

เครื่องดูดฝุ่น

VACUUM CLEANER

ณัฐนรี ศีลารักษ์

Nutnaree Silaruk

ศศิพัชร์ พิสิฐวานนท์

Sasipatch Phisitvanont

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

เครื่องดูดฝุ่น

VACUUM CLEANER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องดูดฝุ่น

VACUUM CLEANER

ผู้จัดทำ	นางสาวณัฐรี ศิลารักษ์	รหัสประจำตัว	58010395
	นางสาวศศิพัชร พิสิฐวานนท์	รหัสประจำตัว	58011202

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

  
ผศ.ดร.ยุธนา คิดใจเดียว

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องดูดฝุ่น	
นักศึกษา	นางสาวณัฐนรี ศิลารักษ์	รหัสประจำตัว 58010395
	นางสาวศศิพัทธ์ พิสิฐวานนท์	รหัสประจำตัว 58011202
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2561	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.ยุพธนา คิดใจเดียว	

### บทคัดย่อ

โครงการนี้นำเสนอการออกแบบและการสร้างเครื่องดูดฝุ่น โดยหลักการทำงานเบื้องต้นของเครื่องดูดฝุ่นนั้น จะแบ่งเป็น 2 ระบบ คือ ระบบอัตโนมัติและระบบแมนนวล ซึ่งหลักการทำงานของระบบอัตโนมัตินั้นจะมีเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก 3 ตัวติดตั้งรอบๆเครื่องดูดฝุ่น เพื่อวัดระยะทางระหว่างเครื่องดูดฝุ่นกับสิ่งกีดขวาง ซึ่งต่ออยู่กับบอร์ดไมโครคอนโทรลเลอร์ โดยมีการเขียนชุดคำสั่งผ่านบอร์ด ไมโครคอนโทรลเลอร์ คอยควบคุมทิศทางของมอเตอร์กระแสตรง และควบคุมความเร็วของมอเตอร์จากค่า พัลส์วิดท์มอดูเลชัน รวมถึงกำหนดให้มอเตอร์เลี้ยวหนีเมื่อตรวจเจอสิ่งกีดขวาง เพื่อหลีกเลี่ยงการชนและสร้างความเสียหายให้ให้แก่ตัวเครื่องดูดฝุ่น ส่วนระบบแมนนวลนั้นจะมีรีโมทคอยควบคุมทิศทางเครื่องดูดฝุ่นผ่านทางระบบ บลูทูธ ซึ่งเครื่องดูดฝุ่นนี้ประกอบด้วยล้อทั้งหมด 3 ล้อ โดยมี 1 ล้อหน้าทำหน้าที่เป็นล้ออิสระ ส่วนอีก 2 ล้อทำหน้าที่ขับเคลื่อนเครื่องดูดฝุ่นให้ทำการเลี้ยวซ้ายหรือเลี้ยวขวา นอกจากนี้เครื่องดูดฝุ่นยังประกอบด้วยมอเตอร์กระแสตรงอีก 3 ตัว โดยมอเตอร์ 1 ตัว จะต่ออยู่กับแปรงปัดหลัก อีก 2 ตัวจะต่ออยู่กับล้อ

Thesis Title	Vacuum cleaner	
Student	Ms.Nutnaree Silaruk	Student ID 58010395
	Ms.Sasipatch Phisitvanont	Student ID 58011202
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2018	
Thesis Advisor	Asst. Prof. Dr.Yuttana Kitjaidure	

## ABSTRACT

This project presents the design and construction of a vacuum cleaner. The basic working principle of the vacuum cleaner is divided into 2 systems: automatic and manual systems. The working principle of the automation system is equipped with 3 ultrasonic wave measuring sensors attached around the vacuum cleaner to measure the distance between the vacuum cleaner and the obstacles. The ultrasonic wave measuring sensors are connected to the Microcontroller which commands the direction and speed of the DC motor by PWM. The Microcontroller also monitors the vacuum cleaner to turn away when detecting obstacles to avoid collisions and damage. The manual system remotely controls the vacuum via the Bluetooth system. This vacuum cleaner consists of all 3 wheels including one front wheel and two rear wheels. The front wheel acts as an independent wheel while the other 2 wheels control the vacuum cleaner to turn left or right. In addition, the vacuum cleaner is composed of 3 DC motors. One motor is connected to the main brush while the other two motors are connected to the wheels

## กิตติกรรมประกาศ

ในการจัดทำโครงการเครื่องดูดฝุ่น ข้าพเจ้าขอขอบพระคุณท่านอาจารย์ที่ปรึกษา ผศ.ดร. ยุทธนา คิดใจเดียว ที่คอยชี้แนะให้ความรู้ ให้คำแนะนำและช่วยแก้ไขปัญหาต่างๆ รวมถึงสถานที่และอุปกรณ์ที่ใช้ในการทำโครงการ อีกทั้งคณะอาจารย์ทุกท่านที่เกี่ยวข้องในการให้ความรู้แก่คณะผู้จัดทำ และนำความรู้นั้นมาประยุกต์ใช้ในโครงการ

ขอขอบพระคุณบิดามารดาที่ช่วยเหลือในเรื่องค่าใช้จ่ายต่างๆ อีกทั้งยังคอยสนับสนุนและเป็นกำลังใจให้เสมอรวมทั้งรุ่นพี่และเพื่อนๆ วิศวกรรมอิเล็กทรอนิกส์ที่ให้ยืมอุปกรณ์ รวมถึงให้คำแนะนำดีๆ ทำให้คณะผู้จัดทำสามารถทำโครงการนี้ให้สำเร็จลุล่วง

สุดท้ายนี้คณะผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจและผู้นำผลงานนี้ไปใช้งานได้

ณัฐรี ศีลารักษ์

ศศิพัชร์ พิสิฐวานนท์



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ขอบเขตการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 งานวิจัยที่เกี่ยวข้อง.....	3
2.1 หลักการเบื้องต้น.....	3
2.2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.2.1 มอเตอร์ไฟฟ้ากระแสตรง.....	3
2.2.2 เซนเซอร์วัดระยะทาง.....	11
2.2.3 ไมโครคอนโทรลเลอร์.....	15
2.2.4 บลูทูธ.....	16
2.2.4 หลักการดูดฝุ่นของเครื่องดูดฝุ่น.....	16
2.3 ทฤษฎีอุปกรณ์.....	18
2.3.1 มอเตอร์เกียร์กระแสตรง.....	18
2.3.2 มอเตอร์กระแสตรง.....	19
2.3.3 ล้อ.....	19
2.3.4 พัดลมดูดอากาศ.....	20

# สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีดำเนินการวิจัย.....	21
3.1 ศึกษาหาข้อมูล.....	21
3.2 ขั้นตอนดำเนินงาน.....	21
3.2.1 วางแผนการทำเครื่องดูดฝุ่น.....	21
3.2.2 ออกแบบการทำงานของเครื่องดูดฝุ่น.....	22
3.2.3 ออกแบบโครงของเครื่องดูดฝุ่น.....	23
3.2.4 ประกอบเครื่องดูดฝุ่น.....	24
3.2.5 การสร้างแอปพลิเคชัน.....	25
บทที่ 4 การทดลอง และผลการทดลอง.....	28
4.1 การวัดความสามารถในการดูดฝุ่น.....	28
4.1.1 วิธีการทดลอง.....	28
4.1.2 ผลการทดลอง.....	28
4.2 ทดสอบการรับ – ส่งสัญญาณของ เซนเซอร์อัลตราโซนิก.....	30
4.2.1 วิธีการทดลอง.....	30
4.2.2 ผลการทดลอง.....	31
4.3 ทดสอบการรับ – ส่งสัญญาณของ บลูทูธ.....	41
4.3.1 วิธีการทดลอง.....	41
4.3.2 ผลการทดลอง.....	41
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ.....	43
5.1 สรุปผลการทดลอง.....	43
5.2 ปัญหาและอุปสรรค.....	43
5.3 ข้อเสนอแนะ.....	43
เอกสารอ้างอิง.....	44
ภาคผนวก.....	45
อุปกรณ์สำหรับทำเครื่องดูดฝุ่น.....	46
คำสั่งที่ใช้ในการควบคุม.....	46

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# สารบัญตาราง

ตารางที่	หน้า
4.1 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 1 .....	40
4.2 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 2 .....	40
4.3 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 3 .....	40



# สารบัญรูป

รูปที่	หน้า
2.1 ส่วนที่อยู่กับที่หรือสเตเตอร์.....	4
2.2 ขดลวดพันอยู่รอบขั้วแม่เหล็ก .....	4
2.3 ลักษณะของขั้วแม่เหล็ก .....	4
2.4 โรเตอร์.....	5
2.5 แปรรงถ่าน.....	6
2.6 ช่องแปรรงถ่าน .....	6
2.7 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม .....	7
2.8 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน.....	7
2.9 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบซอร์ทซ์ชั้นท์คอมปาร์ตมอเตอร์ .....	8
2.10 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบลองชั้นท์คอมปาร์ตมอเตอร์.....	8
2.11 Pulse Width Modulation.....	10
2.12 วงจร H-Bridge .....	11
2.13 ชุดขับกระแส L298N .....	11
2.14 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง .....	12
2.15 การทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก .....	13
2.16 กราฟแสดงค่าอุณหภูมิเฉลี่ยของประเทศไทยในแต่ละภาค.....	14
2.17 ไดอะแกรมการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก.....	14
2.18 Arduino MEGA 2576 R3.....	15
2.19 บลูทูธโมดูล HC-05.....	17
2.20 การดูตของเครื่องดูตฝุ่น .....	17
2.21 มอเตอร์เกียร์กระแสตรง .....	18
2.22 มอเตอร์กระแสตรง .....	19
2.23 ล้อ .....	19
2.24 พัดลมดูดอากาศ .....	20
3.1 บล็อกไดอะแกรมแสดงขั้นตอนการดำเนินงาน .....	21
3.2 บล็อกไดอะแกรมแสดงขั้นตอนการทำงานของเครื่องดูตฝุ่นระบบอัตโนมัติ.....	22

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.3 บล็อกไดอะแกรมแสดงขั้นตอนการทำงานของเครื่องดูดฝุ่นระบบแมนนวล .....	22
3.4 โครงเครื่องดูดฝุ่นส่วนบน .....	23
3.5 โครงเครื่องดูดฝุ่นส่วนล่าง .....	23
3.6 เครื่องดูดฝุ่น .....	24
3.7 แสดงภาพรวมวงจรของเครื่องดูดฝุ่น .....	24
3.8 หน้าต่างทำงานของแอปพลิเคชัน Bcontrol .....	25
3.9 บล็อกโปรแกรมของแอปพลิเคชัน .....	27
4.1 เศษกระดาษบนพื้นก่อนการดูด .....	24
4.2 เศษกระดาษบนพื้นหลังการดูด .....	25
4.3 เศษกระดาษในกล่องเก็บฝุ่น .....	27
4.4 วงจรสำหรับการทดสอบการรับส่งสัญญาณเซนเซอร์อัลตราโซนิก .....	30
4.5 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 5 เซนติเมตร .....	31
4.6 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 5 เซนติเมตร ..	31
4.7 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 10 เซนติเมตร .....	32
4.8 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 10 เซนติเมตร	32
4.9 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 15 เซนติเมตร .....	33
4.10 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 15 เซนติเมตร	33
4.11 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 5 เซนติเมตร .....	34
4.12 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 5 เซนติเมตร	34
4.13 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 10 เซนติเมตร .....	35
4.14 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 10 เซนติเมตร	35
4.15 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 15 เซนติเมตร .....	36
4.16 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 15 เซนติเมตร	36
4.17 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 5 เซนติเมตร .....	37
4.18 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 5 เซนติเมตร	37
4.19 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 10 เซนติเมตร .....	38

## สารบัญญรูป (ต่อ)

รูปที่	หน้า
4.20 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 10 เซนติเมตร	38
4.21 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 15 เซนติเมตร .....	39
4.22 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 15 เซนติเมตร	39
4.23 หน้าต่างรายชื่อบลูทูธ .....	41
4.24 Serial monitor .....	42



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องมาจากการเติบโตทางด้านเศรษฐกิจของสังคมเมือง และการเปิดรับวัฒนธรรมและเทคโนโลยีต่างๆของต่างประเทศ ทำให้กรุงเทพมหานครกลายเป็น ศูนย์กลางในหลายๆด้าน อาทิ ศูนย์กลางทางการสื่อสาร ศูนย์กลางทางการศึกษา เป็นต้น ผู้คนที่อยู่ในต่างจังหวัด ต่างก็หลั่งไหลเข้ามาทำงานในกรุงเทพ ส่งผลให้ผู้คนในสังคมเมืองต่างต้องมีวิถีชีวิตที่เร่งรีบ วุ่นวาย ต้องรับผิดชอบหน้าที่หลายอย่างในแต่ละวัน หรือบางบ้านที่มีแต่ผู้สูงอายุ ทำให้มองข้าม หรือเหนียวล่า ในการจัดการกับการทำความสะอาดที่อยู่อาศัยซึ่งเป็นสิ่งสำคัญมาก เพราะถ้าละเลยแล้วจะทำให้เกิดสิ่งสกปรกเป็นที่สะสมของเชื้อโรคอันก่อให้เกิดอันตรายต่อสุขภาพของคนในบ้านได้ โดยเฉพาะฝุ่นซึ่งส่งผลโดยตรงกับระบบทางเดินหายใจ ดังนั้นเพื่อให้สอดคล้องกับวิถีชีวิตของคนในสังคมเมือง และเพื่อเพิ่มความสะดวกสบาย กลุ่มของข้าพเจ้าจึงได้ออกแบบและสร้างเครื่องดูดฝุ่นที่สามารถทำงานได้ทั้งระบบอัตโนมัติและระบบแมนนวล เพื่อเพิ่มประสิทธิภาพในการทำความสะอาดบ้าน และช่วยให้ประหยัดแรงและลดเวลาในการทำความสะอาดอีกด้วย

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อนำไปสร้างเครื่องดูดฝุ่นให้ช่วยผ่อนแรงและย่นระยะเวลาในการทำความสะอาดที่อยู่อาศัย เพื่อให้เกิดความสะดวกและรวดเร็ว ให้กับกลุ่มคนที่ไม่ค่อยมีเวลาหรือไม่ชื่นชอบทำความสะอาด รวมทั้งกลุ่มคนที่ไม่สามารถทำความสะอาดเองได้เช่น ผู้สูงอายุ ผู้พิการ เป็นต้น

### 1.3 สมมุติฐานของการศึกษา

สร้างเครื่องดูดฝุ่น ที่สามารถทำงานได้อัตโนมัติ โดยจะสามารถเคลื่อนที่ เดินหน้า ถอยหลัง เลี้ยวซ้ายเลี้ยวขวา ตรวจสอบและหลบสิ่งกีดขวางเองได้ รวมถึงสามารถทำงานได้โดยการควบคุมทิศทางเครื่องดูดฝุ่นจากระยะไกลผ่านระบบ Bluetooth ให้เคลื่อนที่ตามต้องการของผู้ใช้งาน

#### 1.4 ขอบเขตการวิจัย

-สามารถทำให้เครื่องดูดฝุ่นเคลื่อนที่เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา และหลบสิ่งกีดขวางได้อัตโนมัติ

-สามารถควบคุมให้เครื่องดูดฝุ่นเคลื่อนที่เดินหน้า ถอยหลัง เลี้ยวซ้ายเลี้ยวขวา และหลบสิ่งกีดขวางได้โดยการควบคุมจากรีโมทผ่านระบบบลูทูธ

-มีแบตเตอรี่ในตัวสามารถทำงานได้โดยไม่ต้องเสียบปลั๊ก

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

-ได้เครื่องดูดฝุ่นที่สามารถนำไปใช้งานได้จริง

-สามารถนำความรู้ทางด้านอิเล็กทรอนิกส์ไปใช้ในการสร้างชิ้นงานได้



## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

#### 2.1 หลักการเบื้องต้น

หลักการเบื้องต้นของเครื่องดูดฝุ่นนั้นจะแบ่งเป็น 2 ระบบ คือ 1. ระบบอัตโนมัติ จะถูกควบคุมด้วย ไมโครคอนโทรลเลอร์ ให้เคลื่อนที่ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ได้อัตโนมัติ และจะมีเซ็นเซอร์วัดระยะทางด้วยอัลตราโซนิก คอยวัดระยะทางเพื่อให้เครื่องดูดฝุ่นสามารถหลบเลี่ยงเมื่อเข้าใกล้สิ่งกีดขวางเองได้

2. ระบบแมนนวล ถูกควบคุมโดย ไมโครคอนโทรลเลอร์ เช่นกัน แต่จะถูกสั่งให้ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ด้วยรีโมท ผ่านระบบบลูทูธ ตามที่ผู้ใช้งานต้องการ

เครื่องดูดฝุ่นนั้นมีแบตเตอรี่ในตัว คือ ถ่านไฮดราย 12V 1300mAh โดยสามารถทำงานได้โดยไม่ต้องเสียบปลั๊ก อีกทั้งยังสามารถช่วยประหยัดแรงและเวลาในการทำความสะอาดบ้านอีกด้วย

#### 2.2 ทฤษฎีที่เกี่ยวข้อง

##### 2.2.1 มอเตอร์ไฟฟ้ากระแสตรง

###### 2.2.1.1 ความหมายของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง หรือดีซีมอเตอร์ (DC Motor) เป็นอุปกรณ์ที่แปลงพลังงานไฟฟ้าให้เป็นพลังงานกล โครงสร้างภายใน มอเตอร์กระแสตรง ประกอบด้วยส่วนหลักๆ สองส่วน ได้แก่ แม่เหล็กถาวรและแกนขดลวด นอกจากนี้ยังมีแปรงถ่าน (Brush) ซึ่งเป็นส่วนเชื่อมต่อเพื่อรับพลังงานไฟฟ้าภายนอกไปยังขดลวดของมอเตอร์ เมื่อขดลวดได้รับไฟฟ้ากระแสตรง จะมีถูกเหนี่ยวนำให้เกิดสนามแม่เหล็กรอบๆ รอบขดลวด

###### 2.2.1.2 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงที่ส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

##### 1. ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator) ประกอบด้วย

-เฟรมหรือโยค (Frame Or Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาม้วนเป็นรูปทรงกระบอก

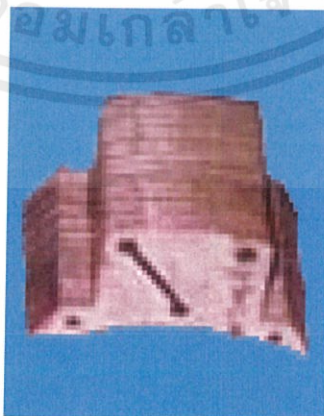


รูปที่ 2.1 ส่วนที่อยู่กับที่หรือสเตเตอร์



รูปที่ 2.2 ขดลวดพันอยู่รอบขั้วแม่เหล็ก

-ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวดส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กันด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กจากขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์มีกำลังหมุน (Torque)



รูปที่ 2.3 ลักษณะของขั้วแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สอง ขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆ แกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น

2 ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์



รูปที่ 2.4 โรเตอร์

ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานั้นจะวางอยู่บนแบร์ริง เพื่อบังคับให้หมุนอยู่ในแนวหนึ่งไม่มีการสั่นสะเทือนได้
  2. แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด (Torque)
  3. คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสายของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้ยึดแน่นติดกับแกนเพลลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่าปฏิกิริยามอเตอร์ (Motor action)
  4. ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ ควรศึกษาต่อไปในเรื่องการพันอาร์มาเจอร์ (Armature Winding) ในโอกาสต่อไป
3. แปรงถ่าน (Brushes)



รูปที่ 2.5 แปรรงถ่าน

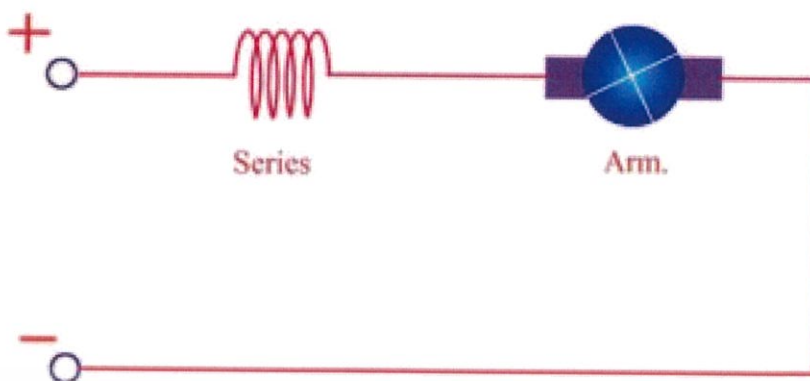


รูปที่ 2.6 ช่องแปรรงถ่าน

ทำด้วยคาร์บอนมีรูปร่างเป็นแท่งสี่เหลี่ยมผืนผ้าในของแปรรงมีสีปังกตอยู่ด้านบนเพื่อให้ถ่านนี้สัมผัสกับซีคอมมิวเตเตอร์ตลอดเวลาเพื่อรับกระแส และส่งกระแสไฟฟ้าระหว่างขดลวดอาร์มาเจอร์กับวงจรไฟฟ้าจากภายนอก คือถ้าเป็นมอเตอร์กระแสไฟฟ้าตรงจะทำหน้าที่รับกระแสจากภายนอกเข้าไปยังคอมมิวเตเตอร์ให้ลวดอาร์มาเจอร์เกิดแรงบิดทำให้มอเตอร์หมุนได้

