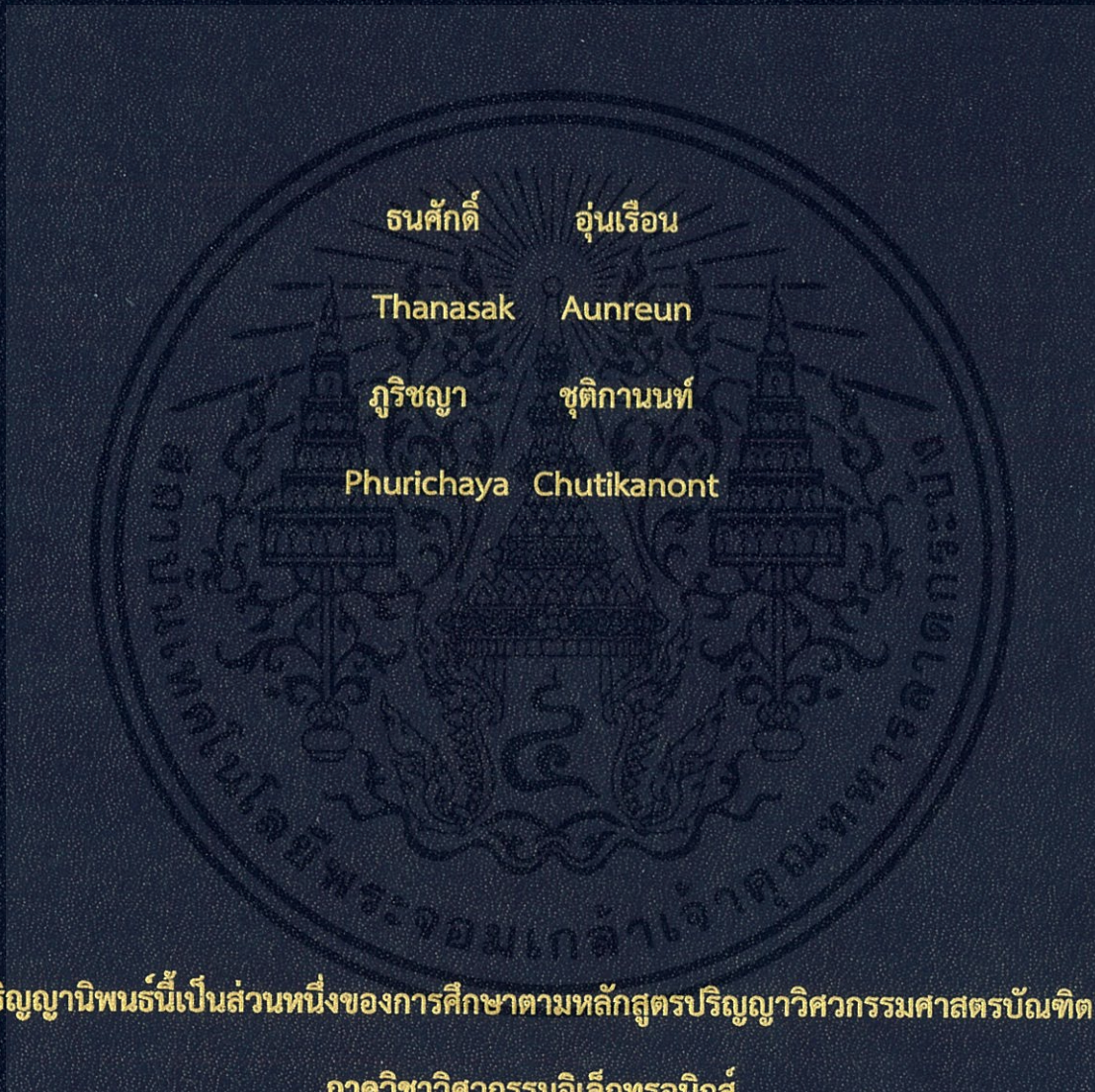


หุ่นยนต์ส่งของ

Messenger Robot



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

หุ่นยนต์ส่งของ

Messenger Robot



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ส่งของ

Messenger Robot

โดย

ธนศักดิ์ อุ่นเรือน

Thanasak Aunreun

ภุริชญา ชุติกานนท์

Phurichaya Chutikanont

อาจารย์ที่ปรึกษาโครงการ

รองศาสตราจารย์ ด็อกเตอร์ สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ส่งของ

