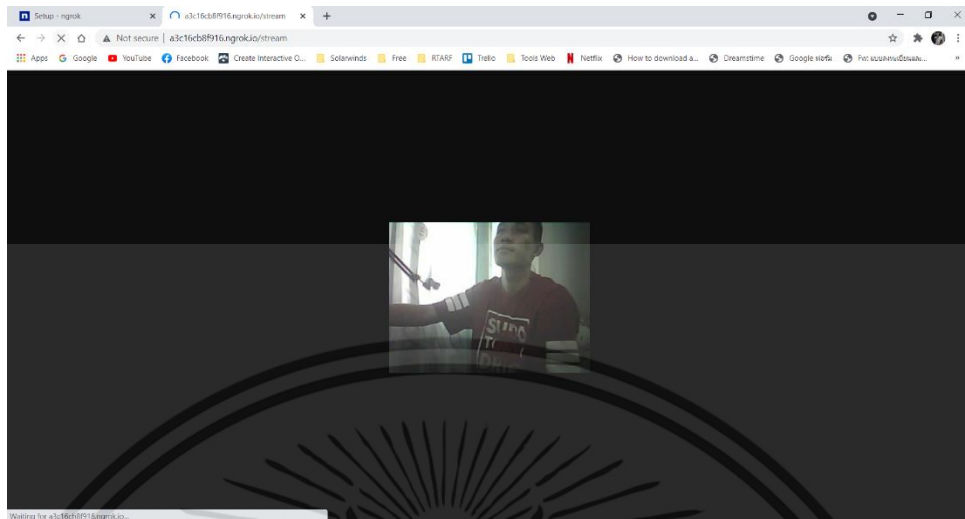
```

C:\Users\eark_\Downloads\ngrok\ngrok-stable-windows-amd64\ngrok.exe - ngrok http http://192.168.2.40:81
ngrok by @inconshreveable
Session Status      online
Account             earkmagic@gmail.com (Plan: Free)
Update              update available (version 2.3.40, Ctrl-U to update)
Version             2.3.39
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://a3c16cb8f916.ngrok.io -> http://192.168.2.40:81
                    https://a3c16cb8f916.ngrok.io -> http://192.168.2.40:81
Connections
  ttl   opn   rt1   rt5   p50   p90
  21    1    0.03  0.04  0.75  90.52
HTTP Requests
-----
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK
GET /stream          200 OK

```

ข.19 Session status เมื่อมีการเรียกใช้งาน URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

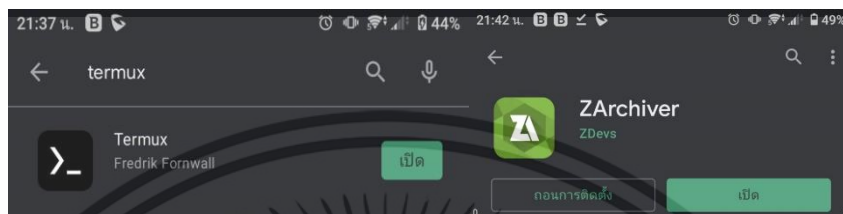


ข.20 ภาพการทดสอบ URL จาก Ngrok Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

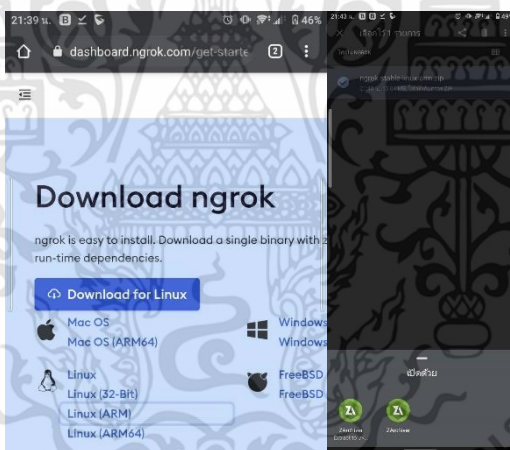
#### ข.4 การใช้งาน Ngrok ผ่านโทรศัพท์มือถือ Android

1. ติดตั้ง Application Termux เพื่อใช้ CLI Linux และ ติดตั้ง Application ZArchiver เพื่อใช้แตกไฟล์ ZIP



ข.21 Application Termux และ ZArchiver

2. ดาวน์โหลด Ngrok client สำหรับ OS Linux ARM จากนั้น ใช้ Application ZArchiver เพื่อใช้แตกไฟล์