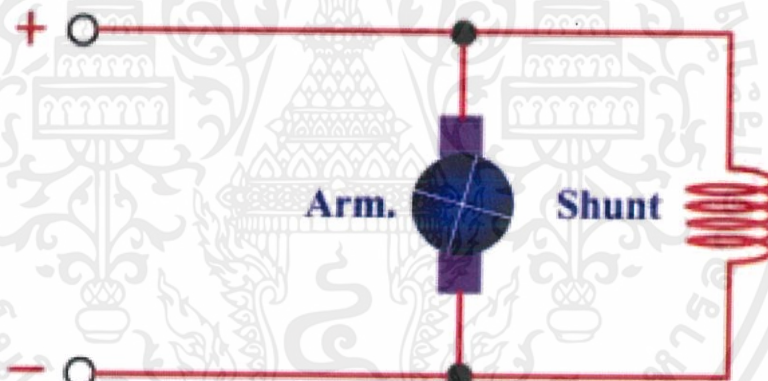
### 2.2.1.3 ชนิดของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์แบบอนุกรม (Series Motor) คือมอเตอร์ที่ต่อขดลวดสนามแม่เหล็กอนุกรมกับอาร์มาเจอร์ของมอเตอร์ชนิดนี้ว่าซีรีสฟิลด์ (Series Field) มีคุณลักษณะที่ดีคือให้แรงบิดสูงนิยมใช้เป็นต้นกำลังของรถไฟฟ้าrolleyของเครนไฟฟ้าความเร็วรอบของมอเตอร์อนุกรมเมื่อไม่มีโหลดความเร็วจะสูงมากแต่ถ้ามีโหลดมาต่อความเร็วก็จะลดลงตามโหลด โหลดมากหรือทำงานหนักความเร็วลดลง แต่ขดลวดของมอเตอร์ไม่เป็นอันตรายจากคุณสมบัตินี้จึงนิยมนำมาใช้กับเครื่องใช้ไฟฟ้าในบ้านหลายอย่าง เช่น เครื่องดูดฝุ่นเครื่องผสมอาหาร สว่านไฟฟ้าจักรเย็บผ้าเครื่องเป่าผม มอเตอร์กระแสตรงแบบอนุกรมใช้งานหนักได้ดีเมื่อใช้งานหนักกระแสจะมากความเร็วรอบจะลดลงเมื่อไม่มีโหลดมาต่อความเร็วจะสูงมากอาจเกิดอันตรายได้ดังนั้น เมื่อเริ่มสตาร์ทมอเตอร์แบบอนุกรมจึงต้องมีโหลดมาต่ออยู่เสมอ



รูปที่ 2.7 วงจรการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม

มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt Motor) หรือเรียกว่าชันทมอเตอร์ มอเตอร์แบบขนานนี้ขดลวดสนามแม่เหล็ก(Field Coil) จะต่อขนานกับขดลวด ชุดอาเมเจอร์มอเตอร์แบบขนานนี้มีคุณลักษณะมีความเร็วคงที่แรงบิดเริ่มหมุนต่ำ แต่ความเร็วรอบคงที่ชันทมอเตอร์ส่วนมากเหมาะสำหรับงานดังนี้ พัดลมเพราะพัดลมต้องการความเร็วคงที่และต้องการเปลี่ยนความเร็วได้ง่าย

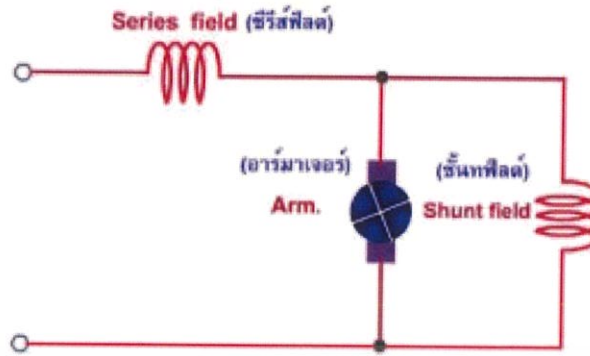


รูปที่ 2.8 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน

มอเตอร์ไฟฟ้ากระแสตรงแบบผสม (Compound Motor) หรือเรียกว่าคอมปาวด์มอเตอร์ มอเตอร์ไฟฟ้ากระแสตรงแบบผสมนี้จะนำคุณลักษณะที่ดีของมอเตอร์ไฟฟ้ากระแสตรง แบบขนาน และแบบอนุกรมมารวมกัน มอเตอร์แบบผสม มีคุณลักษณะพิเศษคือมีแรงบิดสูง (High starting torque) แต่ความเร็วรอบคงที่ ตั้งแต่ยังไม่มีโหลดจนกระทั่งมีโหลดเต็มที่มอเตอร์แบบผสมมีวิธีการต่อขดลวดขนานหรือขดลวดชั้นที่อยู่ 2 วิธี

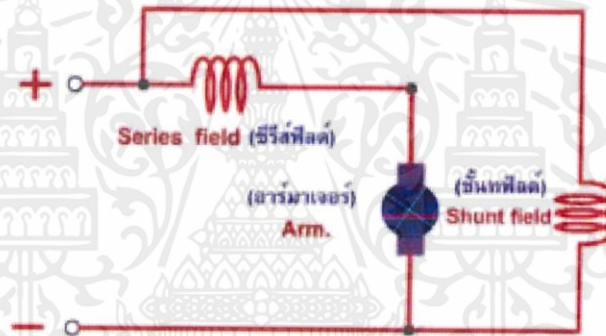
วิธีที่ 1 ใช้ต่อขดลวดแบบชันทขนานกับอาเมเจอร์เรียกว่า ชอทชันท (Short Shunt Compound Motor) ดังรูปวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบชอร์ตชั๊นท์คอมปาว์ดมอเตอร์

วิธีที่ 2 คือต่อขดลวด ขนานกับขดลวดอนุกรมและขดลวดอาเมเจอร์เรียกว่าลวงชั๊นท์คอมปาว์ดมอเตอร์ (Long shunt motor) ดังรูปวงจร



รูปที่ 2.10 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบลวงชั๊นท์คอมปาว์ดมอเตอร์

เนื่องจากมอเตอร์ไฟฟ้ากระแสตรงมีราคาถูกและใช้งานง่าย จึงพบการนำมอเตอร์ไฟฟ้ากระแสตรง มาใช้งานได้หลากหลาย เช่น ของเล่นขนาดเล็ก จักรยานไฟฟ้า แขนกลหุ่นยนต์และเครื่องจักรต่างๆ ในโรงงานอุตสาหกรรมรวมถึงในการทำโครงการครั้งนี้ได้เลือกใช้ มอเตอร์เกียร์ 12 V 50 rpm สำหรับล้อ 2 ตัว และ มอเตอร์ 12 V 1500 rpm สำหรับแปรงปัดหลัก 1 ตัว

การใช้งาน มอเตอร์กระแสตรง ต้องใช้กระแสสูงในการทำงาน ดังนั้น ไมโครคอนโทรลเลอร์ จะไม่สามารถเชื่อมต่อโดยตรง กับ มอเตอร์กระแสตรงได้ จึงต้องมีชุดขับกระแส

#### 2.2.1.4 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง

โดยทั่วไปจะควบคุมองค์ประกอบ 3 ประการคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

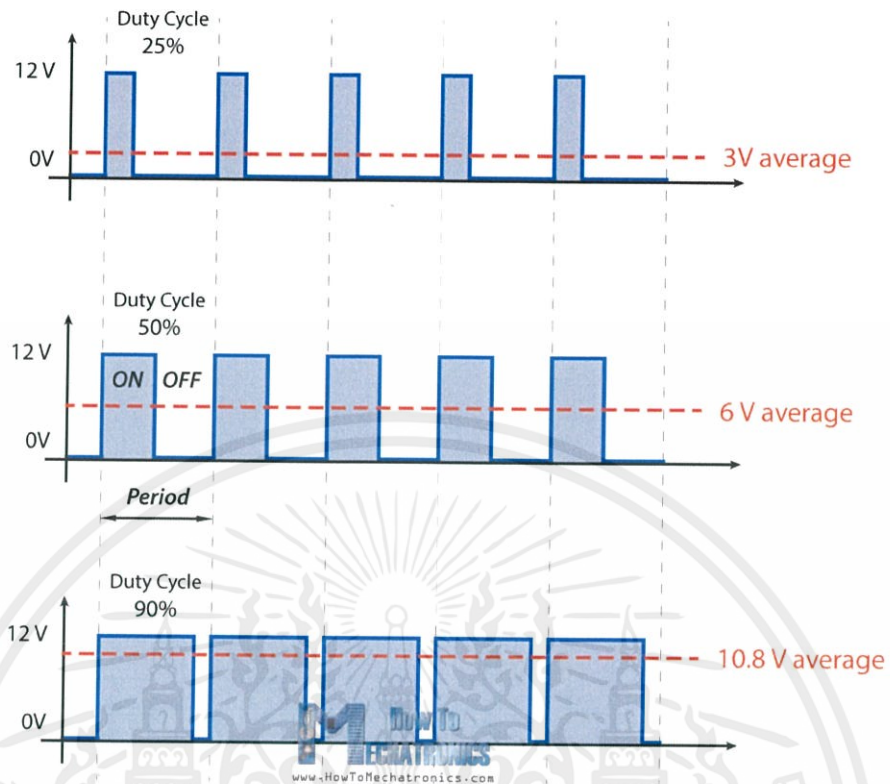
1. ควบคุมความเร็วของมอเตอร์ (Speed Control)
2. ควบคุมแรงบิดของมอเตอร์ (Torque Control)
3. ควบคุมทิศทางการหมุนของมอเตอร์ (Direction Control)

การควบคุมความเร็วของมอเตอร์กระแสตรงนั้น จะทำได้โดยการปรับแรงดันที่จ่ายให้มอเตอร์ ส่วนการควบคุมแรงบิด ทำโดยการควบคุมกระแสที่ผ่านขดลวดอาเมเจอร์ และที่จ่ายให้กับขดลวดสเตเตอร์ ในกรณีสเตเตอร์แบบใช้ขดลวดพัน สำหรับการควบคุมการหมุน หรือการสลับทิศทางการหมุนนั้น ในกรณีมอเตอร์ไฟตรง สามารถทำได้โดยการสลับขั้วแหล่งจ่ายไฟ ที่จ่ายให้แก่มอเตอร์ สำหรับกรณีของ มอเตอร์ไฟฟ้ากระแสตรงขนาดเล็ก แบบใช้แม่เหล็กถาวรเป็นสเตเตอร์ จะไม่สามารถควบคุมอะไรได้มากนัก โดยการควบคุมความเร็วของมอเตอร์ จะถูกจำกัดด้วยขนาดแรงดัน โดยสามารถปรับได้ไม่เกินค่าแรงดันสูงสุดที่จ่ายให้มอเตอร์ ส่วนการควบคุมแรงบิด อาจทำโดยใช้ตัวต้านทานปรับค่าแบบขดลวด (Wire Wound Resistor) แต่ก็เกิดกำลังไฟฟ้าสูญเสียที่ตัวต้านทาน ฉะนั้นในปัจจุบันจึงมักนิยมใช้การควบคุม ด้วยวงจรมอดูเลเตอร์ (Pulse Width Modulator) ซึ่งจะใช้วิธีจ่ายไฟให้แก่มอเตอร์เป็นช่วง ๆ โดยการควบคุมแรงดัน คือการปรับช่วงกว้างของพัลส์ที่จ่ายให้นั่นเอง ซึ่งวิธีนี้จะทำให้ลดกำลังสูญเสียได้มาก สำหรับการกลับทางหมุนของมอเตอร์ อาจใช้วิธีสลับขั้วด้วยมือ หรือใช้วงจรรีเลย์หรืออิเล็กทรอนิกส์เข้าไปควบคุม

#### 2.2.1.5 PWM DC Motor Control

PWM หรือ pulse width modulation เป็นเทคนิคที่ใช้ควบคุม ความเร็วของมอเตอร์ โดยใช้หลักการ เปิด ปิด มอเตอร์ด้วยความเร็วสูง จนผลค่าเฉลี่ยของแรงดันที่ได้ออกมาเทียบเท่ากับการเปลี่ยนแรงดันโดยตรงPWM นั้นโดยทั่วไปจะมีการสร้างลูกคลื่นสี่เหลี่ยม (Square Wave) ออกมาโดยกำหนดคาบของสัญญาณ (Period) หลักการสำคัญของ PWM คือการปรับเปลี่ยนความกว้างของลูกคลื่นในแต่ละคาบ โดยถ้าลูกคลื่นสั้นก็จะทำให้แรงดันเฉลี่ยที่ออกมามีค่าน้อย และถ้าลูกคลื่นยาวแรงดันเฉลี่ยก็จะมีค่ามากขึ้น จากรูปด้านล่าง V เฉลี่ย (เส้นสีส้ม) จะสูงหรือต่ำนั้นขึ้นอยู่กับความกว้างของลูกคลื่น ซึ่งความกว้างของลูกคลื่นนี้เรียกว่า pulse width หรือ Duty Cycle

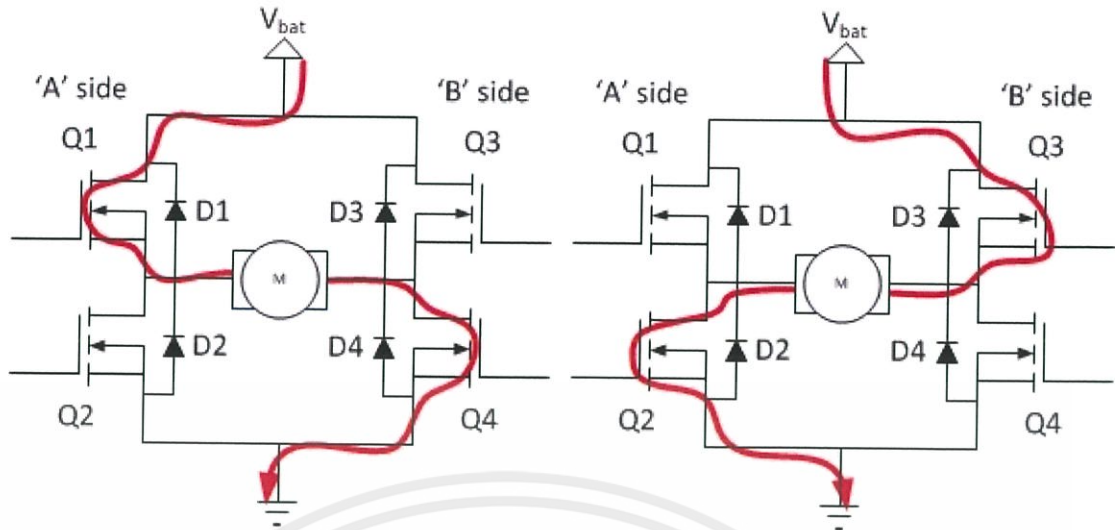
## Pulse Width Modulation



รูปที่ 2.11 Pulse Width Modulation

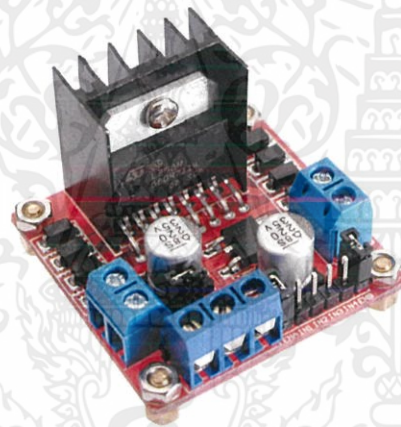
### 2.2.1.5 การขับกระแสให้มอเตอร์กระแสตรง

ชุดขับกระแสดีซีมอเตอร์ส่วนใหญ่จะเป็นแบบ H-Bridge ซึ่งวงจรประกอบด้วยทรานซิสเตอร์หรือมอดเฟส โดยทำหน้าที่เป็นสวิตช์เปิดปิด จำนวน 4 ชุด (Q1-Q4) โดยต่อกับ DC Motor ดังรูปที่ 5 ซึ่งสามารถควบคุมการไหลของกระแสได้ เมื่อส่งสัญญาณควบคุมให้มอสเฟต Q1 และ Q4 ทำงาน และปิดการทำงานของมอสเฟต Q2 และ Q3 กระแสจะไหลจากจุด Vbat ไปจุด GND จึงทำให้มอเตอร์เริ่มหมุน เมื่อส่งสัญญาณควบคุมให้มอสเฟต Q2 และ Q3 ทำงาน และปิดการทำงานของมอสเฟต Q1 และ Q4 กระแสที่ไหลเป็นผลให้มอเตอร์หมุนกลับทิศ



รูปที่ 2.12 วงจร H-Bridge

ในโครงการนี้ชุดขับเคลื่อนที่ใช้ คือ ชุดขับเคลื่อน L298N



รูปที่ 2.13 ชุดขับเคลื่อน L298N

## 2.2.2 เซนเซอร์วัดระยะทาง

### 2.2.2.1 ความหมายของเซนเซอร์วัดระยะทาง

HC-SR04 เป็นเซนเซอร์โมดูลสำหรับตรวจจับวัตถุและวัดระยะทางแบบไม่สัมผัส [1-2] โดยใช้คลื่นอัลตราโซนิก ซึ่งเป็นคลื่นเสียงความถี่สูงเกินกว่าการได้ยินของมนุษย์ มีความถี่ที่ประมาณ 40KHz ข้อดีของการใช้ความถี่นี้ คือมีลักษณะของความยาวคลื่นที่สั้น ส่งผลให้คลื่นไม่แตกกระจายออกเป็นวงกว้าง และสามารถยิงคลื่นตรงไปชนวัตถุใด ๆ ก็ได้ และนอกจากนี้ความถี่ 40kHz ยังเป็นความถี่ที่มีระยะเดินทางเพียงพอกับการใช้งาน หากใช้ความถี่สูงขึ้น จะทำให้คลื่นเดินทางได้ในระยะทางที่ลดลง ทำให้เมื่อนำมาใช้งานจริงจะวัดระยะได้ในระยะที่สั้น วัดระยะได้ตั้งแต่ 2 – 400 เซนติเมตร หรือ 1 –

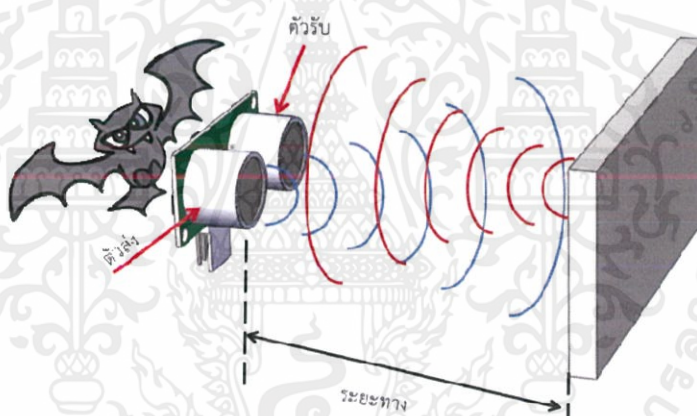
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

156 นิ้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำ เหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติ หรืองานด้านหุ่นยนต์

### 2.2.2.2 หลักการทำงานของเซนเซอร์วัดระยะทาง

หลักการทำงาน จะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของค้างคาว ตามรูปที่ 2.14 โดยจะประกอบไปด้วยตัว รับ-ส่ง อัลตราโซนิก ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ เมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาที่ใช้ในการเดินทางไป-กลับ (t) ก็จะสามารถคำนวณหาระยะห่างของวัตถุ (S) ได้จาก  $S = 346 \times 0.5t$

รุ่น HC-SR04 เป็นรุ่นที่ถูกนิยมใช้งานมากที่สุด การสื่อสารกับไมโครคอนโทรลเลอร์จะใช้วิธีทริกสัญญาณ ขาใช้งานจะมี 4 ขา คือขา VCC Trig Echo และ GND ในรุ่นนี้รองรับแรงดันไฟเลี้ยงที่ 5V กรณีที่แรงดันไฟเลี้ยงน้อยกว่า 5V โมดูลจะไม่สามารถทำงานได้ตามปกติ



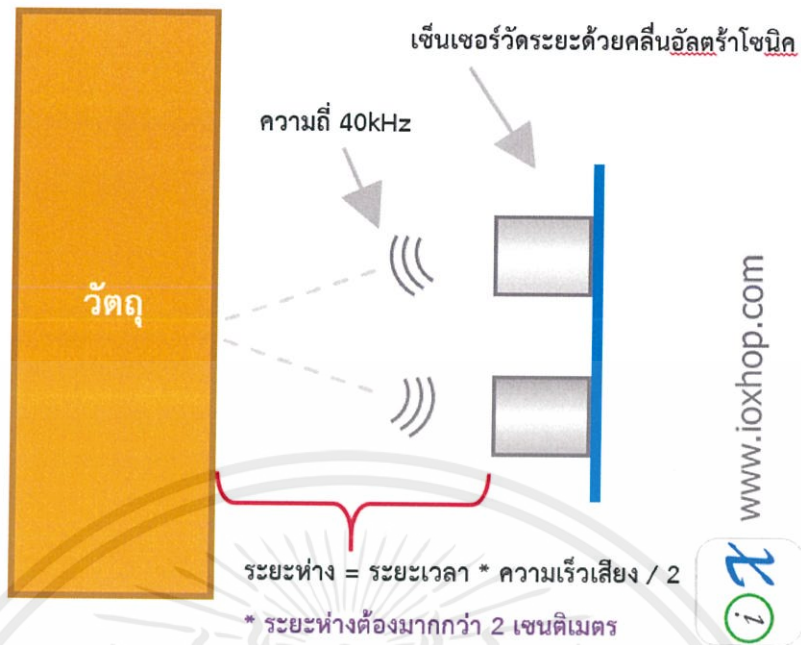
รูปที่ 2.14 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง

เพื่อให้การคำนวณหาระยะเป็นไปด้วยความง่าย โมดูลเซนเซอร์นี้จึงได้ประมวลผลให้เรียบร้อยแล้ว และส่งผลลัพธ์ของการคำนวณเป็นสัญญาณพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้

### 2.2.2.3 หลักการวัดระยะด้วยคลื่นอัลตราโซนิก

หลักการที่สำคัญของการวัดระยะด้วยคลื่นอัลตราโซนิก คือการส่งคลื่นอัลตราโซนิกจำนวนหนึ่งออกไปจากตัวส่ง (Transmitter) เมื่อคลื่นวิ่งไปชนกับวัตถุ คลื่นจะมีการสะท้อนกลับมา แล้ววิ่งกลับไปชนตัวรับ (Receiver) ด้วยการเริ่มนับเวลาที่ส่งคลื่นออกไป จนถึงได้รับคลื่นกลับมานี้เอง ทำให้เราสามารถหาระยะห่างระหว่างวัตถุกับเซ็นเซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