Messenger Robot

ผู้จัดทำ นายชนศักดิ์ อุ่นเรือน รหัสประจำตัว 58010523

นางสาวภูริชญา ชุติกานนท์ รหัสประจำตัว 58010986

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(รองศาสตราจารย์ ด็อกเตอร์ สุรพันธุ์ เอื้อไพบูลย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์ส่งของ	
นักศึกษา	นายธนศักดิ์ อุ่นเรือน	รหัสประจำตัว 58010523
	นางสาวภริชญา ชุติกานนท์	รหัสประจำตัว 58010986
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2561	
อาจารย์ที่ปรึกษาโครงการ	รองศาสตราจารย์ ด็อกเตอร์ สุรพันธ์ เอื้อไพบูลย์	

บทคัดย่อ

โครงการนี้นำเสนอ หุ่นยนต์ส่งของ (Messenger Robot) มีจุดประสงค์ทำขึ้นเพื่ออำนวยความสะดวกในการส่งเอกสารหรือสิ่งของภายในอาคาร โดยป้อนข้อมูลของจุดหมายปลายทางที่ต้องการ โดยใช้ระบบ RFID (Radio Frequency identification) ในการระบุตำแหน่งของจุดหมายปลายทาง

Thesis title	Messenger Robot		
Students	Mr. Thanasak Aunreun	Student ID	58010523
	Ms. Phurichaya Chutikaont	Student ID	58010986
Degree	Bachelor of Engineering		
Program	Electronics Engineering		
Year	2018		
Thesis Advisor	Associate Prof. Dr. Surapan Airphaiboon		

ABSTRACT

‘A Messenger Robot’, this project is intended to facilitate an indoor transportation and position itself by Radio Frequency (RFID) to move automatically after identifying the destination.

กิตติกรรมประกาศ

ปริญญานิพนธ์หัวข้อเรื่อง “หุ่นยนต์ส่งของ” สามารถผ่านไปได้ด้วยดีจากความช่วยเหลือและการให้คำปรึกษาจาก รองศาสตราจารย์ ตีอกเตอร์ สุรพันธุ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษาที่ได้คอยช่วยเหลือในเรื่องอุปกรณ์และให้ความรู้รวมถึงการแก้ปัญหาต่าง ๆ ในระหว่างการผลิตชิ้นงานออกมา รวมถึงขอขอบคุณทุกคนที่มีส่วนเหลือและกำลังใจจากที่ต่าง ๆ ที่มีให้โครงการนี้สามารถผ่านไปได้

สุดท้ายนี้คณะผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจและผู้นำผลงานนี้ไปใช้งานได้



นายธนศักดิ์ อุ่นเรือน

นางสาวภุริชญา ชุติกานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวข้อ	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์	1
1.3 สมมติฐานของการศึกษา	1
1.4 ขอบเขตของการศึกษา	1
บทที่ 2 หลักการและทฤษฎี	2
2.1 Radio Frequency Identification (RFID)	2
2.1.1 รายละเอียดและส่วนประกอบของ RFID	2
2.1.1.1 Transponder	2
2.1.1.2 Interrogator	3
2.1.1.3 ฮาร์ดแวร์ หรือ ระบบที่ใช้ประมวลผล	4
2.1.2 ประเภทของป้ายอาร์เอฟไอดี (RFID)	4
2.1.2.1 ชนิด Passive	4
2.1.2.2 ชนิด Active	5
2.1.3 การทำงานของ RFID - 1-Bit Transponder (EAS)	5
2.1.4 Proximity ID-12 RFID 125 KHz Reader Breakout Board V2	6
2.2 Arduino	6
2.2.1 Arduino Mega 2560 R3	7