ข.22 ดาวน์โหลด Ngrok สำหรับ OS Linux ARM และแตกไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิด CLI Termux จากนั้นทำการย้าย File ที่ได้จากข้อ 2 มาไว้ที่ Path HOME ของ Termux เพื่อง่ายต่อการเรียกใช้งาน โดยใช้คำสั่ง `mv -v /sdcard/Ngrok /$HOME`

```

22:29 น. @ B >- 43%
Welcome to Termux!
Wiki: https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat: https://gitter.im/termux/termux
IRC channel: #termux on freenode

Working with packages:
* Search packages: pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages: pkg upgrade

Subscribing to additional repositories:
* Root: pkg install root-repo
* Unstable: pkg install unstable-repo
* X11: pkg install x11-repo

Report issues at https://termux.com/issues
$ cd /sdcard/Ngrok/ngrok-stable-linux-arm64
$ ls
$ mv -v /sdcard/Ngrok/ngrok-stable-linux-arm64/ngrok /$HOME
copied '/sdcard/Ngrok/ngrok-stable-linux-arm64/ngrok' ->
'//data/data/com.termux/files/home/ngrok'
removed '/sdcard/Ngrok/ngrok-stable-linux-arm64/ngrok'
$

```

ข.23 การย้าย File Ngrok มาไว้ที่ Path HOME

4. เปิด Cli session ใหม่ เพื่อกลับมาที่ ./Home และเช็คการย้ายว่ามี File Ngrok มาหรือยัง โดยใช้คำสั่ง `ls` จากนั้นจัดการสิทธิ์ในการเข้าถึงในระดับ Administrator โดยใช้คำสั่ง `chmod +x ngrok-stable-linux-arm`

```

22:29 น. @ B >- 42%
Welcome to Termux!
Wiki: https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat: https://gitter.im/termux/termux
IRC channel: #termux on freenode

Working with packages:
* Search packages: pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages: pkg upgrade

Subscribing to additional repositories:
* Root: pkg install root-repo
* Unstable: pkg install unstable-repo
* X11: pkg install x11-repo

Report issues at https://termux.com/issues
$ ls
$ ngrok ngrok-stable-linux-arm
$ chmod +x ngrok
$ ls
$ ngrok ngrok-stable-linux-arm
$

```

ข.24 จัดการสิทธิ์ในการเข้าถึงในระดับ Administrator

5. เปิดใช้งาน Ngrok Client และ Connect Account โดยใช้คำสั่ง `./ngrok authtoken` ตามด้วย Auth Token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

22:15 น. B > B
Welcome to Termux!

Wiki: https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat: https://gitter.im/termux/termux
IRC channel: #termux on freenode

Working with packages:

* Search packages: pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages: pkg upgrade

Subscribing to additional repositories:

* Root: pkg install root-repo
* Unstable: pkg install unstable-repo
* X11: pkg install x11-repo

Report issues at https://termux.com/issues

$ ls
ngrok-stable-linux-arm
$ cd ngrok-stable-linux-arm/
$ ls
ngrok
$ chmod +x ngrok
$ ls
ngrok
$ ./ngrok authtoken 1s4KNDJTlyqv00K3hGj5t85qdHP_5KJkBmh4
u8CLumQcYKDNW

```

ข.25 เปิดใช้งาน Ngrok Client และ Connect Account

6. ทำการ Forward IP ของ บอร์ด ESP32 Camera เข้ากับ Ngrok เพื่อการใช้งานผ่าน Domain ของ Ngrok server

```

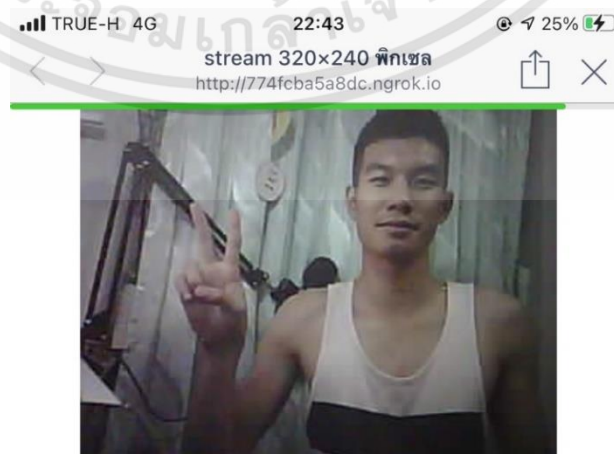
22:35 น. B > B
ngrok by @inconsreveable (Ctrl+C to quit)

Session Status      online
Account             earkmagic@gmail.com (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://774fcb5a8dc.ngrok.io -> http://192.168.2.42:81
                    https://774fcb5a8dc.ngrok.io -> http://192.168.2.42:81
Connections
  ttl    opn    rt1    rt5    p50    p90
   0     0     0.00  0.00  0.00  0.00

```

ข.26 เปิดใช้งาน Ngrok เพื่อการใช้งาน Domain

7. ทดสอบการใช้งาน ผ่าน Internet 4G True



ข.27 ทดสอบ URL จาก Smartphone ผ่าน 4G True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้