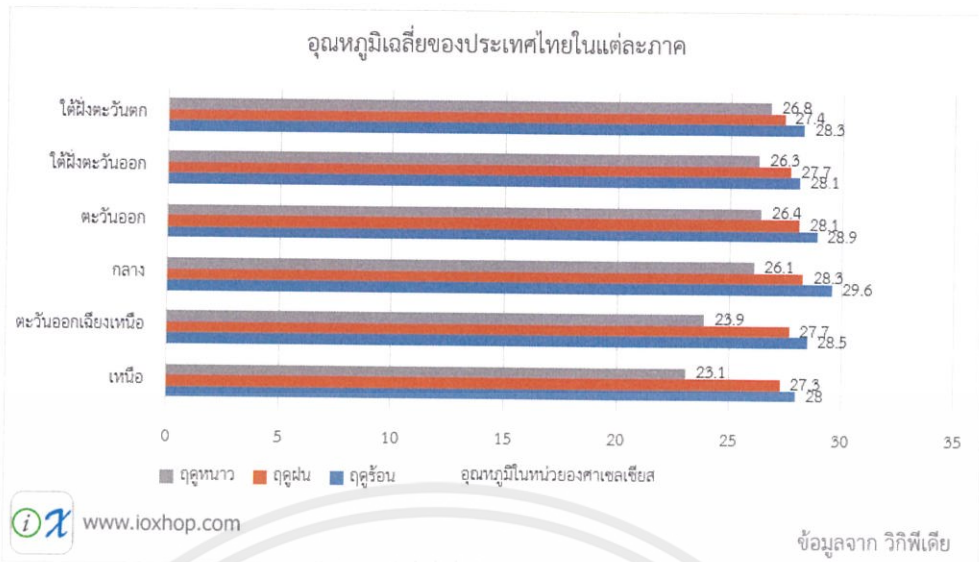


รูปที่ 2.15 การทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

ระยะเวลาที่ได้จากการวัดช่วงเวลาการเดินทางไปและกลับนี้ เราสามารถนำค่าที่ได้ไปเปรียบเทียบกับอัตราเร็วที่เสียงสามารถเดินทางได้ไปในอากาศได้เลย โดยอัตราเร็วเสียงที่เดินทางได้ในอากาศสามารถหาได้ตามสูตร

$$\text{อัตราเร็วของเสียงในอากาศ} = 331 + (0.606 * \text{อุณหภูมิในหน่วยองศาเซลเซียส}) \text{ m/s}$$

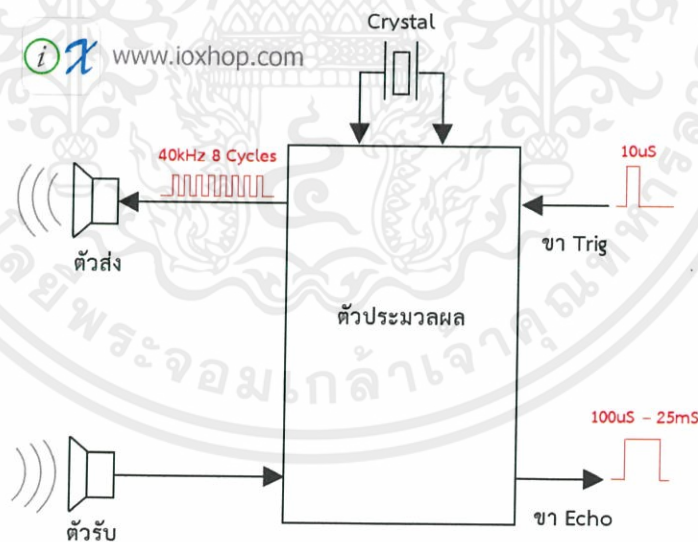
สังเกตว่า อัตราเร็วของเสียงที่เดินทางในอากาศนั้น จะขึ้นอยู่กับอุณหภูมิ ณ ขณะนั้นด้วย ดังนั้นในเซ็นเซอร์อัลตราโซนิกบางรุ่น จึงมีเซ็นเซอร์วัดอุณหภูมิมาด้วย ทำให้สามารถวัดระยะทางได้แม่นยำมากยิ่งขึ้น สำหรับในรุ่นที่ไม่มีเซ็นเซอร์วัดอุณหภูมิ ท่านสามารถนำเซ็นเซอร์วัดอุณหภูมิมาต่อเพื่อแก้ค่าความผิดพลาดเองได้ หรือใช้ค่าอุณหภูมิเฉลี่ยทั้งปีของประเทศไทยได้ โดยอุณหภูมิเฉลี่ยของประเทศไทยทั้งปีจะอยู่ที่ 27 องศาเซลเซียส



รูปที่ 2.16 กราฟแสดงค่าอุณหภูมิเฉลี่ยของประเทศไทยในแต่ละภาค

#### 2.2.2.4 หลักการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

ในโมดูลเซ็นเซอร์อัลตราโซนิกนั้น จะมีวงจรที่แตกต่างกัน เนื่องจากแต่ละรุ่นมีความสามารถที่แตกต่างกัน แต่ยังคงมีหลักการทำงานที่สำคัญที่เหมือนกัน



รูปที่ 2.17 ไดอะแกรมการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

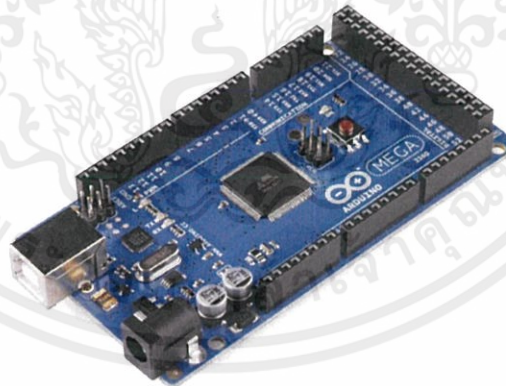
จากรูปที่ 2.17 จะเห็นว่า เมื่อมีการส่งสัญญาณเข้าไปที่ Trig วงจรภายในจะเริ่มสร้างความถี่

40kHz จำนวน 8 ลูกคลื่นออกไป โดยใช้ความถี่จากคริสตอลเป็นตัวอ้างอิง แล้วตัวส่งที่เปรียบเสมือนลำโพง จะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นวิ่งกลับมาที่ตัวรับ ที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้าจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo จะเห็นว่า แกนหลักของเซ็นเซอร์จะเป็นตัวประมวลผล ซึ่งตัวประมวลผลนี้ ในแต่ละรุ่นก็จะแตกต่างกัน อย่างในรุ่น HC-SR04 จะใช้ไอซีไมโครคอนโทรลเลอร์เบอร์ ATtiny24 ในรุ่น US-100 ก็จะใช้ไมโครคอนโทรลเลอร์แบบเดียวกัน แต่ไม่ทราบเบอร์ เนื่องจากเบอร์ไอซีบนโมดูลโดนลบ

### 2.2.3 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้าหรือระบบอิเล็กทรอนิกส์ต่างๆ ไมโครคอนโทรลเลอร์นั้นเปรียบเสมือนคอมพิวเตอร์ขนาดเล็กสามารถเขียนโปรแกรมลงไปได้ ไมโครคอนโทรลเลอร์ก็เลยเข้ามาเกี่ยวข้องกับเรื่องรับกับความอยากนำไปควบคุมระบบที่ความรู้ความเข้าใจที่ต้องการโดยให้มีขนาดเล็กที่สุด แต่ว่าไม่ใช่เพียงขนาดเล็กเท่านั้น มันยังสามารถป้อนชุดคำสั่งให้สามารถปฏิบัติงานได้อย่างอัตโนมัติ ด้วยแบบการเขียนโปรแกรมภาษาต่างๆ ตามความชำนาญ นัก ออกแบบ พัฒนาผลิตภัณฑ์ ตลอดจนนักประดิษฐ์ทั้งหลาย ต่างหลีกเลี่ยงไม่ได้เลยที่จะต้องอาศัยวงจรอิเล็กทรอนิกส์ เข้าไปมีส่วนเกี่ยวข้องในการควบคุม

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดูยโน) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย



รูปที่ 2.18 Arduino MEGA 2560 R3

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino

Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้

Arduino Mega 2560 คือบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาจาก ATmega2560 มี 54 digital input/output โดยมี 14 ขา สามารถใช้เป็น output แบบ PWM ได้ มี analog inputs 16 ขา มี UARTs(hardware serial ports) 4 ขา ทำงานที่ความถี่ 16 MHz สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้ adaptor AC-to-DC เพื่อเริ่มต้นใช้งาน และมีปุ่ม reset สามารถต่อเข้ากับ shields ที่ออกแบบเพื่อใช้งานกับ Arduino Duemilanove หรือ Diecimila.

## 2.2.4 บลูทูธ

บลูทูธคือ ระบบสื่อสารของอุปกรณ์อิเล็กทรอนิกส์แบบสองทาง ด้วยคลื่นวิทยุระยะสั้น (Short-Range Radio Links) โดยปราศจากการใช้สายเคเบิล หรือ สายสัญญาณเชื่อมต่อ และไม่จำเป็นต้องใช้การเดินสายแบบเส้นตรงเหมือนกับอินฟราเรด ซึ่งถือว่าเพิ่มความสะดวกมากกว่าการเชื่อมต่อแบบอินฟราเรด ที่ใช้ในการเชื่อมต่อระหว่างโทรศัพท์มือถือ กับอุปกรณ์ ในโทรศัพท์เคลื่อนที่รุ่นก่อนๆ และในการวิจัย ไม่ได้มุ่งเฉพาะการส่งข้อมูลเพียงอย่างเดียว แต่ยังคงศึกษาถึงการส่งข้อมูลที่เป็นเสียง เพื่อใช้สำหรับ Headset บนโทรศัพท์มือถือเช่นกัน

### 2.2.4.1 การทำงานของบลูทูธ

บลูทูธ จะใช้สัญญาณวิทยุความถี่สูง 2.4 GHz. (กิกะเฮิรซ์) แต่จะแยกย่อยออกไป ตามแต่ละประเทศ 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 ป้องกันการดักสัญญาณระหว่างสื่อสาร โดยระบบจะสลับช่องสัญญาณไปมา จะมีความสามารถในการเลือกเปลี่ยนความถี่ที่ใช้ในการติดต่อเองอัตโนมัติ โดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ทำให้การดักฟังหรือลักลอบขโมยข้อมูลทำได้ยากขึ้น

โดยหลักของบลูทูธจะถูกออกแบบมาเพื่อใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากใช้การขนส่งข้อมูลในจำนวนที่ไม่มาก อย่างเช่น ไฟล์ภาพ, เสียง, แอปพลิเคชันต่างๆ และสามารถเคลื่อนย้ายได้ง่าย ขอให้อยู่ในระยะที่กำหนดไว้เท่านั้น (ประมาณ 5-10 เมตร) นอกจากนี้ยังใช้พลังงานต่ำ กินไฟน้อย และสามารถใช้งานได้นาน โดยไม่ต้องนำไปชาร์จไฟบ่อยๆ

ส่วนความสามารถการส่งถ่ายข้อมูลของ บลูทูธ จะอยู่ที่ 1 Mbps (1 เมกกะบิตต่อวินาที) และ คงจะไม่มีปัญหาอะไรมากกับขนาดของไฟล์ที่ใช้กันบนโทรศัพท์มือถือ หรือ การใช้งานแบบทั่วไป ซึ่ง ถือว่าเหลือเฟือมาก แต่ถ้าเป็นข้อมูลที่มีขนาดใหญ่ล่ะก็ คงจะช้าเกินไป และถ้าถูกนำไปเปรียบกับ Wireless LAN (WLAN) แล้ว ความสามารถของ บลูทูธ คงจะห่างชั้นกันเยอะ ซึ่งในส่วนของ WLAN ก็ยังมีระยะการรับ-ส่งที่ไกลกว่า แต่ขอได้เปรียบของ Bluetooth จะอยู่ที่ขนาดที่เล็กกว่า การติดตั้งทำได้ง่ายกว่า และที่สำคัญ การใช้พลังงานก็น้อยกว่ามาก อยู่ที่ 0.1 วัตต์ หากเทียบกับคลื่นมือถือแล้ว ยัง ห่างกันอยู่หลายเท่า

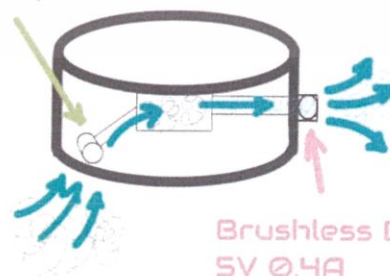
โดยในโครงการนี้เลือกใช้ บลูทูธโมดูล HC-05 มาเชื่อมต่อกับโทรศัพท์มือถือ HC05 เป็นโมดูล บลูทูธที่ใช้งานในการเชื่อมต่อกับสมาร์ตทีวีต่างๆ ให้สมาร์ตทีวีสามารถสื่อสารกับ ไมโครคอนโทรลเลอร์ (Arduino AVR PIC etc.) ได้ ผ่าน Serial port โมดูลรุ่น HC05 สามารถตั้งให้ ใช้งานเป็นได้ทั้งโหมด Master (ให้อุปกรณ์อื่นมาเชื่อมต่อ) และโหมด Slave (เชื่อมต่อกับอุปกรณ์อื่น) การตั้งค่าต่างๆ เช่น ชื่ออุปกรณ์ รหัสผ่าน ทำได้ผ่าน AT Command ซึ่งจะต้องมีการต่อขาพิเศษ เพื่อให้โมดูลเข้าโหมดการตั้งค่า หรือกดปุ่มบนโมดูลค้างไว้



รูปที่ 2.19 บลูทูธโมดูล HC-05

## 2.2.5 หลักการดูดฝุ่นของเครื่องดูดฝุ่น

DC Motor  
18000rpm 12V



รูปที่ 2.20 การดูดของเครื่องดูดฝุ่น

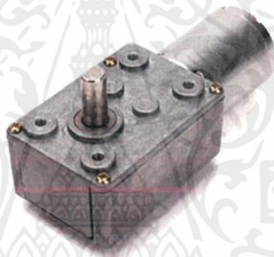
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของเครื่องดูดฝุ่นคือการใช้มอเตอร์เพื่อขับเคลื่อนใบพัดให้หมุนด้วยความเร็วสูง เพื่อให้อากาศในส่วนของเครื่องดูดถูกเติมเข้าไปในพัดลมอย่างต่อเนื่อง จะทำให้สูญญากาศในห้อง สูญญากาศเพื่อสร้างความแตกต่างความดันลบกับความดันบรรยากาศภายนอก ภายใต้การกระทำของความแตกต่างความดันฝุ่นและสิ่งสกปรกที่หายใจเข้ากับอากาศจะถูกกรองผ่านตัวกรองและฝุ่นละออง และสิ่งสกปรกจะถูกรวบรวมในกล่องเก็บฝุ่นฝุ่น ยิ่งความแตกต่างของแรงดันลบและปริมาณอากาศมากขึ้นเท่าใดความสามารถในการดูดซับฝุ่นยิ่งมากขึ้น

โดยเครื่องดูดฝุ่นนี้มีเส้นทางเดินอากาศตามรูปที่ 2.20 โดยจะมีมอเตอร์กระแสตรง 12V 1800 rpm ต่อกับแปรงปัดเพื่อให้เกิดลมพัดเข้าไปในเครื่องดูดฝุ่นและที่ด้านหลังของตัวเครื่องมีพัดลมดูดอากาศ 5V 0.4A ทำหน้าที่สร้างแรงดูดเสริมกำลังดูดให้กับเครื่องดูดฝุ่น

## 2.3 ทฤษฎีอุปกรณ์

### 2.3.1 มอเตอร์เกียร์กระแสตรง



รูปที่ 2.21 มอเตอร์เกียร์กระแสตรง

มอเตอร์เกียร์ เป็นการรวมกันของมอเตอร์และหัวเกียร์ ซึ่งการเพิ่มหัวเกียร์ให้กับมอเตอร์นั้นจะเป็นการช่วย ลดความเร็วและเพิ่มแรงบิดให้กับมอเตอร์ โดยความเร็วของมอเตอร์เกียร์นั้นจะมีหน่วยเป็น รอบต่อนาที หรือ rpm (Revolutions per minute) ซึ่งมอเตอร์ชนิดนี้จะมีทั้งแบบกระแสสลับ และกระแสตรง ซึ่งมอเตอร์ชนิดนี้จะหมุนเมื่อมีการจ่ายไฟ และจะล๊อคแกนเมื่อไม่มีการจ่ายไฟ กล่าวคือมอเตอร์จะไม่ทำงานและไม่สามารถหมุนได้เมื่อไม่จ่ายไฟให้กับมอเตอร์

เนื่องจากมอเตอร์ชนิดนี้มีราคาถูกและหาซื้อได้ง่ายอีกทั้งให้แรงบิดที่สูง จึงสามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น ของเล่นขนาดเล็ก แขนกลหุ่นยนต์ เป็นต้น ซึ่งมอเตอร์ตามรูปที่ 2.21 เป็นมอเตอร์ที่ใช้ในการขับเคลื่อนล้อของเครื่องดูดฝุ่น ซึ่งเป็นมอเตอร์กระแสตรง 12 V 50 rpm

### 2.3.2 มอเตอร์กระแสตรง



รูปที่ 2.22 มอเตอร์กระแสตรง

มอเตอร์ตามรูปที่ 2.22 เป็นมอเตอร์กระแสตรง 12 V 1800 rpm เนื่องจากเป็นมอเตอร์สำหรับแปรงปัดฝุ่นจึงต้องการมอเตอร์ที่มีความเร็วสูง

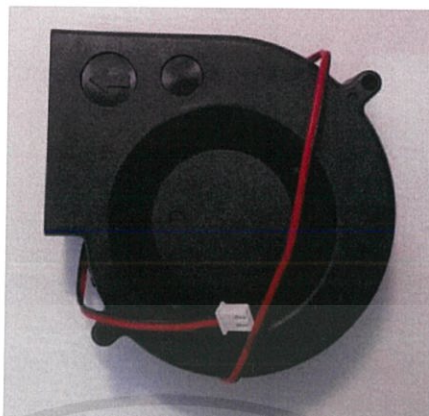
### 2.3.3 ล้อ



รูปที่ 2.23 ล้อ

ล้อ คือวัตถุรูปวงกลมถูกยึดไว้ด้วยเพลลาที่บริเวณจุดศูนย์กลาง ทำให้ล้อสามารถหมุนรอบเพลลาได้ มักใช้ประกอบกับยานพาหนะเพื่อให้เคลื่อนที่ได้โดยง่ายบนพื้น ล้อในปัจจุบันมีอยู่หลายแบบหลายขนาดโดยทำจากวัสดุหลายอย่างตามการใช้งาน เช่น ล้อเกวียนทำด้วยไม้ ล้อแม็ก ล้อรถยนต์ ที่มียางรถยนต์หุ้ม ล้อรถไฟที่ทำด้วยเหล็ก ล้อใส่ตู้ที่ทำจากพลาสติก เป็นต้นแต่ในโครงงานนี้ใช้ ล้ออิสระขนาด จำนวน 1 ล้อ ล้อหุ่นยนต์ขนาด จำนวน 2 ล้อ

### 2.3.4 พัดลมดูดอากาศ



รูปที่ 2.24 พัดลมดูดอากาศ

พัดลมดูดอากาศขนาด 5v 0.4A ติดอยู่หลังเครื่องดูดฝุ่นทำหน้าที่สร้างแรงดูดให้กับเครื่องดูด

ฝุ่น

## บทที่ 3

# วิธีดำเนินการวิจัย

### 3.1 ศึกษาหาข้อมูล

ศึกษาหาข้อมูลเกี่ยวกับเครื่องดูดฝุ่น เพื่อที่จะออกแบบอย่างไรและเลือกใช้อุปกรณ์อะไรบ้างเพื่อประสิทธิภาพการทำงานที่ดีของเครื่องดูดฝุ่นและความสะดวกของผู้ใช้งาน จากการศึกษาได้ข้อสรุปว่าจะทำการแบ่งระบบของเครื่องดูดฝุ่นออกเป็น 2 ระบบ โดยแบ่งเป็น ระบบอัตโนมัติและระบบแมนนวล โดยจะใช้ Arduino ในการควบคุมการทำงานทั้งหมดซึ่งส่งงานผ่าน ไมโครคอนโทรลเลอร์ซึ่งระบบอัตโนมัติจะใช้ เซนเซอร์อัลตราโซนิก ในการวัดระยะทางเพื่อป้องกันการชนของเครื่องดูดฝุ่นกับสิ่งกีดขวางแล้วก่อให้เกิดความเสียหาย ส่วนระบบแมนนวลจะใช้รีโมทในการควบคุมเครื่องดูดฝุ่นโดยส่งงานผ่านบลูทูธโดยใช้ บลูทูธโมดูล HC-05 ในส่วนของการขับเคลื่อนของล้อจะใช้มอเตอร์กระแสตรง โดยมีวงจรขับเคลื่อนต่อกับมอเตอร์คอยรับคำสั่งจาก ไมโครคอนโทรลเลอร์ คอยควบคุมทิศทางและกำหนดความเร็วให้กับมอเตอร์ ในส่วนของแปรงปัดจะใช้มอเตอร์กระแสตรงเช่นกัน โดยมีมอเตอร์สำหรับแปรงปัดข้าง และมอเตอร์สำหรับแปรงปัดหลัก ทำงานแยกกับมอเตอร์ของล้อ โดยแบตเตอรี่ที่ใช้จ่ายไฟให้กับเครื่องดูดฝุ่นต้องมีกระแสจ่ายให้เพียงพอสำหรับมอเตอร์ทั้ง 5 ตัว