สารบัญ(ต่อ)

หัวข้อ	หน้า
2.2 Arduino (ต่อ)	6
2.2.2 การเขียนโปรแกรมบน Arduino	7
2.3 Ultrasonic Sensor	9
2.4 Motor Drive Module L298N	10
2.5 Transistor BD139	12
2.6 Limit Switch	13
บทที่ 3 การออกแบบ	13
3.1 ขั้นตอนการดำเนินงาน	14
3.2 ออกแบบการทดลอง	14
3.2.1 ทดสอบตัวมอเตอร์ขับเคลื่อนล้อ	14
3.2.2 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board	14
3.2.3 นำหุ่นไปทดสอบจริงตามที่กำหนดไว้	14
3.2.3.1 ทดสอบการเคลื่อนที่ของหุ่นยนต์ในการเคลื่อนที่ไปตามทางตรง	14
3.2.3.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์	14
3.2.3.3 ทดสอบการระบุจุดหมายของหุ่นยนต์	14
บทที่ 4 การทดลองและผลการทดลอง	16
4.1 ทดสอบตัวมอเตอร์ขับเคลื่อนล้อ	16
4.2 นำอุปกรณ์มาต่อและการทำงานร่วมกันหลังจากเขียนโปรแกรม Arduino แล้วเสร็จ	16
4.2.1 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board	16
4.3 นำหุ่นไปทดสอบจริงตามที่กำหนดไว้	17
4.3.1 ทดสอบการเคลื่อนที่ของหุ่นยนต์ในการเคลื่อนที่ไปตามทางตรง	17

สารบัญ(ต่อ)

หัวข้อ	หน้า
4.3 นำหุ่นไปทดสอบจริงตามที่กำหนดไว้ (ต่อ)	17
4.3.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์	18
4.3.3 ทดสอบการระบุจุดหมายของหุ่นยนต์จาก ID12 และ RFID Tag Card	19
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	20
สรุปผลการทดลอง	20
ข้อเสนอแนะ	20
เอกสารอ้างอิง	21
ภาคผนวก	23



สารบัญรูป

หัวข้อ	หน้า
รูปที่	
2.1 แสดงการทำงานของ RFID	2
2.2 แสดงส่วนประกอบของ RFID	2
2.3 แสดงตัวอย่างเครื่องอ่านป้ายชนิดมือถือ	3
2.4 แสดงตัวอย่างเครื่องอ่านป้าย	3
2.5 ภาพแสดงมาตรฐานของ RFID	4
2.6 แสดง RFID แบบ Passive ชนิดพวงกุญแจ ชนิดแคปซูล และชนิดบัตร	4
2.7 แสดง RFID แบบ Active ชนิดบัตรที่อยู่ในบรรจุภัณฑ์ขนาดเล็ก	5
2.8 ภาพแสดงตัวอย่าง RFID ที่นิยมใช้ในศูนย์การค้า	5
2.9 Proximity ID-12 RFID 125 KHz	6
2.10 แสดงตัวอย่าง Lay out และ Pin ของบอร์ด Arduino	6
2.11 บอร์ด Arduino Mega 2560 R3	7
2.12 แสดงตัวอย่างการใช้งาน Arduino	7
2.13 แสดงตัวอย่างการเลือกหมายเลข Comport ของบอร์ด	8
2.14 แสดงตัวอย่างการอัปโหลดโค้ดและการใช้งานโปรแกรม	8
2.15 ตัวอย่างการเชื่อมต่อของโมดูล ultrasonic Sensor และ Arduino Board	9
2.16 แสดงการวัดระยะทางและสูตรการหาระยะทางระหว่างตัวโมดูลและวัตถุ	9
2.17 แสดงสมการการหาอัตราเร็วของเสียงในอากาศ	10
2.18 ไดอะแกรมการทำงานของเซนเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก	10
2.19 แสดงรายละเอียดของตัวโมดูล L298N	11
2.20 แสดงวงจร H-Bridge ภายในตัวโมดูล L298N	11