### 3.2 ขั้นตอนดำเนินงาน

#### 3.2.1 วางแผนการทำเครื่องดูดฝุ่น

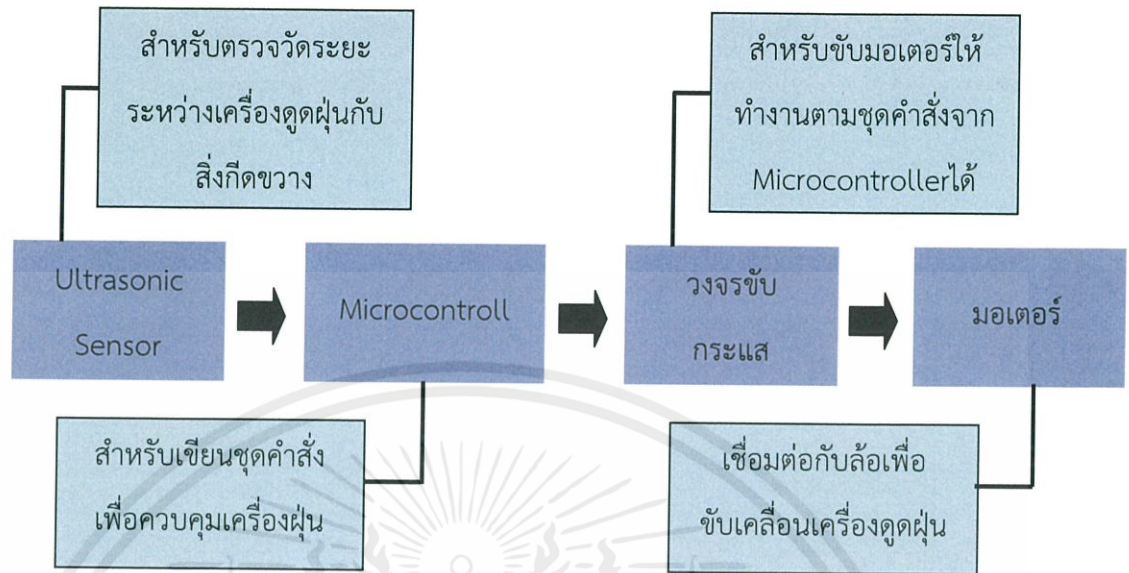


รูปที่ 3.1 บล็อกไดอะแกรมแสดงขั้นตอนการดำเนินงาน

ทำการวางแผนการทำโครงการโดยสร้างบล็อกไดอะแกรมแสดงขั้นตอนการทำงาน

### 3.2.2 ออกแบบการทำงานของเครื่องดูดฝุ่น

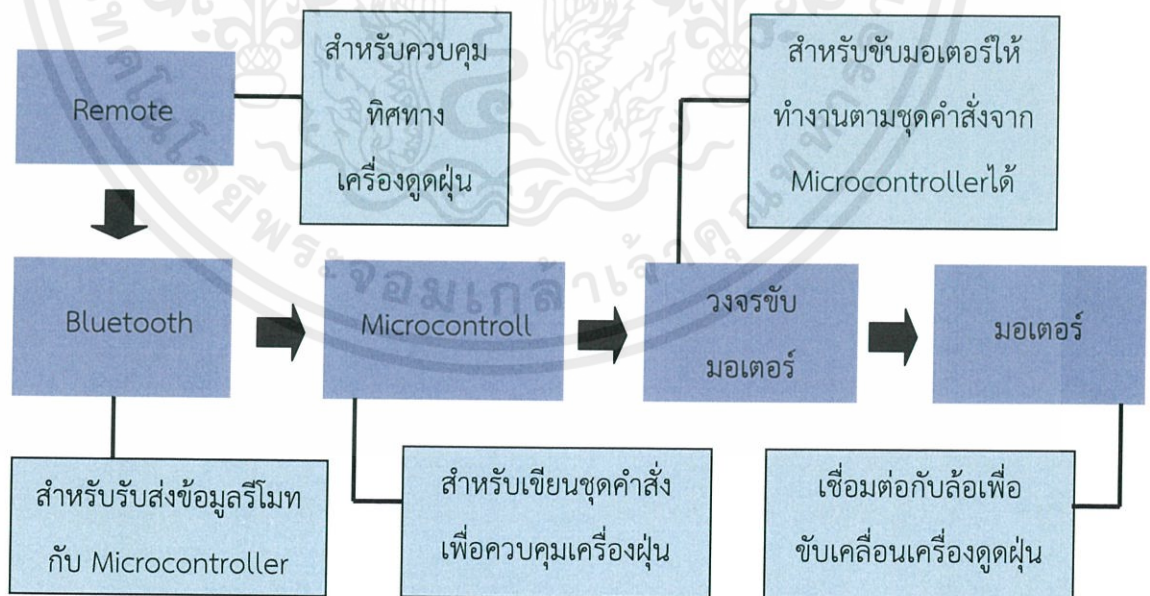
-ระบบอัตโนมัติ



รูปที่ 3.2 บล็อกไดอะแกรมแสดงขั้นตอนการทำงานของเครื่องดูดฝุ่นระบบอัตโนมัติ

จากบล็อกไดอะแกรมแสดงให้เห็นว่า Microcontroller จะรับข้อมูลจากเซนเซอร์อัลตราโซนิกเพื่อควบคุมทิศทางของมอเตอร์ผ่านวงจรขับกระแส และการควบคุมความเร็วของมอเตอร์จะถูกกำหนดด้วยค่า PWM โดยผ่านวงจรขับกระแสเช่นกัน

-ระบบแมนนวล



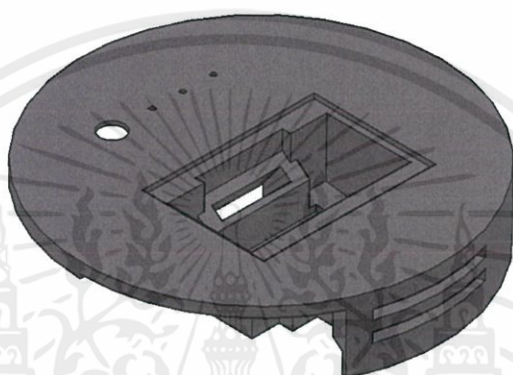
รูปที่ 3.3 บล็อกไดอะแกรมแสดงขั้นตอนการทำงานของเครื่องดูดฝุ่นระบบแมนนวล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อกไดอะแกรมแสดงให้เห็นว่ารีโมทจะส่งข้อมูลผ่านระบบลูทูปไปยัง Microcontroller โดยระบบนี้สามารถใช้รีโมทควบคุมทิศทางของมอเตอร์ได้ โดยจะสามารถควบคุมทิศทางของมอเตอร์ผ่านวงจรขับกระแสที่ได้รับคำสั่งจากชุดคำสั่งของ Microcontroller และการควบคุมความเร็วของมอเตอร์จะถูกกำหนดด้วยค่า PWM โดยผ่านวงจรขับกระแสเช่นกัน

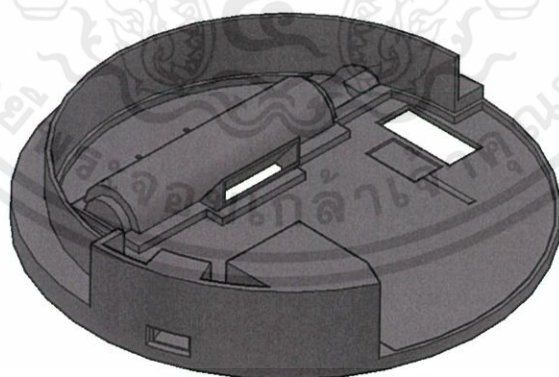
### 3.2.3 ออกแบบโครงของเครื่องดูดฝุ่น

-โครงเครื่องดูดฝุ่นส่วนบน



รูปที่ 3.4 โครงเครื่องดูดฝุ่นส่วนบน

-โครงเครื่องดูดฝุ่นส่วนล่าง



รูปที่ 3.5 โครงเครื่องดูดฝุ่นส่วนล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4 ประกอบเครื่องดูดฝุ่น

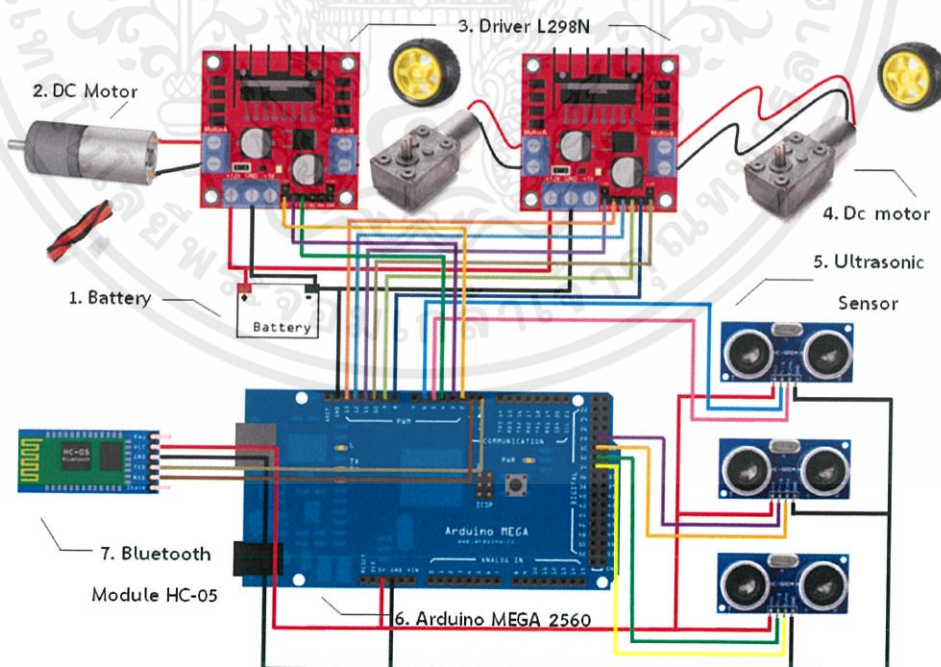
#### 3.2.4.1 ประกอบโครงเครื่องดูดฝุ่น

โครงเครื่องดูดฝุ่นจะถูกแบ่งเป็น 2 ส่วนแล้วนำมาประกอบเข้าด้วยกัน โดยโครงทำขึ้นจากการพิมพ์ 3 มิติ และ ตัดอะคริลิก



รูปที่ 3.6 เครื่องดูดฝุ่น

#### 3.2.4.2 ประกอบวงจรเครื่องดูดฝุ่น



รูปที่ 3.7 แสดงภาพรวมวงจรของเครื่องดูดฝุ่น

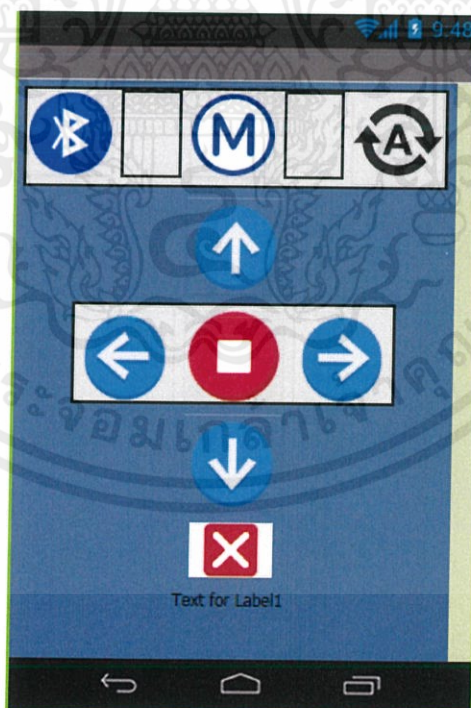
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบวงจรของเครื่องดูดฝุ่น

1. แบตเตอรี่ ใช้ถ่านไฮดรายจำนวน 10 ก้อน 12v ความจุ 1300 mAh เป็นแหล่งจ่ายไฟของเครื่องดูดฝุ่น
2. มอเตอร์กระแสตรง เป็นมอเตอร์ขนาด 12V 1800 rpm ใช้คู่กับแปรงปัดหลัก
3. บอร์ดขับกระแส L298N ใช้ขับมอเตอร์ที่ใช้ขับเคลื่อนเครื่องดูดฝุ่น และขับมอเตอร์ที่ใช้กับแปรงปัดหลัก
4. มอเตอร์กระแสตรง 12V 50 rpm ใช้ขับเคลื่อนล้อทั้ง 2 ข้างของเครื่องดูดฝุ่น
5. เซนเซอร์อัลตราโซนิก HC-SR04 จำนวน 3 ตัว สำหรับวัดระยะทาง ตีกรอบเครื่องดูดฝุ่น
6. Arduino เป็น Microcontroller ให้กับเครื่องดูดฝุ่น
7. Bluetooth Module HC-05 เชื่อมต่อกับรีโมท ให้สามารถสื่อสารกับ Microcontroller ในระบบแมนนวล

### 3.2.5 การสร้างแอปพลิเคชัน

#### 3.2.5.1 ออกแบบแอปพลิเคชัน



รูปที่ 3.8 หน้าต่างการทำงานของแอปพลิเคชัน Bcontrol

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการออกแบบหน้าต่างแอปพลิเคชันของระบบ Android โดยใช้ MIT App Inventor ใน  
สร้าง เน้นการออกแบบให้เข้าใจง่ายต่อการใช้งานของผู้ใช้เครื่องดูฝุ่น



ปุ่มBluetooth เมื่อกดปุ่มนี้จะปรากฏหน้าต่างรายชื่อ Bluetooth ให้เลือกให้การ  
เชื่อมต่อกับโทรศัพท์มือถือ



ปุ่มเปลี่ยนโหมดเป็นระบบแมนนวล เมื่อกดปุ่มนี้แอปพลิเคชันจะทำการส่งตัวอักษร 'M'



ปุ่มเปลี่ยนโหมดเป็นระบบอัตโนมัติ เมื่อกดปุ่มนี้แอปพลิเคชันจะทำการส่งตัวอักษร 'A'  
ผ่านบลูทูธ



ปุ่มเดินหน้า เมื่อกดปุ่มนี้เครื่องดูฝุ่นจะเดินหน้า แอปพลิเคชันจะทำการส่งตัวอักษร 'F'  
ผ่านบลูทูธ



ปุ่มเดินหน้า เมื่อกดปุ่มนี้เครื่องดูฝุ่นจะเดินหน้า แอปพลิเคชันจะทำการส่งตัวอักษร 'F'  
ผ่านบลูทูธ



ปุ่มเลี้ยวซ้าย เมื่อกดปุ่มนี้เครื่องดูฝุ่นจะเลี้ยวซ้าย แอปพลิเคชันจะทำการส่งตัวอักษร 'L'  
ผ่านบลูทูธ



ปุ่มเลี้ยวขวา เมื่อกดปุ่มนี้เครื่องดูฝุ่นจะเลี้ยวขวา แอปพลิเคชันจะทำการส่งตัวอักษร 'R'  
ผ่านบลูทูธ



ปุ่มหยุด เมื่อกดปุ่มนี้เครื่องดูฝุ่นจะหยุดทำงาน แอปพลิเคชันจะทำการส่งตัวอักษร 'S'  
ผ่านบลูทูธ



ปุ่มปิดแอปพลิเคชัน เมื่อกดปุ่มนี้จะเป็นการออกจากแอปพลิเคชันไปหน้าโฮมของ  
โทรศัพท์มือถือ

### 3.2.5.2 เขียนบล็อกให้แอปพลิเคชัน

The image displays three sections of code blocks from an Android Studio interface, numbered 1, 2, and 3.

**Section 1:**

- when Bluetoothicon - .BeforePicking
  - do set Bluetoothicon - . Elements - to BluetoothClient1 - . AddressesAndNames -
- when Bluetoothicon - .AfterPicking
  - do set Bluetoothicon - . Selection - to call BluetoothClient1 - .Connect address Bluetoothicon - . Selection -

**Section 2:**

- when Manual - .Click
  - do call BluetoothClient1 - .SendText text "M"
- when Forward - .LongClick
  - do call BluetoothClient1 - .SendText text "F"
- when Left - .LongClick
  - do call BluetoothClient1 - .SendText text "L"
- when Stop - .Click
  - do call BluetoothClient1 - .SendText text "S"
- when Right - .LongClick
  - do call BluetoothClient1 - .SendText text "R"
- when Backward - .LongClick
  - do call BluetoothClient1 - .SendText text "B"
- when Automatic - .Click
  - do if Automatic - . Enabled - then call BluetoothClient1 - .SendText text "A" set Label1 - . Text - to "A"
- when Forward - .TouchUp
  - do call BluetoothClient1 - .SendText text "S"
- when Right - .TouchUp
  - do call BluetoothClient1 - .SendText text "S"
- when Left - .TouchUp
  - do call BluetoothClient1 - .SendText text "S"
- when Backward - .TouchUp
  - do call BluetoothClient1 - .SendText text "S"

**Section 3:**

- when exit - .Click
  - do close application

รูปที่ 3.9 บล็อกโปรแกรมของแอปพลิเคชัน

1. ส่วนของการเชื่อมต่อบลูทูธกับอุปกรณ์ต่างๆ
2. ส่วนการส่งข้อมูลเมื่อกดปุ่มต่างๆจะส่งตัวอักษรตามที่กำหนดออกไป เช่น เมื่อกดปุ่มเดินหน้า โทรศัพท์จะทำการส่งตัวอักษร 'F' ออกไป
3. ส่วนของการปิดแอปพลิเคชัน

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 การวัดความสามารถในการดูดฝุ่น

##### 4.1.1 วิธีการทดลอง

1. ทดสอบให้เครื่องดูดฝุ่นเดินผ่านจุดที่มีฝุ่น ขนาดเล็กโดยใช้เศษกระดาษขนาดไม่เกิน  $0.5 \times 0.5 \text{ cm}^2$

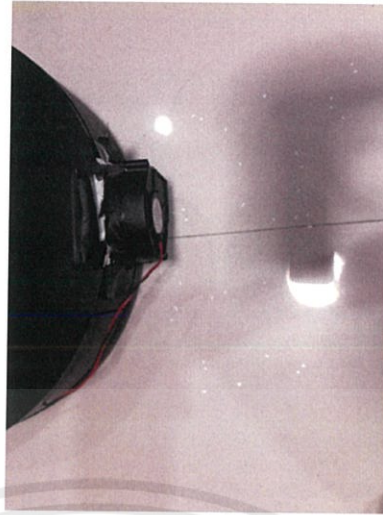


รูปที่ 4.1 เศษกระดาษบนพื้นก่อนการดูด

2. ตรวจสอบที่พื้นว่ากระดาษถูกดูดไปหรือไม่
3. ตรวจสอบดูว่าในกล่องเก็บฝุ่นมีกระดาษถูกดูดเข้าไปหรือไม่

##### 4.1.2 ผลการทดลอง

เมื่อเครื่องดูดฝุ่นเดินผ่านจุดที่มีกระดาษ พบว่าบนพื้นยังมีเศษกระดาษ อยู่อย่างกระจัดกระจายจากลมของเครื่องดูดฝุ่น



รูปที่ 4.2 เศษกระตาศบนพื้นหลังการดูด

เมื่อตรวจดูในกล่องเก็บฝุ่นพบเศษกระตาศอยู่เล็กน้อย



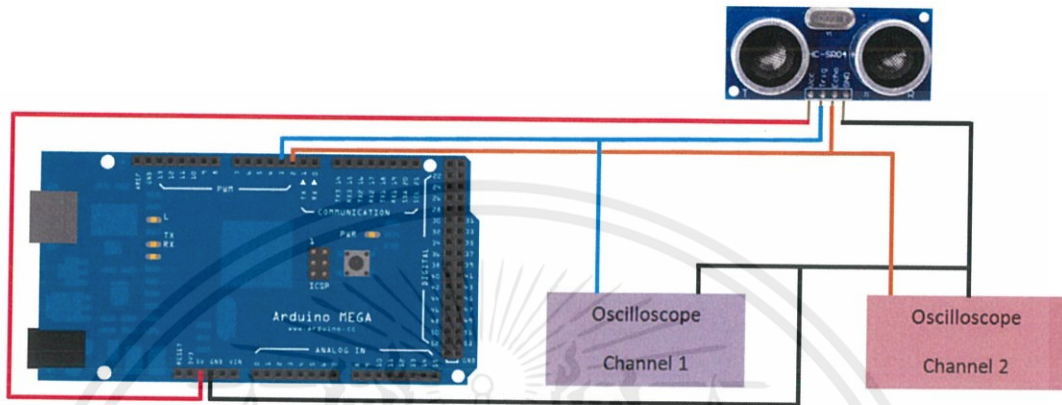
รูปที่ 4.3 เศษกระตาศในกล่องเก็บฝุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ทดสอบการรับ – ส่งสัญญาณของ เซนเซอร์อัลตราโซนิก

### 4.2.1 วิธีการทดลอง

ทำการเปรียบเทียบระยะทางที่ เซนเซอร์อัลตราโซนิกทั้ง 3 ตัววัดได้ที่ระยะทาง 5,10 และ 15 เซนติเมตร



รูปที่ 4.4 วงจรสำหรับการทดสอบการรับส่งสัญญาณเซนเซอร์อัลตราโซนิก

ต้องวงจรตามรูปที่ 4.4 เพื่อทำการทดสอบความแม่นยำของเซนเซอร์

1. ต่อขา Echo ของ เซนเซอร์อัลตราโซนิกเข้ากับ port 2 ของ Arduino
2. ต่อขา Trig ของ เซนเซอร์อัลตราโซนิกเข้ากับ port 3 ของ Arduino
3. ต่อขา GND ของ เซนเซอร์อัลตราโซนิกเข้ากับ GND ของ Arduino
4. ต่อขา vcc ของ เซนเซอร์อัลตราโซนิกเข้ากับ 5V ของ Arduino
5. ทำการวัดค่าโดยใช้ Channel 1 วัดค่า Trig และ ใช้ Channel 2 วัดค่า Echo
6. ทำการวัดพัลส์ของ Echo ที่ระยะ 5,10 และ 15 เซนติเมตร ของ เซนเซอร์อัลตราโซนิกทั้ง 3 ตัว
7. นำค่าที่วัดได้มาคำนวณหาระยะทางโดยใช้สูตร

$$\text{ระยะทาง} = \frac{\text{ระยะเวลา} \times \text{อัตราเร็วเสียงในอากาศ}}{2} \quad (4.1)$$

$$\text{อัตราเร็วเสียงในอากาศ} = 331 + (0.606 * \text{อุณหภูมิในหน่วยองศาเซลเซียส}) \text{ m/s}$$

## 4.2.2 ผลการทดลอง

เซนเซอร์อัลตราโซนิกตัวที่ 1

ที่ระยะ 5 เซนติเมตร

```

resultUltra
void setup() {
  pinMode(trigPin2, OUTPUT);
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(10);
  pinMode(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distance2 = duration2 * 0.034 / 2;
  if (distance2 >= 15 || distance2 <= 2) {
    Y = 0;
  }
  else {
    Y = 1;
  }
  Serial.println("Ultrasonic 2");
  Serial.println(distance2);
  delay(100);

  digitalWrite(trigPin3, LOW);
  digitalWrite(trigPin3, HIGH);
  digitalWrite(trigPin3, LOW);
  delayMicroseconds(10);
  duration3 = pulseIn(echoPin3, HIGH);
  distance3 = duration3 * 0.034 / 2;
  if (distance3 >= 15 || distance3 <= 2) {
    Z = 0;
  }
  else {
    Z = 1;
  }
  Serial.println("Ultrasonic 3");
  Serial.println(distance3);
  delay(100);
}

```

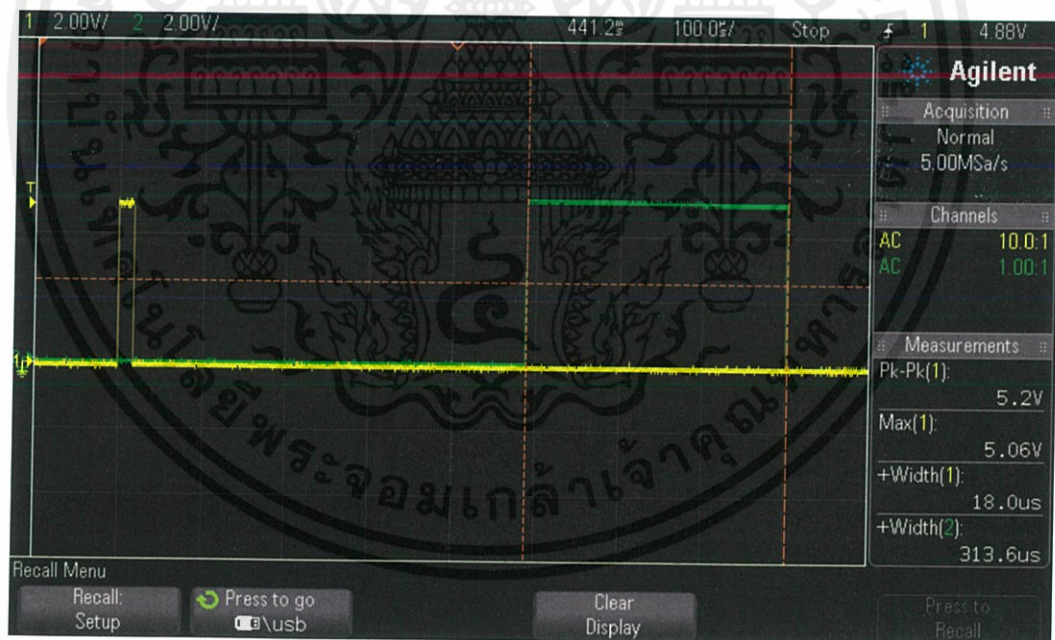
Serial Monitor Output:

```

Ultrasonic 2
503
Ultrasonic 3
2377
Ultrasonic 1
5
Ultrasonic 2
503
Ultrasonic 3
2376
Ultrasonic 1
5
Ultrasonic 2
505
Ultrasonic 3
2376
Ultrasonic 1
5
Ultrasonic 2
502

```

รูปที่ 4.5 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 5 เซนติเมตร



รูปที่ 4.6 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 5

เซนติเมตร

ค่าพัลส์ที่วัดได้ = 313.6 us ค่าที่คำนวณได้ = 5.33 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระยะ 10 เซนติเมตร

```

result@bra | Arduino 1.6.7
File: Edit Sketch Tools Help

result@bra
--- --
int Y = 0;
int Z = 0;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);
}

void loop() {

  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = duration1 * 0.034 / 2;
  if (distance1 >= 15 || distance1 <= 2) {
    X = 0;
  }
  else {
    X = 1;
  }
  Serial.println("Ultrasonic 1");
}

```

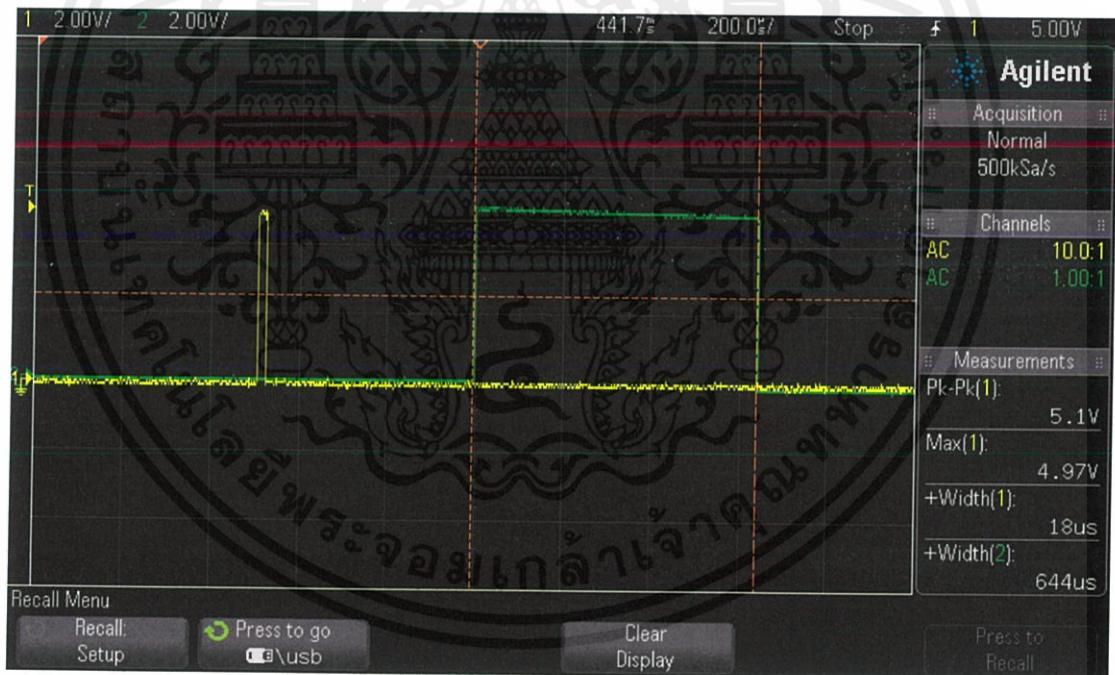
Serial Monitor Output:

```

10
Ultrasonic 2
221
Ultrasonic 3
2372
Ultrasonic 1
10
Ultrasonic 2
500
Ultrasonic 3
2372
Ultrasonic 2
499
Ultrasonic 3
2372
Ultrasonic 1
10
Ultrasonic 2
500
Ultrasonic 3
2373
Ultrasonic 1
9
Ultrasonic 2
500

```

รูปที่ 4.7 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 10 เซนติเมตร

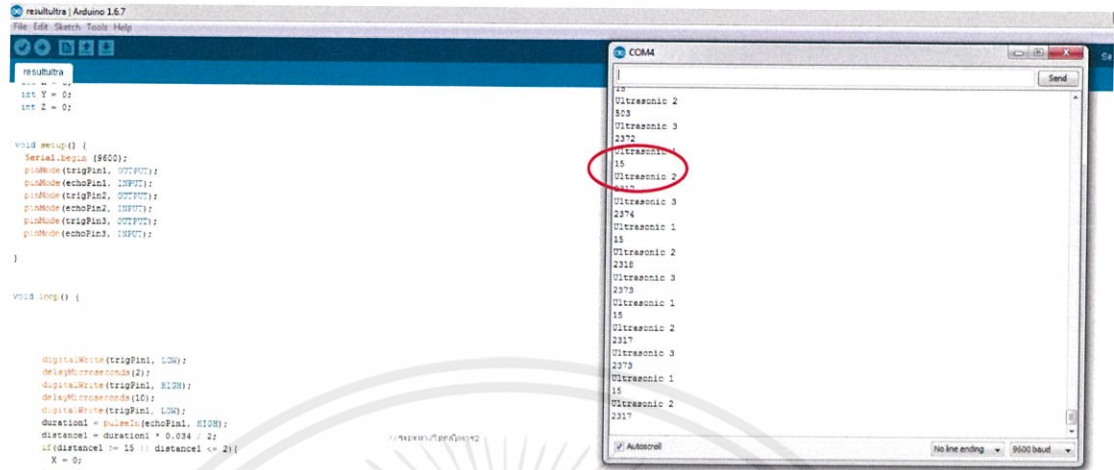


รูปที่ 4.8 กราฟแสดงค่าพัลส์ของเอกโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 10 เซนติเมตร

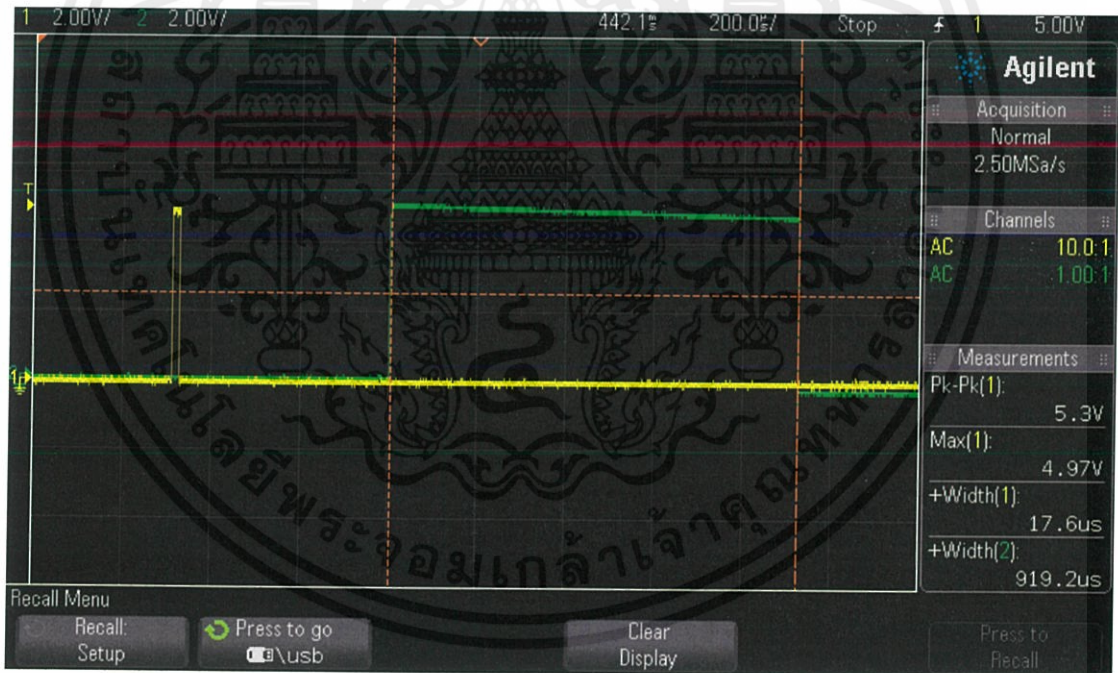
ค่าพัลส์ที่วัดได้ = 644 us ค่าที่คำนวณได้ = 10.95 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระยะ 15 เซนติเมตร



รูปที่ 4.9 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 15 เซนติเมตร



รูปที่ 4.10 กราฟแสดงค่าพัลส์ของเอโคและทรिकของเซนเซอร์อัลตราโซนิกตัวที่ 1 ที่ระยะ 15 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 919.2 us ค่าที่คำนวณได้ = 15.63 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 5 เซนติเมตร

```

resultultra
int Y = 0;
int Z = 0;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
}

void loop() {
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = duration1 * 0.034 / 2;
  if (distance1 >= 15 || distance1 <= 2) {
    X = 0;
  }
  else {
    X = 1;
  }
  Serial.println("Ultrasonic 1");
}

```

Serial Monitor Output:

```

Ultrasonic 2
6
Ultrasonic 3
2374
Ultrasonic 1
6
Ultrasonic 2
5
Ultrasonic 1
2373
Ultrasonic 1
21
Ultrasonic 2
5
Ultrasonic 3
2372
Ultrasonic 1
21
Ultrasonic 2
6
Ultrasonic 3
2372
Ultrasonic 1
21
Ultrasonic 2
6

```

รูปที่ 4.11 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 5 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 338 us ค่าที่คำนวณได้ = 5.75 เซนติเมตร



รูปที่ 4.12 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 5 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 338 us ค่าที่คำนวณได้ = 5.75 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระยะ 10 เซนติเมตร

```

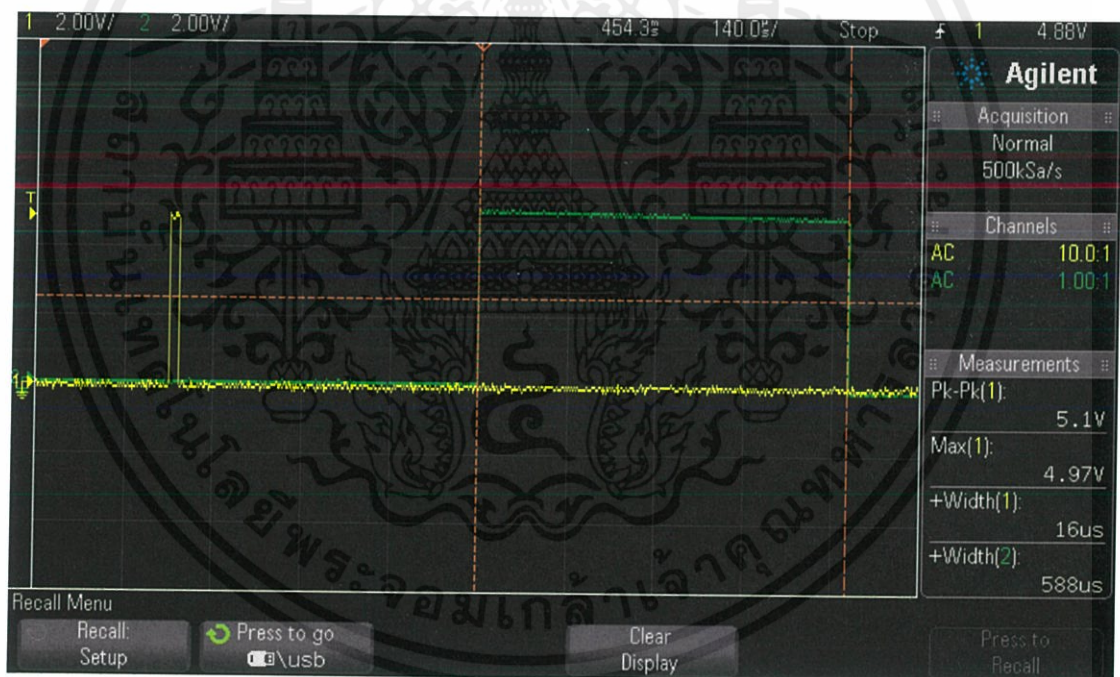
resultultra | Arduino 1.6.7
File Edit Sketch Tools Help

resultultra
Serial.begin(9600);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);

void setup() {
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);
}

void loop() {
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = duration1 * 0.034 / 2;
  if (distance1 >= 15 || distance1 <= 2) {
    X = 0;
  }
  else {
    X = 1;
  }
  Serial.println("Ultrasonic 1");
}
  
```

รูปที่ 4.13 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 10 เซนติเมตร



รูปที่ 4.14 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 10 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 588 us ค่าที่คำนวณได้ = 9.99 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระยะ 15 เซนติเมตร

```

resultura | Arduino 1.6.7
File Edit Sketch Tools Help
resultura
---
int Y = 0;
int Z = 0;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);
}

void loop() {
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  duration1 = duration1 * 0.0294 / 2;
  if (distance1 >= 15 || distance1 <= 2) {
    X = 0;
  }
  else {
    X = 1;
  }
  Serial.println("Ultrasonic 1");
}

```

Serial Monitor Output:

```

Ultrasonic 3
2371
Ultrasonic 1
24
Ultrasonic 2
15
Ultrasonic 3
228
Ultrasonic 1
11
Ultrasonic 2
15
Ultrasonic 3
228
Ultrasonic 1
22
Ultrasonic 2
15
Ultrasonic 3
2372
Ultrasonic 1
24
Ultrasonic 2
15
Ultrasonic 3
2371

```

รูปที่ 4.15 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 15 เซนติเมตร



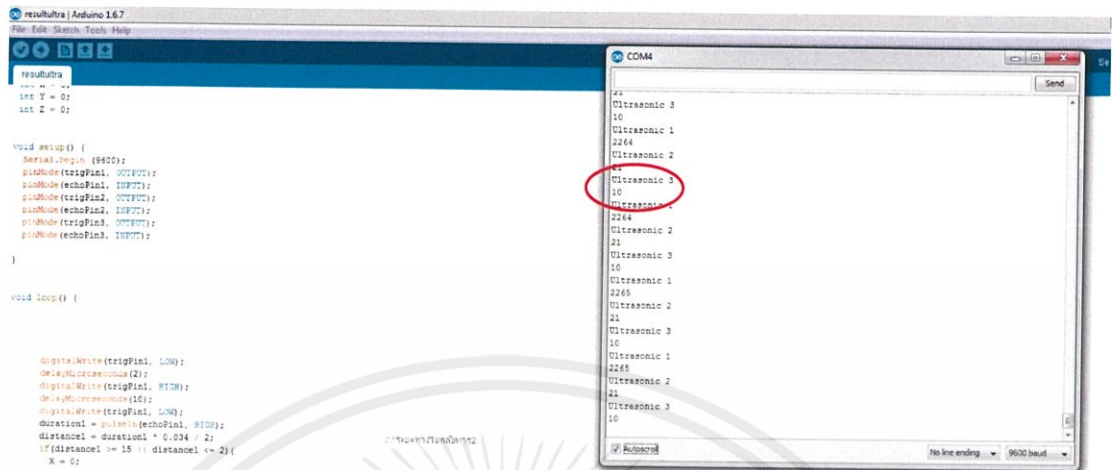
รูปที่ 4.16 กราฟแสดงค่าพัลส์ของเอคโคและทริกของเซนเซอร์อัลตราโซนิกตัวที่ 2 ที่ระยะ 15 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 938.4us ค่าที่คำนวณได้ = 15.95เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ที่ระยะ 10 เซนติเมตร



รูปที่ 4.19 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 10 เซนติเมตร

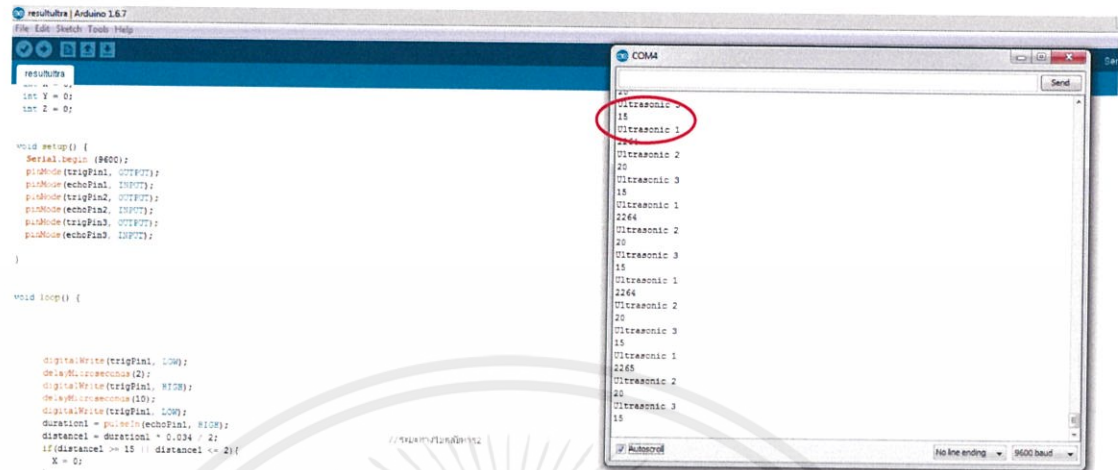


รูปที่ 4.20 กราฟแสดงค่าพัลส์ของเอคโคและทริคของเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 10 เซนติเมตร

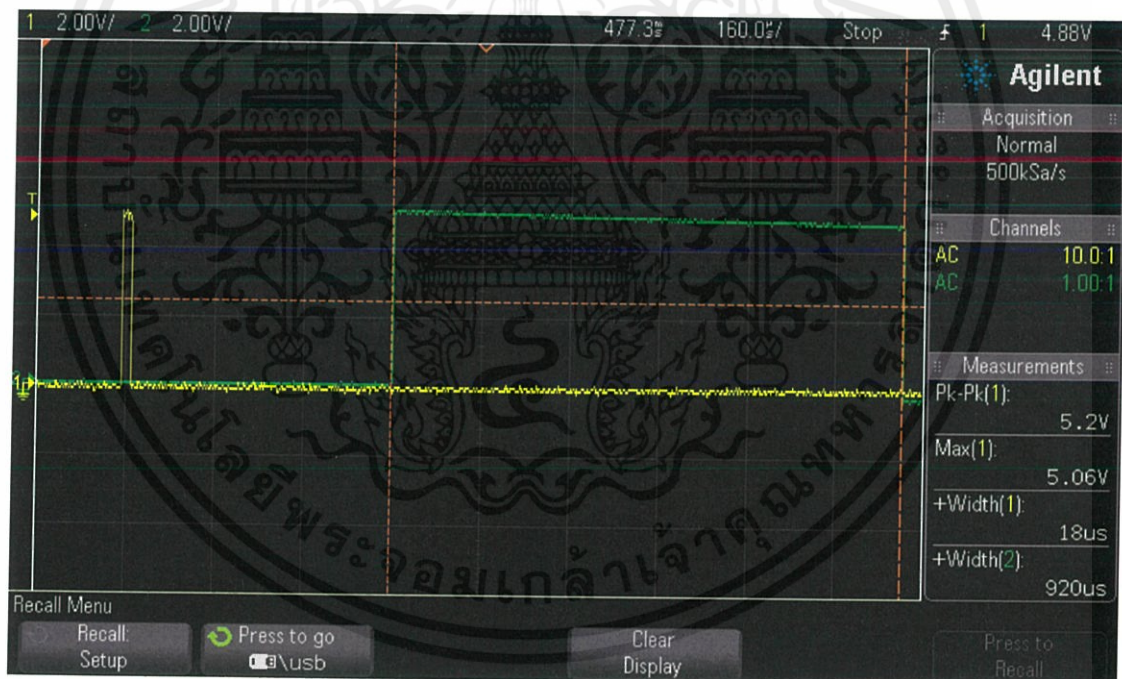
ค่าพัลส์ที่วัดได้ = 610 us ค่าที่คำนวณได้ = 10.37 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระยะ 15 เซนติเมตร