สารบัญรูป(ต่อ)

หัวข้อ	หน้า
รูปที่	
2.21 แสดงลักษณะ BD139	12
2.22 แสดงลักษณะวงจรภายในของ BD139	12
2.23 แสดงการต่อวงจรเพื่อใช้งาน BD139	12
2.24 Limit Switch ประเภทหนึ่ง เรียกว่า Micro Switch	13
2.25 ตัวอย่างการต่อ Micro Switch กับ Arduino	13
3.1 Block Diagram แสดง ขั้นตอนการทำงานของหุ่นยนต์	14
3.2 แสดงลำดับการทำงาน (Flow Chart) ของตัวหุ่นยนต์	15
3.3 รูปสำเร็จของ Messenger Robot เมื่อตกแต่งเสร็จแล้ว	15
4.1 มอเตอร์ขับเคลื่อนล้อ	16
4.2 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board	16
4.3 ตัวหุ่นยนต์เมื่อเชื่อมต่อทุกอุปกรณ์เข้าด้วยกันแล้วเสร็จ	17
4.4 พื้นที่การทดสอบการเคลื่อนที่ทางตรงของหุ่นยนต์	17
4.5 แสดงการทดสอบการเคลื่อนที่ของหุ่นยนต์	18
4.6 แสดงการทดสอบการตอบสนองเซนเซอร์ของหุ่นยนต์กับสิ่งกีดขวาง	18
4.7 แสดงการทดสอบการระบุจุดหมายจาก Limit Switch กับ RFID Tag card ของจุดหมายปลายทาง	19
4.8 ทดสอบการหยุดของหุ่นยนต์เมื่อตรวจพบ RFID Tag Card ของจุดหมายปลายทาง	19

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันการส่งข้อมูลทางเอกสารผ่านจดหมายอิเล็กทรอนิกส์นับเป็นสิ่งที่แพร่หลายแต่สิ่งของและเอกสารบางประเภทยังคงจำเป็นต้องมีการส่งแบบจับต้องได้ โดยเฉพาะการส่งสิ่งของหรือเอกสารภายในอาคารหรือภายในสำนักงานจึงเป็นที่มาของโครงการนี้ซึ่งก็คือหุ่นยนต์ที่สามารถส่งสิ่งของภายในอาคารได้ โดยที่ผู้ใช้ไม่จำเป็นต้องส่งสิ่งของให้ถึงที่หมายที่ต้องการด้วยตนเอง

1.2 ความมุ่งหมายและจุดประสงค์ของการศึกษา

ในการทำโครงการครั้งนี้ จัดทำขึ้นเพื่อ

1.2.1 เป็นการฝึกการปฏิบัติงานของเพื่อเป็นพื้นฐานในการปฏิบัติงานและฝึกการทำงานเป็นหมู่คณะ

1.2.2 บูรณาการความรู้ที่ได้จากการเรียนในภาคทฤษฎีนำมาใช้ในการปฏิบัติงานจริง

1.2.3 เสริมสร้างความเข้าใจเกี่ยวกับหลักการทฤษฎีและการประยุกต์ใช้งานเพื่อปฏิบัติจริงของ RFID และการทำงานของ RFID ควบคู่ไปกับ Arduino

1.3 สมมุติฐานการศึกษา

เมื่อได้รับ INPUT เป็นตำแหน่งของจุดหมายปลายทางที่ต้องการส่งสิ่งของของหุ่นยนต์ส่งของจะเคลื่อนที่ไปยังจุดหมายที่ถูกระบุมาใน Input โดยไม่ต้องใช้การบังคับมือ