รูปที่ 4.21 Serial monitor แสดงระยะทางเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 15 เซนติเมตร



รูปที่ 4.22 กราฟแสดงค่าพัลส์ของแอกโคและทรคของเซนเซอร์อัลตราโซนิกตัวที่ 3 ที่ระยะ 15 เซนติเมตร

ค่าพัลส์ที่วัดได้ = 920 us ค่าที่คำนวณได้ = 15.64 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางผลการทดลอง

ตารางที่ 4.1 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 1

ระยะทาง(cm)	ค่าพัลส์ที่วัดได้(us)	ค่าที่คำนวณได้(cm)	ค่าความคลาดเคลื่อน (%)
5	313.6	5.33	6.6%
10	644	10.95	9.5%
15	919.2	15.63	4.2%

ตารางที่ 4.2 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 2

ระยะทาง(cm)	ค่าพัลส์ที่วัดได้(us)	ค่าที่คำนวณได้(cm)	ค่าความคลาดเคลื่อน (%)
5	338	5.75	15%
10	588	9.99	0.1%
15	938.4	15.95	6.3%

ตารางที่ 4.3 ตารางผลการทดลองของเซนเซอร์อัลตราโซนิกตัวที่ 3

ระยะทาง(cm)	ค่าพัลส์ที่วัดได้(us)	ค่าที่คำนวณได้(cm)	ค่าความคลาดเคลื่อน (%)
5	314	5.34	6.8%
10	610	10.37	3.7%
15	920	15.64	4.3%

ตัวอย่างการคำนวณ

จากสูตร

$$\text{ระยะทาง} = \frac{\text{ระยะเวลา} \times \text{อัตราเร็วเสียงในอากาศ}}{2} \quad (4.1)$$

$$\text{ระยะทาง} = \frac{314 \times 0.034}{2}$$

$$\text{ระยะทาง} = 5.33 \text{ เซนติเมตร}$$

การคำนวณค่าความคลาดเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสูตร

$$\% \text{ความคลาดเคลื่อน} = \left| \frac{\text{ค่าที่วัดได้}-\text{ค่าจริง}}{\text{ค่าจริง}} \right| \times 100\% \quad (4.2)$$

$$\% \text{ความคลาดเคลื่อน} = \left| \frac{5.33-5}{5} \right| \times 100\%$$

$$\% \text{ความคลาดเคลื่อน} = 6.6\%$$

### 4.3 ทดสอบการรับ - ส่งสัญญาณของ บลูทูธ

#### 4.3.1 วิธีการทดลอง

1. เชื่อมต่อบลูทูธระหว่างโทรศัพท์มือถือกับเครื่องดูดฝุ่น
2. ทดสอบกดปุ่ม เติมน้ำถอยหลังแล้วสังเกตว่าการทำงานของเครื่องดูดฝุ่นสอดคล้องกับการกดปุ่มหรือไม่

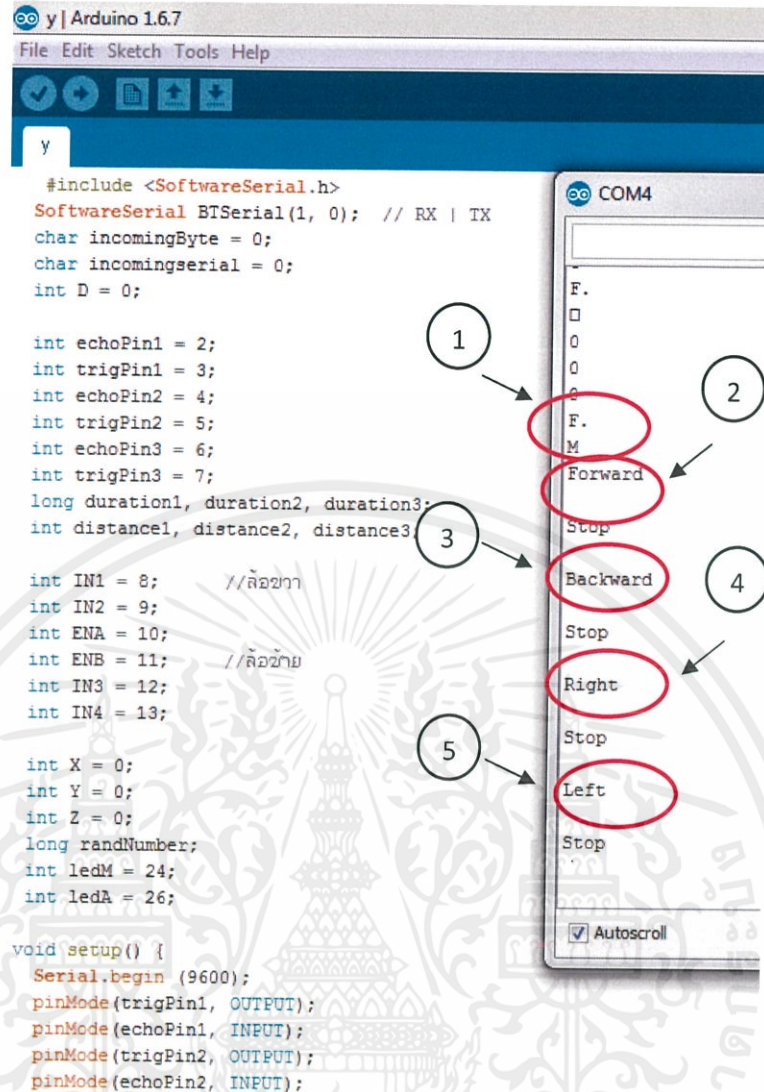
#### 4.3.2 ผลการทดลอง

เมื่อกดปุ่ม



รูปที่ 4.23 หน้าต่างรายชื่อบลูทูธ

เมื่อกดปุ่มบลูทูธจะปรากฏหน้าต่างรายชื่อบลูทูธขึ้นมาให้เลือก



รูปที่ 4.24 Serial monitor

1. เมื่อกดปุ่ม  จะเปลี่ยนเป็นโหมดแมนนวล
2. เมื่อกดปุ่ม  เครื่องดูดฝุ่นจะเดินหน้า
3. เมื่อกดปุ่ม  เครื่องดูดฝุ่น จะถอยหลัง
4. เมื่อกดปุ่ม  เครื่องดูดฝุ่นจะเลี้ยวขวา
5. เมื่อกดปุ่ม  เครื่องดูดฝุ่นจะเลี้ยวซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการทดลอง

จากการทดลองเพื่อทดสอบการทำงานของเครื่องดูดฝุ่น พบว่าเครื่องดูดฝุ่นสามารถทำงานได้ตามที่ต้องการ ทั้งสองระบบ คือระบบอัตโนมัติ เครื่องดูดฝุ่นสามารถเดินรอบห้องและหลบหลีกสิ่งกีดขวางได้เอง ส่วนระบบแมนนวล ผู้ใช้สามารถควบคุมเครื่องดูดฝุ่นให้เคลื่อนที่ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ได้ตามต้องการโดยการใช้แอปพลิเคชันในโทรศัพท์มือถือ ที่เชื่อมต่อกับเครื่องดูดฝุ่นผ่านระบบบลูทูธ ควบคุมโดยการกดปุ่มให้เครื่องดูดฝุ่นเคลื่อนที่ไปในทิศทางต่างๆ และสามารถกดเลือกระบบการทำงานของเครื่องดูดฝุ่นได้ ผ่านแอปพลิเคชันในโทรศัพท์มือถือ โดยทั้งสองระบบสามารถดูดฝุ่นได้อย่างมีประสิทธิภาพ

จากการทดลองทดสอบการดูดฝุ่นของเครื่องดูดฝุ่น เครื่องดูดฝุ่นสามารถดูดฝุ่นเข้าไปเก็บไว้ในกล่องเก็บฝุ่นได้ ส่วนการทดสอบการรับส่งของเซนเซอร์อัลตราโซนิกที่ระยะต่างๆพบว่า ค่าพัลส์ที่วัดได้เมื่อนำมาคำนวณตามสูตรแล้วได้ค่าใกล้เคียงกับค่าที่ทำการวัด ซึ่งระยะทางที่เซนเซอร์อัลตราโซนิกวัดได้ส่งผลโดยตรงต่อการทำงานของเครื่องดูดฝุ่นในการหลบสิ่งกีดขวาง และการทดสอบการรับส่งสัญญาณบลูทูธระหว่างโทรศัพท์มือถือกับเครื่องดูดฝุ่น พบว่าแอปพลิเคชันทำงานได้ถูกต้อง แต่เมื่อกดปุ่มการเคลื่อนที่ในทิศทางต่างๆพบว่าเครื่องดูดฝุ่นจะทำงานหลังจากกดปุ่มประมาณ 1 วินาที

### 5.2 ปัญหาและอุปสรรค

จากการทดลองพบปัญหาคือ

- การทำโครงของเครื่องใช้เวลานานกว่าที่กำหนดเนื่องจาก การออกแบบมีความซับซ้อนมาก
- ชิ้นงาน 3มิติ ที่พิมพ์ออกมามีความคลาดเคลื่อนจากที่ออกแบบ เนื่องจากค่าความผิดพลาดของเครื่องพิมพ์ ทำให้เกิดปัญหาเมื่อทำการประกอบเครื่องดูดฝุ่น

### 5.3 ข้อเสนอแนะ

- เครื่องดูดฝุ่นควรมีน้ำหนักเบากว่าเดิม
- ควรศึกษาเครื่องพิมพ์ 3มิติ และเครื่อง laser cut ให้ดีเนื่องจากตัวเครื่องมีค่าความผิดพลาดเวลาออกแบบจะได้สามารถทำการเผื่อค่าความผิดพลาดนั้นได้ เพื่อให้ได้ชิ้นงานตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] มอเตอร์ไฟฟ้ากระแสตรง [ออนไลน์]. 12 กุมภาพันธ์ 2562. เข้าถึงได้จาก : [http://www.cre-engineer.com/images/pulldown\\_1304840984/DC%20electric%20motors.pdf](http://www.cre-engineer.com/images/pulldown_1304840984/DC%20electric%20motors.pdf)
- [2] Arduino [ออนไลน์]. 13 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>
- [3] Arduino MEGA 2560 R3 [ออนไลน์]. 13 กุมภาพันธ์ 2562. เข้าถึงได้จาก : [https://www.gravitechthai.com/product\\_detail.php?d=55](https://www.gravitechthai.com/product_detail.php?d=55)
- [4] PWM [ออนไลน์]. 20 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>
- [5] L298N [ออนไลน์]. 20 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://sea.banggood.com/Wholesale-Dual-H-Bridge-DC-Stepper-Motor-Drive-Controller-Board-Module-Arduino-L298N-p-42826.html>
- [6] อัลตราโซนิกเซนเซอร์ [ออนไลน์]. 20 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://www.ioxhop.com/article/56/เปรียบเทียบโมดูลวัดระยะด้วยคลื่นอัลตราโซนิก รุ่นต่าง-ๆ-พร้อมวิธีใช้งาน>
- [7] บลูทูธ [ออนไลน์]. 20 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://www.siamphone.com/news/bluetooth/page.htm>
- [8] HC-05 [ออนไลน์]. 20 กุมภาพันธ์ 2562. เข้าถึงได้จาก : <https://www.ioxhop.com/article/26/การใช้งาน-bluetooth-module-hc05-แบบละเอียด>



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## อุปกรณ์สำหรับทำเครื่องดูดฝุ่น

ไมโครคอนโทรลเลอร์ Arduino MEGA 2560	1
ชุดขับกระแส L298N	1
เซนเซอร์อัลตราโซนิก HC-SR04	3
บลูทูธโมดูล HC-05	1
มอเตอร์กระแสตรง 12 V (มอเตอร์เกียร์)	2
มอเตอร์กระแสตรง 12 V	1
แบตเตอรี่ 12 V 1300mAh	1
กล่องเก็บฝุ่น	1
แปรงปัดฝุ่น	1
พัดลมดูดอากาศขนาด 5V 0.4A	1
สายไฟขนาดต่างๆ	
ท่อหด	

## คำสั่งที่ใช้ในการควบคุม

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial BTSerial(1, 0); // RX | TX
```

```
char incomingByte = 0;
```

```
char incomingserial = 0;
```

```
int D = 0;
```

```

int echoPin1 = 5;

int trigPin1 = 6;

int echoPin2 = 30;

int trigPin2 = 28;

int echoPin3 = 34 ;

int trigPin3 = 32;

long duration1, duration2, duration3;

int distance1, distance2, distance3;

int ENA = 8;

int IN1 = 9; //ล้อขวา

int IN2 = 10;

int IN3 = 11;

int IN4 = 12;

int ENB = 13; //ล้อซ้าย

int ENAV = 2;

int IN1V = 3;

int IN2V = 4;

int FAN = 7;

```

```

int n = 0;

int m = 0;

int U = 0;

int V = 0;

int W = 0;

long randNumber;

int ledM = 24;

int ledA = 26;

void setup() {

  Serial.begin (9600);

  pinMode(trigPin1, OUTPUT);

  pinMode(echoPin1, INPUT);

  pinMode(trigPin2, OUTPUT);

  pinMode(echoPin2, INPUT);

  pinMode(trigPin3, OUTPUT);

  pinMode(echoPin3, INPUT);

  pinMode(ENA, OUTPUT);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  pinMode(IN3, OUTPUT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(ENAV, OUTPUT);

pinMode(IN1V, OUTPUT);

pinMode(IN2V, OUTPUT);

pinMode(ledM, OUTPUT);

pinMode(ledA, OUTPUT);

randomSeed(analogRead(0));

pinMode(22, OUTPUT); // this pin will pull the HC-05 pin 34 (key
pin) HIGH to switch module to AT mode
digitalWrite(22, HIGH);

BTSerial.begin(9600); // HC-05 default speed in AT command more
หรือ 9600

//D = 0;
}

void loop() {

if (Serial.available() > 0){

incomingByte = Serial.read();

switch (incomingByte){

```

```

case 'X':

n = n+1;

if ((n%2) == 0){

    digitalWrite(FAN, LOW);

    Serial.println("F off");

}

else{

    digitalWrite(FAN, HIGH);

    Serial.println("F on");

}

break;

case 'Y':

m = m+1;

if ((m%2) == 0 ){

    digitalWrite(IN1V, LOW);

    digitalWrite(IN2V, LOW);

    analogWrite(ENAV, 0);

    Serial.println("M off");

}

else{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(IN1V, LOW);

digitalWrite(IN2V, HIGH);

analogWrite(ENAV, 255);

Serial.println("M on");

}

break;

case 'S': // stop all motor

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, 0);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, 0);

Serial.println("Stop\n");

break;

case 'F': // turn it on going forward

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, 170);

digitalWrite(IN3, LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(IN4, HIGH);

analogWrite(ENB, 200);

Serial.println("Forward\n");

break;

case 'B':                // turn it on going backward

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, 110);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, 200);

Serial.println("Backward\n");

break;

case 'R':                // turn right

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, 0);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, 200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("Right\n");

break;

case 'L':                // turn left

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, 170);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, 0);

Serial.println("Left\n");

break;

case 'A':

digitalWrite(ledA, HIGH);

digitalWrite(ledM, LOW);

Serial.println("A\n");

D = 1;

while(D == 1){

    digitalWrite(trigPin1, LOW);

    delayMicroseconds(2);

    digitalWrite(trigPin1, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

duration1 = pulseIn(echoPin1, HIGH);

distance1 = duration1 * 0.0347362 / 2;           //ระยะทางไปกลับหาร2

if((distance1 > 15) || (distance1 <= 0)){

    U = 0;

}

else{

    U = 1;

    Serial.println(distance1);

}

Serial.println(U);

delay(100);

digitalWrite(trigPin2, LOW);

delayMicroseconds(2);

digitalWrite(trigPin2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin2, LOW);

duration2 = pulseIn(echoPin2, HIGH);

distance2 = duration2 * 0.0347362 / 2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((distance2 >= 15) || (distance2 <= 0)){

    V = 0;

}

else{

    V = 1;

    Serial.println(distance2);

}

Serial.println(V);

delay(100);

digitalWrite(trigPin3, LOW);

delayMicroseconds(2);

digitalWrite(trigPin3, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin3, LOW);

duration3 = pulseIn(echoPin3, HIGH);

distance3 = duration3 * 0.0347362 / 2;

if((distance3 >= 15) || (distance3 <= 0)){

    W = 0;

}

else{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

W = 1;

Serial.println(distance3);

}

Serial.println(W);

delay(100);

if(( U == 0 ) && ( V == 0 ) && ( W == 0 )){

Serial.println("F.");

digitalWrite(IN1,HIGH);

digitalWrite(IN2,LOW);

analogWrite(ENA,170);

digitalWrite(IN3,LOW);

digitalWrite(IN4,HIGH);

analogWrite(ENB,200);

}

if(( U == 0 ) && ( V == 0 ) && ( W == 1 )){

Serial.println("F.");

digitalWrite(IN1,HIGH);

digitalWrite(IN2,LOW);

analogWrite(ENA,170);

digitalWrite(IN3,LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(IN4,HIGH);

analogWrite(ENB,200);

}

if(( U == 1 ) && ( V == 0 ) && ( W == 0 )){

Serial.println("F.");

digitalWrite(IN1,HIGH);

digitalWrite(IN2,LOW);

analogWrite(ENA,170);

digitalWrite(IN3,LOW);

digitalWrite(IN4,HIGH);

analogWrite(ENB,200);

}

if(( U == 1 ) && ( V == 0 ) && ( W == 1 )){

Serial.println("F.");

digitalWrite(IN1,HIGH);

digitalWrite(IN2,LOW);

analogWrite(ENA,170);

digitalWrite(IN3,LOW);

digitalWrite(IN4,HIGH);

analogWrite(ENB,200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(( U == 1 ) && ( V == 1 ) && ( W == 0 )){

  Serial.println("R.");

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  analogWrite(ENA,170);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,LOW);

  analogWrite(ENB,0);

  delay(5000);
}

if(( U == 0 ) && ( V == 1 ) && ( W == 1 )){

  Serial.println("L.");

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,LOW);

  analogWrite(ENA,0);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

  analogWrite(ENB,200);

  delay(5000);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(( U == 1 ) && ( V == 1 ) && ( W == 1 )){

  Serial.println("S.");

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,LOW);

  analogWrite(ENA,0);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,LOW);

  analogWrite(ENB,0);
}

if( ( U == 0 ) && ( V == 1 ) && ( W == 0 )){

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,LOW);

  analogWrite(ENA,0);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,LOW);

  analogWrite(ENB,0);

  delay(100);

  randomNumber = random(2);

  if(randomNumber == 0){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("L.");

digitalWrite(IN1,LOW);           //left

digitalWrite(IN2,LOW);

analogWrite(ENA,0);

digitalWrite(IN3,HIGH);

digitalWrite(IN4,LOW);

analogWrite(ENB,200);

delay(5000);
}

if(randNumber == 1){
Serial.println("R.");

digitalWrite(IN1,LOW);           //Right

digitalWrite(IN2,HIGH);

analogWrite(ENA,170);

digitalWrite(IN3,LOW);

digitalWrite(IN4,LOW);

analogWrite(ENB, 0);

delay(5000);

}

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

incomingserial = Serial.read();

Serial.println(incomingserial);

if(incomingserial == 'M'){

    digitalWrite(ledA, LOW);

    digitalWrite(ledM, HIGH);

    digitalWrite(IN1, LOW);

    digitalWrite(IN2, LOW);

    analogWrite(ENA, 0);

    digitalWrite(IN3, LOW);

    digitalWrite(IN4, LOW);

    analogWrite(ENB, 0);

    D = 0;

}

U = 0;

V = 0;

W = 0;

}

break;

case 'M':

    digitalWrite(ledA, LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(ledM, HIGH);

Serial.println("M\n");

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, 0);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, 0);

break;

default:

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, 0);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, 0);

}

}

}
```

## Ultrasonic Ranging Module HC - SR04

### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

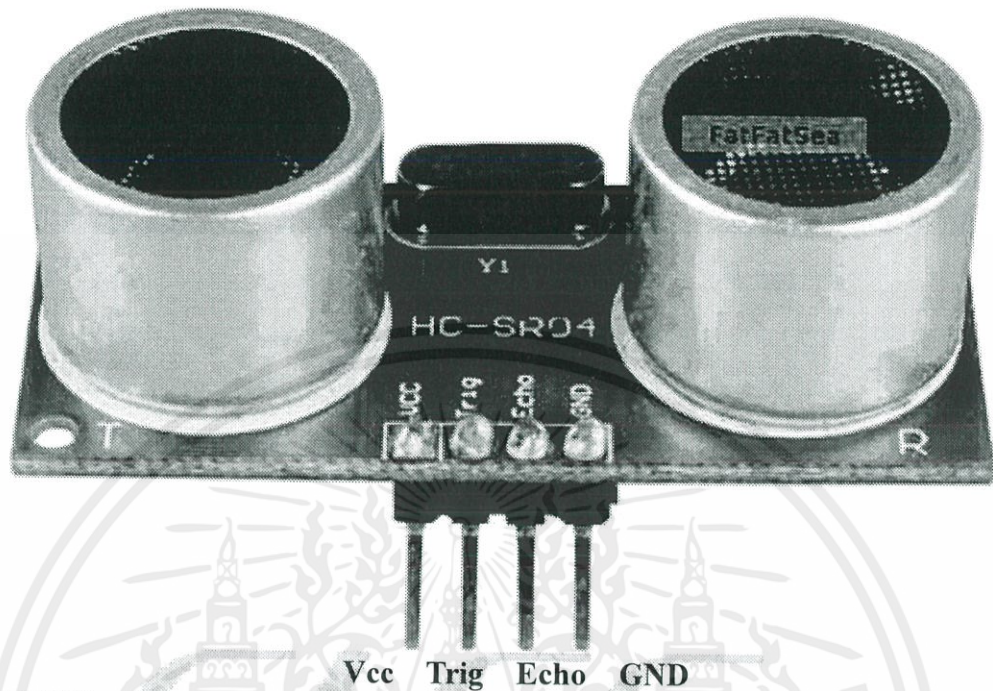
Test distance = (high level time×velocity of sound (340M/S) / 2,

### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

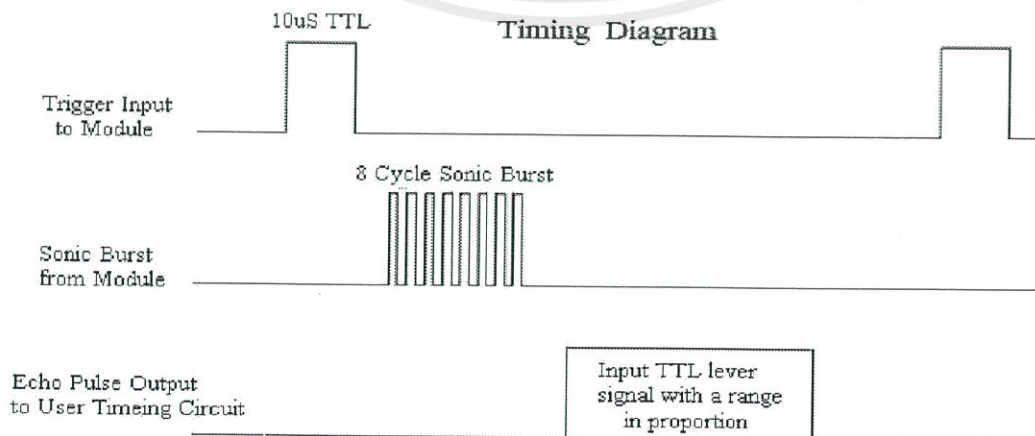
### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



## Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion . You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{S} / 58 = \text{centimeters}$  or  $\mu\text{S} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

---

## Attention:

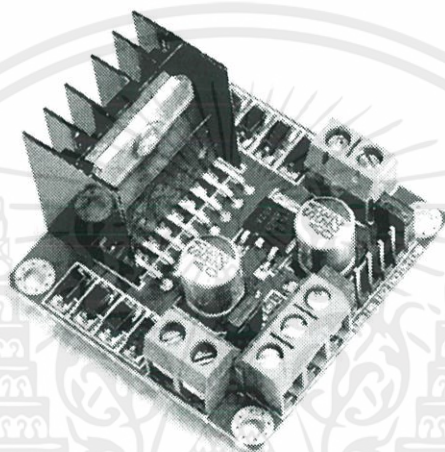
- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

[www.Elecfreaks.com](http://www.Elecfreaks.com)



## L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

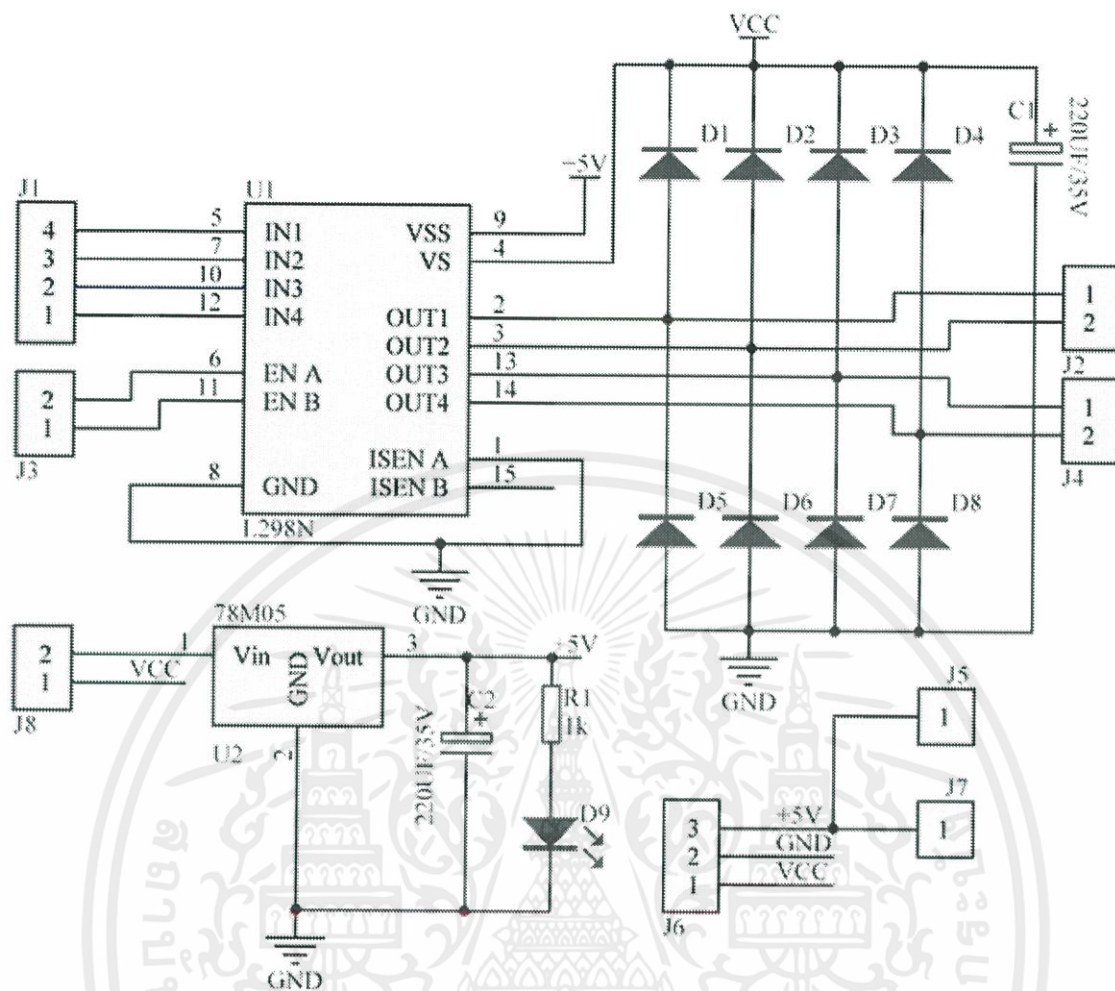


**SKU: MDU-1049**

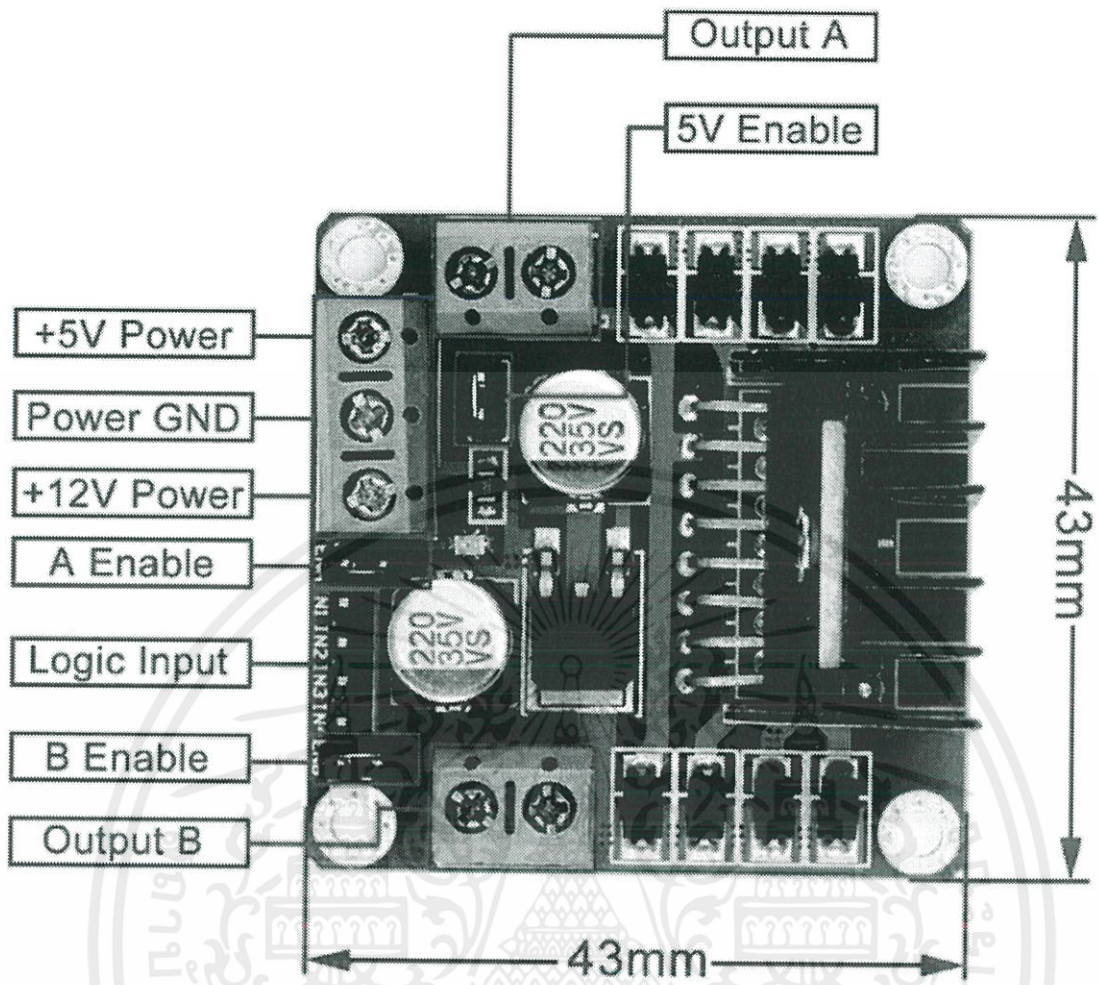
### Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low:  $-0.3V \leq V_{in} \leq 1.5V$ .
- High:  $2.3V \leq V_{in} \leq V_{ss}$ .
- Enable signal input voltage range :
  - Low:  $-0.3 \leq V_{in} \leq 1.5V$  (control signal is invalid).
  - High:  $2.3V \leq V_{in} \leq V_{ss}$  (control signal active).
- Maximum power consumption: 20W (when the temperature  $T = 75\text{ }^{\circ}\text{C}$ ).
- Storage temperature:  $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$ .
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

## Schematic Diagram:



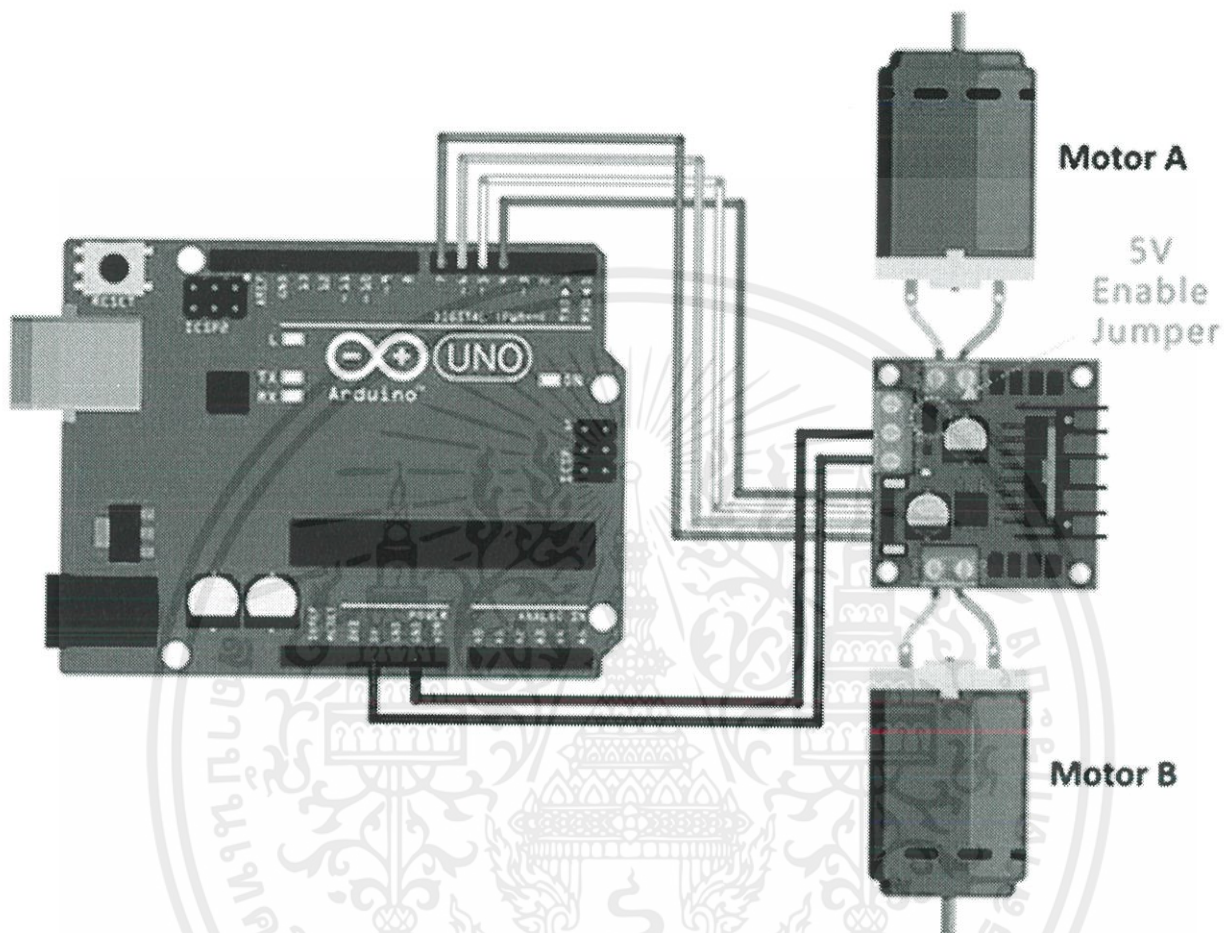
## Board Dimension & Pins Function:



## Connection Examples:

### Controlling 2-DC Motor with +5V Arduino onboard Power Supply:

Below is the circuit connection use the on-board +5V power supply from Arduino board, and should be done without the 5V Enable Jumper on (Active 5V). This connection can drive two 5V DC motors simultaneously.



### Sketch Listing:

Copy and paste the sketch below to Arduino IDE and upload to Arduino Uno/Mega board.

```
/*=====
// Author      : Handson Technology
// Project     : Arduino Uno
// Description  : L298N Motor Driver
// Source-Code : L298N_Motor.ino
// Program:    Control 2 DC motors using L298N H Bridge Driver
//=====
*/

// Definitions Arduino pins connected to input H Bridge
int IN1 = 4;
int IN2 = 5;
int IN3 = 6;
int IN4 = 7;

void setup()
{
  // Set the output pins
```

```

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
}

void loop()
{
  // Rotate the Motor A clockwise
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  delay(2000);
  // Motor A
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  delay(500);

  // Rotate the Motor B clockwise
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  delay(2000);
  // Motor B
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
  delay(500);

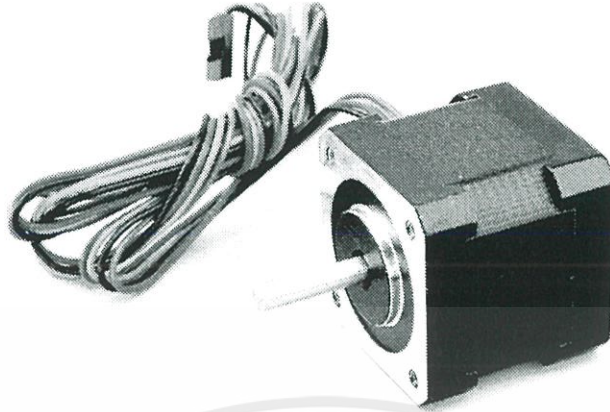
  // Rotates the Motor A counter-clockwise
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  delay(2000);
  // Motor A
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  delay(500);

  // Rotates the Motor B counter-clockwise
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  delay(2000);
  // Motor B
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
  delay(500);
}

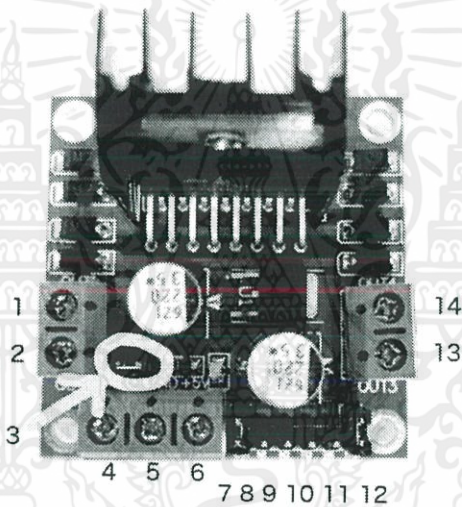
```

## Controlling Stepper Motor

In this example we have a typical NEMA-17 stepper motor with four wires:



The key to successful stepper motor control is identifying the wires - that is which one is which. You will need to determine the A+, A-, B+ and B- wires. With our example motor these are red, green, yellow and blue. Now let's get the wiring done.



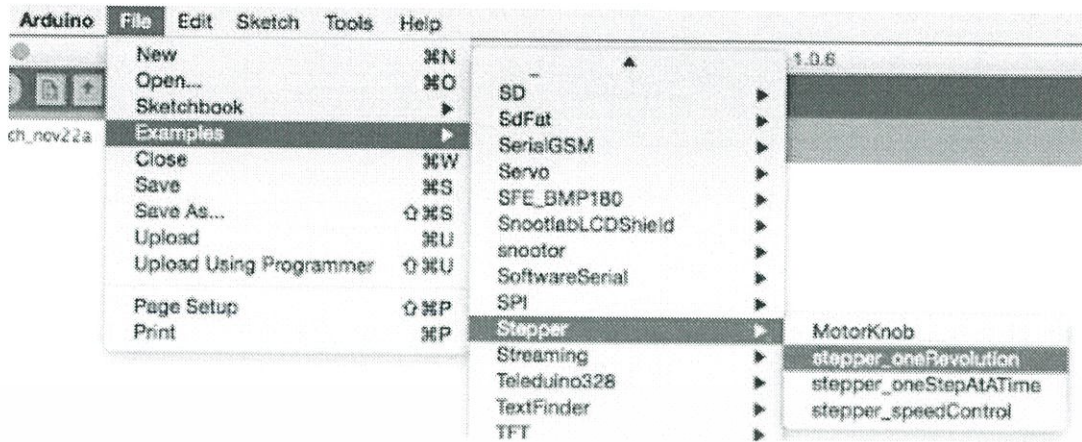
Connect the A+, A-, B+ and B- wires from the stepper motor to the module connections 1, 2, 13 and 14 respectively. Place the jumpers included with the L298N module over the pairs at module points 7 and 12. Then connect the power supply as required to points 4 (positive) and 5 (negative/GND).

Once again if your stepper motor's power supply is less than 12V, fit the jumper to the module at point 3 which gives you a neat 5V power supply for your Arduino.

Next, connect L298N module pins IN1, IN2, IN3 and IN4 to Arduino digital pins D8, D9, D10 and D11 respectively. Finally, connect Arduino GND to point 5 on the module, and Arduino 5V to point 6 if sourcing 5V from the module.

Controlling the stepper motor from your sketches is very simple, thanks to the *Stepper* Arduino library included with the Arduino IDE as standard.

To demonstrate your motor, simply load the “*stepper\_oneRevolution*” sketch that is included with the *Stepper* library, for example:



Finally, check the value for

```
const int stepsPerRevolution = 200;
```

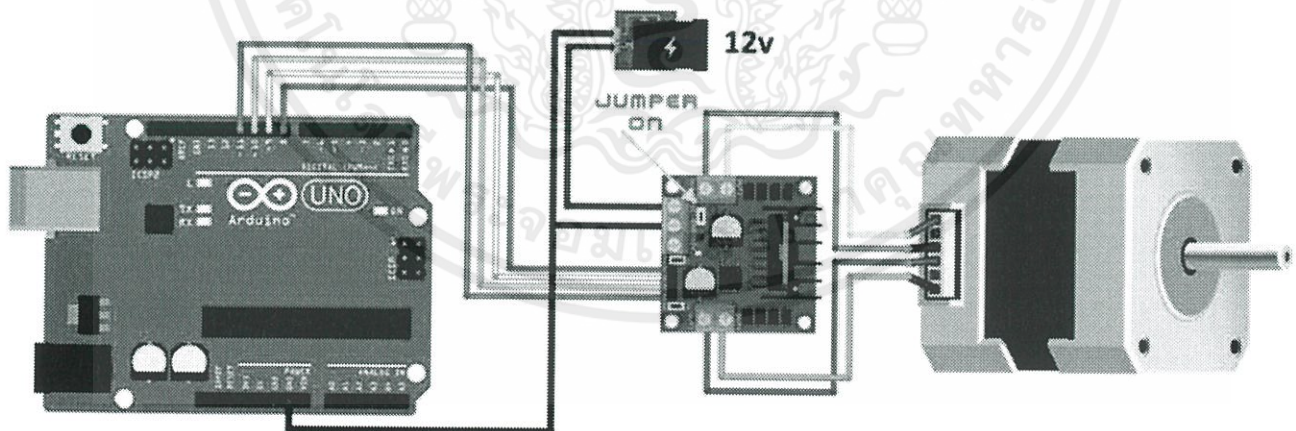
in the sketch and change the 200 to the number of steps per revolution for your stepper motor, and also the speed which is preset to 60 RPM in the following line:

```
myStepper.setSpeed(60);
```

Now you can save and upload the sketch, which will send your stepper motor around one revolution, then back again. This is achieved with the function

```
myStepper.step(stepsPerRevolution); // for clockwise  
myStepper.step(-stepsPerRevolution); // for anti-clockwise
```

### Connection for the sketch “*stepper\_oneRevolution*”:



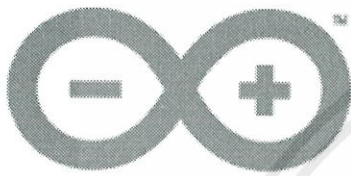
### Web Resources:

# RobotShop

[www.robotshop.com](http://www.robotshop.com)

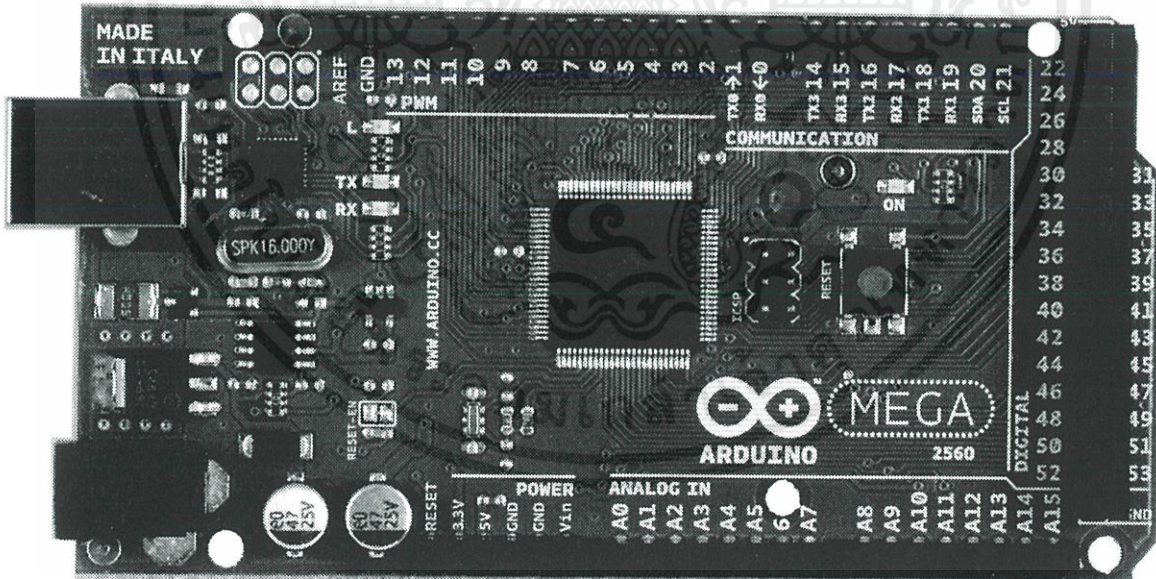


La robotique à votre service! - Robotics at your service!

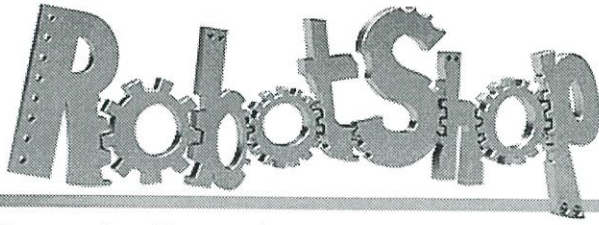


ARDUINO

## Arduino Mega 2560 Datasheet



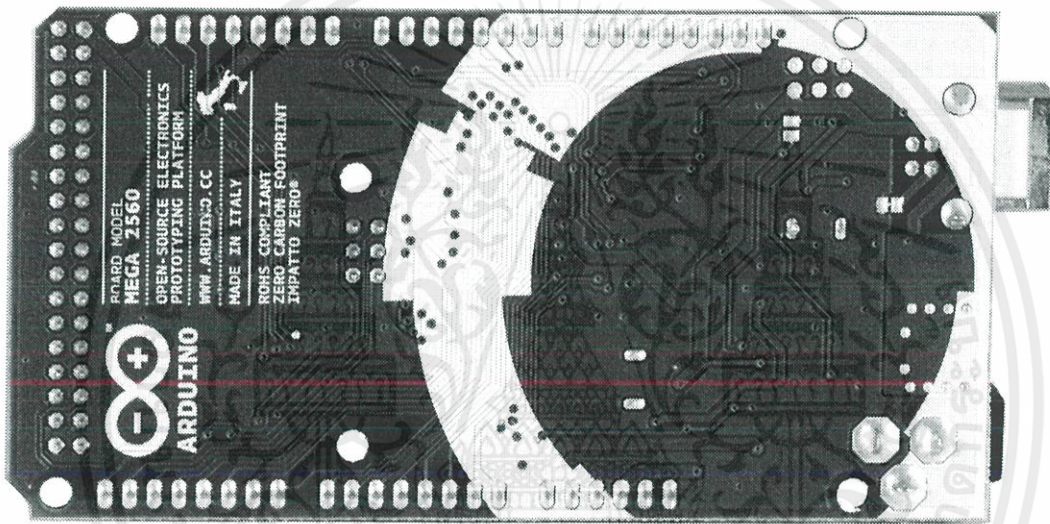
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



www.robotshop.com



La robotique à votre service! - Robotics at your service!



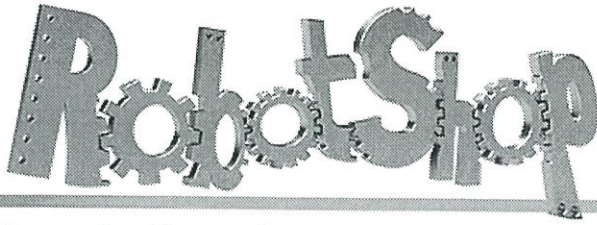
## Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



www.robotshop.com



La robotique à votre service! - Robotics at your service!

Schematic: [arduino-mega2560-schematic.pdf](#)

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



La robotique à votre service! - Robotics at your service!

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

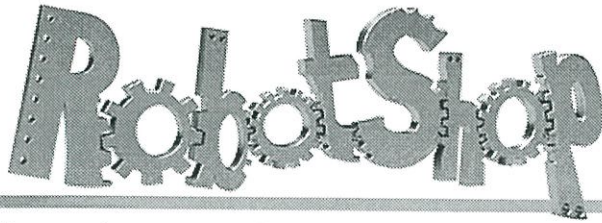
## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



www.robotshop.com



La robotique à votre service! - Robotics at your service!

value, the LED is on, when the pin is LOW, it's off.

- **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Decimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

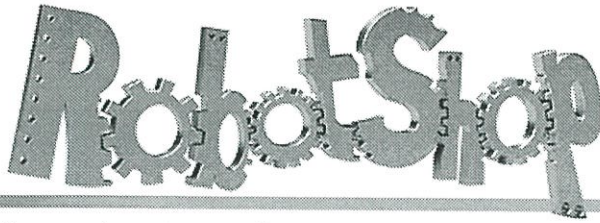
A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

## Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It



www.robotshop.com



La robotique à votre service! - Robotics at your service!

communicates using the original STK500 protocol ([reference](#), [C header files](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

## Automatic (Software) Reset

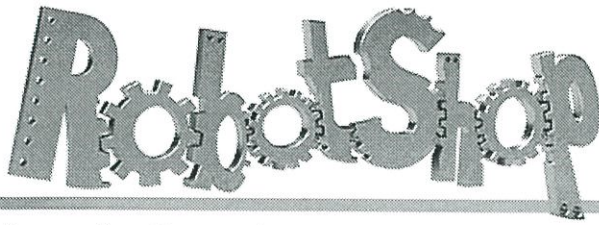
Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



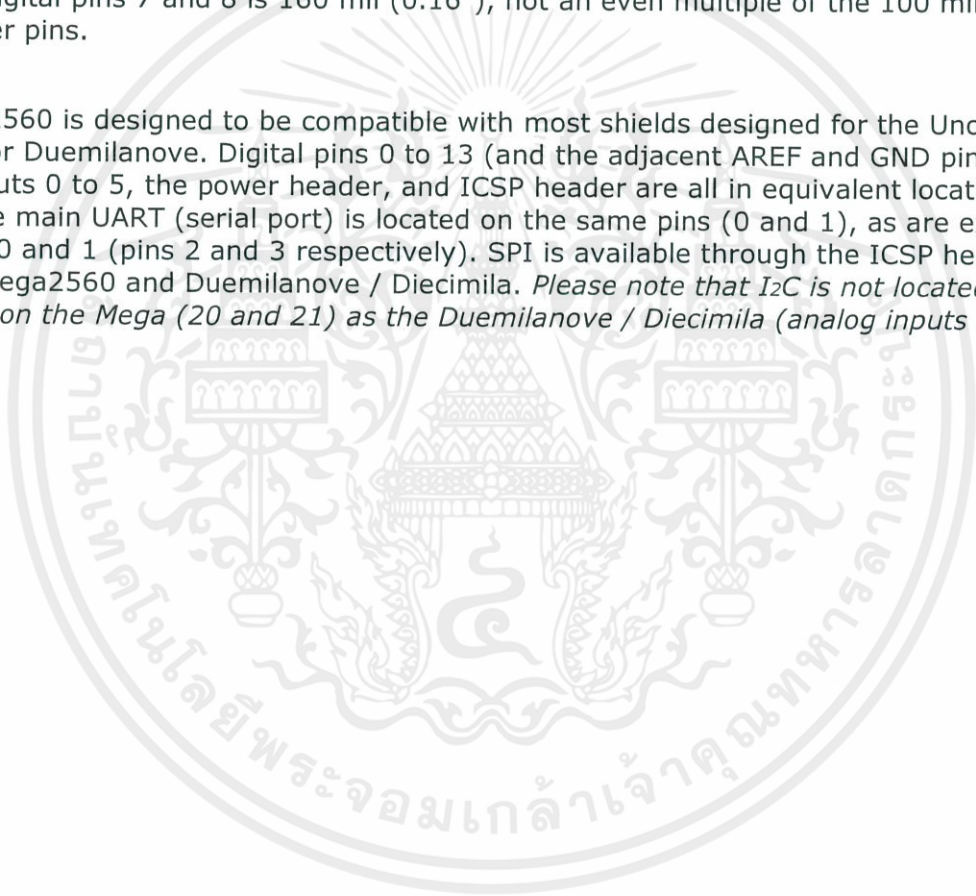
[www.robotshop.com](http://www.robotshop.com)



**La robotique à votre service! - Robotics at your service!**

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

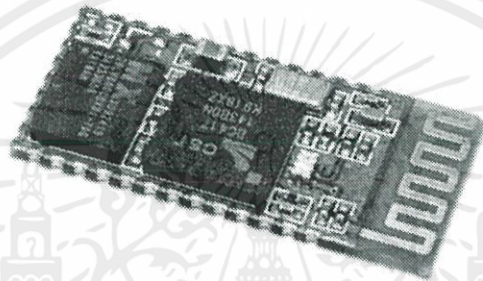


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# HC-05

## -Bluetooth to Serial Port Module

### Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

### Specifications

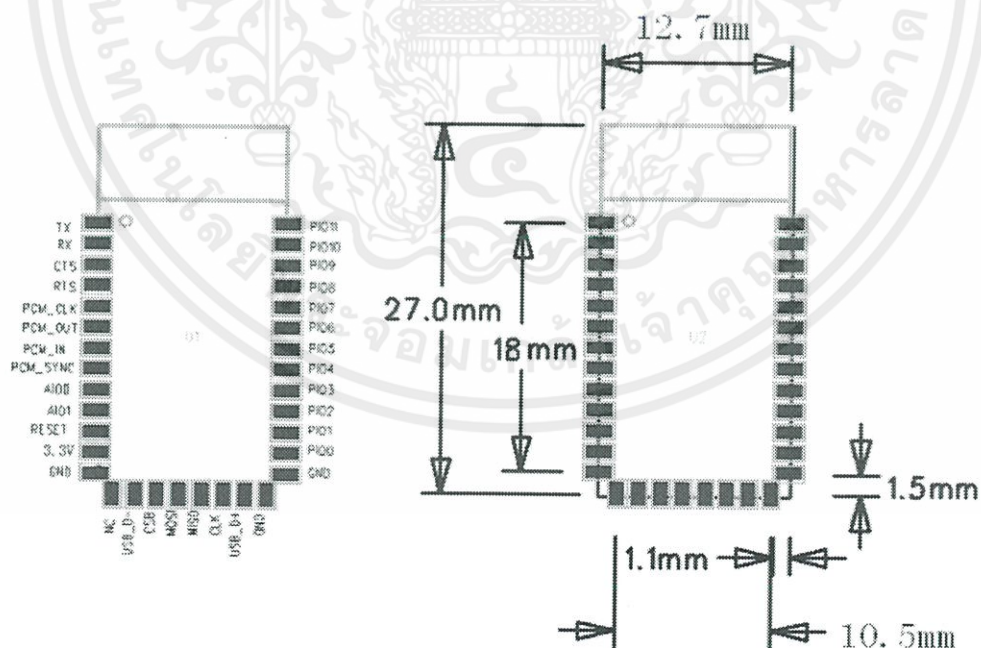
#### Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

## Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

## Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13	VSS	Ground pot	
	21			
	22			
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	

PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

<b>RESETB</b>	11	CMOS input with weak internal pull-up	Reset if low.input debounced so must be low for >5MS to cause a reset	
<b>UART_RTS</b>	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
<b>UART_CTS</b>	3	CMOS input with weak internal pull-down	UART clear to send, active low	
<b>UART_RX</b>	2	CMOS input with weak internal pull-down	UART Data input	
<b>UART_TX</b>	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
<b>SPI_MOSI</b>	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
<b>SPI_CSB</b>	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
<b>SPI_CLK</b>	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
<b>SPI_MISO</b>	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
<b>USB_-</b>	15	Bi-Directional		

USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	

## AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

```
AT+VERSION?\r\n
+VERSION:2.0-20100601
OK
```

## 4. Restore default

Command	Respond	Parameter
AT+ORGL	OK	-

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

## 5. Get module address

Command	Respond	Parameter
AT+ADDR?	+ADDR:<Param> OK	Param: address of Bluetooth module

Bluetooth address: NAP: UAP : LAP

Example:

AT+ADDR?\r\n

+ADDR:1234:56:abcdef

OK

## 6. Set/Check module name:

Command	Respond	Parameter
AT+NAME=<Param>	OK	Param: Bluetooth module name (Default :HC-05)
AT+NAME?	+NAME:<Param> OK (/FAIL)	

Example:

AT+NAME=HC-05\r\n set the module name to "HC-05"

OK

AT+NAME=ITeadStudio\r\n

OK

AT+NAME?\r\n

+NAME:ITeadStudio

OK

## 7. Get the Bluetooth device name:

Command	Respond	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK 2. FAIL	Param1,Param 2 : the address of Bluetooth device

Example: (Device address 00:02:72:od:22:24, name: ITead)

AT+RNAME? 0002, 72, od2224\r\n

+RNAME:ITead

OK

## 8. Set/Check module mode:

Command	Respond	Parameter
AT+ROLE=<Param>	OK	Param: 0- Slave
AT+ROLE?	+ROLE:<Param>	

	OK	1-Master 2-Slave-Loop
--	----	--------------------------

## 9. Set/Check device class

Command	Respond	Parameter
AT+CLASS=<Param>	OK	Param: Device Class
AT+ CLASS?	1. +CLASS:<Param> OK 2. FAIL	

## 10. Set/Check GIAC (General Inquire Access Code)

Command	Respond	Parameter
AT+IAC=<Param>	1.OK 2. FAIL	Param: GIAC (Default : 9e8b33)
AT+IAC	+IAC:<Param> OK	

Example:

AT+IAC=9e8b3f\r\n

OK

AT+IAC?\r\n

+IAC: 9e8b3f

OK

## 11. Set/Check -- Query access patterns

Command	Respond	Parameter
AT+INQM=<Param>,<Param2>,<Param3>	1.OK 2. FAIL	Param: 0— inquiry_mode_standard 1— inquiry_mode_rssi Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s)
AT+ INQM?	+INQM : <Param>,<Param2>,<Param3> OK	

Example:

AT+INQM=1,9,48\r\n

OK

AT+INQM\r\n

+INQM:1, 9, 48

OK

## 12. Set/Check PIN code:

Command	Respond	Parameter
AT+PSWD=<Param>	OK	Param: PIN code (Default 1234)
AT+ PSWD?	+ PSWD : <Param> OK	

## 13. Set/Check serial parameter:

Command	Respond	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: Baud Param2: Stop bit
AT+ UART?	+UART=<Param>,<Param2>,<Param3> OK	Param3: Parity

Example:

```
AT+UART=115200, 1,2,\r\n
OK
AT+UART?
+UART:115200,1,2
OK
```

## 14. Set/Check connect mode:

Command	Respond	Parameter
AT+CMODE=<Param>	OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop
AT+ CMODE?	+ CMODE:<Param> OK	

## 15. Set/Check fixed address:

Command	Respond	Parameter
AT+BIND=<Param>	OK	Param: Fixed address (Default 00:00:00:00:00:00)
AT+ BIND?	+ BIND:<Param> OK	

Example:

```
AT+BIND=1234, 56, abcdef\r\n
OK
AT+BIND?\r\n
+BIND:1234:56:abcdef
OK
```

## 16. Set/Check LED I/O

Command	Respond	Parameter
AT+POLAR=<Param1>,<Param2>	OK	Param1: 0- PIO8 low drive LED 1- PIO8 high drive LED
AT+ POLAR?	+ POLAR=<Param1>,<Param2> OK	

		Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED
--	--	--

## 17. Set PIO output

Command	Respond	Parameter
AT+PIO=<Param1>,<Param2>	OK	Param1: PIO number Param2: PIO level 0- low 1- high

Example:

1. PIO10 output high level

AT+PIO=10, 1\r\n

OK

## 18. Set/Check – scan parameter

Command	Respond	Parameter
AT+IPSCAN=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Query time interval
AT+IPSCAN?	+IPSCAN:<Param1>,<Param2>,<Param3>,<Param4> OK	Param2: Query duration Param3: Paging interval Param4: Call duration

Example:

AT+IPSCAN =1234,500,1200,250\r\n

OK

AT+IPSCAN?

+IPSCAN:1234,500,1200,250

## 19. Set/Check – SHIFF parameter

Command	Respond	Parameter
AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Max time Param2: Min time
AT+ SNIFF?	+SNIFF:<Param1>,<Param2>,<Param3>,<Param4> OK	Param3: Retry time Param4: Time out

## 20. Set/Check security mode

Command	Respond	Parameter
AT+SENM=<Param1>,<Param2>	1. OK 2. FAIL	Param1: 0—sec_mode0+off
AT+ SENM?	+ SENM:<Param1>,<Param2>	1—sec_mode1+non_se

	OK	cure 2—sec_mode2_service 3—sec_mode3_link 4—sec_mode_unknow n Param2: 0—hci_enc_mode_off 1—hci_enc_mode_pt_t o_pt 2—hci_enc_mode_pt_t o_pt_and_bcast
--	----	--

### 21. Delete Authenticated Device

Command	Respond	Parameter
AT+PMSAD=<Param>	OK	Param: Authenticated Device Address

Example:

```
AT+PMSAD =1234,56,abcdef\r\n
OK
```

### 22. Delete All Authenticated Device

Command	Respond	Parameter
AT+RMAAD	OK	-

### 23. Search Authenticated Device

Command	Respond	Parameter
AT+FSAD=<Param>	1. OK 2. FAIL	Param: Device address

### 24. Get Authenticated Device Count

Command	Respond	Parameter
AT+ADCN?	+ADCN: <Param> OK	Param: Device Count

### 25. Most Recently Used Authenticated Device

Command	Respond	Parameter
AT+MRAD?	+ MRAD: <Param> OK	Param: Recently Authenticated Device Address

### 26. Get the module working state

Command	Respond	Parameter
---------	---------	-----------

AT+ STATE?	+ STATE: <Param> OK	Param: "INITIALIZED" "READY" "PAIRABLE" "PAIRED" "INQUIRING" "CONNECTING" "CONNECTED" "DISCONNECTED" "NUKNOW"
------------	------------------------	--

## 27. Initialize the SPP profile lib

Command	Respond	Parameter
AT+INIT	1. OK 2. FAIL	-

## 28. Inquiry Bluetooth Device

Command	Respond	Parameter
AT+INQ	+INQ: <Param1> , <Param2> , <Param3> .... OK	Param1: Address Param2: Device Class Param3 : RSSI Signal strength

Example:

```

AT+INIT\r\n
OK
AT+IAC=9e8b33\r\n
OK
AT+CLASS=0\r\n
AT+INQM=1,9,48\r\n
At+INQ\r\n
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
  
```

## 28. Cancel Inquiring Bluetooth Device

Command	Respond	Parameter
AT+ INQC	OK	-

## 29. Equipment Matching

Command	Respond	Parameter
AT+PAIR=<Param1>,<Param2>	1. OK 2. FAIL	Param1: Device Address Param2: Time out

## 30. Connect Device

Command	Respond	Parameter
AT+LINK=<Param>	1. OK 2. FAIL	Param: Device Address

Example:

AT+FSAD=1234,56,abcdef\r\n

OK

AT+LINK=1234,56,abcdef\r\n

OK

## 31. Disconnect

Command	Respond	Parameter
AT+DISC	1. +DISC:SUCCESS OK 2. +DISC:LINK_LOSS OK 3. +DISC:NO_SLC OK 4. +DISC:TIMEOUT OK 5. +DISC:ERROR OK	Param: Device Address

## 32. Energy-saving mode

Command	Respond	Parameter
AT+ENSNIFF=<Param>	OK	Param: Device Address

## 33. Exerts Energy-saving mode

Command	Respond	Parameter
AT+ EXSNIFF =<Param>	OK	Param: Device Address

## Revision History

Rev.	Description	Release date
v1.0	Initial version	7/18/2010