1.4 ขอบเขตการศึกษา

คุณสมบัติของหุ่นยนต์ส่งของคือตัวของหุ่นยนต์นั้นจะสามารถเคลื่อนที่ไปยังจุดหมายปลายทางที่ถูกป้อนให้เป็น Input ได้อย่างถูกต้อง

บทที่ 2

หลักการและทฤษฎี

2.1 Radio Frequency Identification (RFID)

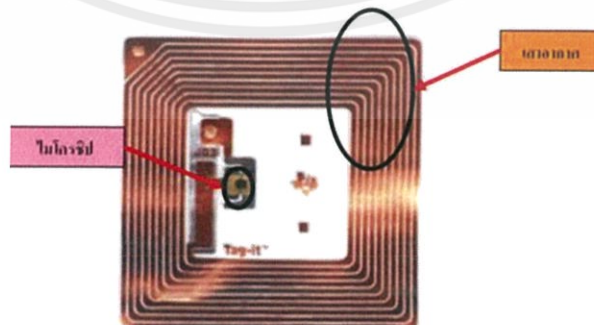
2.1.1 รายละเอียดและส่วนประกอบของ RFID

Radio Frequency Identification คือการระบุข้อมูลของสิ่งต่าง ๆ โดยใช้คลื่นความถี่วิทยุจุดเด่นคือความสามารถในการอ่านข้อมูลได้โดยไม่ต้องมีการสัมผัส อีกทั้งยังแม่นยำและมีความไวในการอ่านข้อมูลที่สูง ซึ่งประกอบด้วย 3 องค์ประกอบหลักได้แก่ ป้าย (Tag, Transponder), เครื่องอ่านป้าย (Reader, Interrogator) และ ฮาร์ดแวร์ หรือระบบที่ใช้ประมวลผล



รูปที่ 2.1 แสดงการทำงานของ RFID

2.1.1.1 Transponder ตัวจัดเก็บและส่งข้อมูล จะอยู่ในรูปฉลากหรือป้าย และเจ้าตัวฉลากหรือป้ายนี้ จะถูกบันทึกข้อมูลที่เกี่ยวข้องกับวัตถุนั้นๆเอาไว้ โดยเรียกทั่วไปว่า "Tag" มันจะทำหน้าที่ส่งสัญญาณหรือข้อมูลที่ถูกบันทึกอยู่ใน Tag ตอบสนองไปยังตัวอ่านข้อมูล



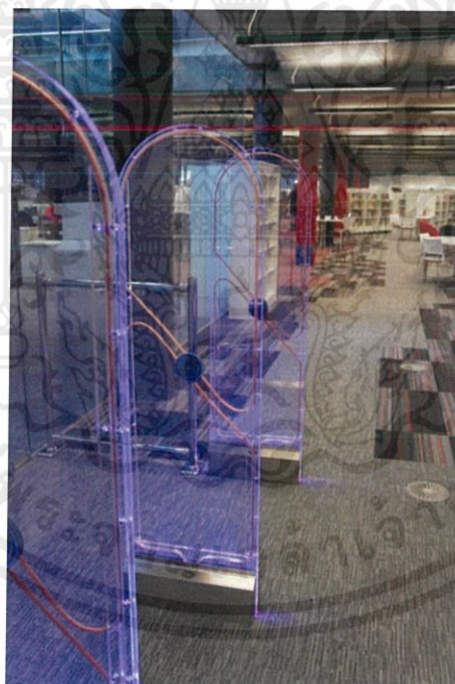
รูปที่ 2.2 แสดงส่วนประกอบของ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.2 Interrogator หรือ เครื่องอ่านป้าย (Reader) มีหน้าที่เชื่อมต่อกับป้ายเพื่อทำการอ่านหรือเขียนข้อมูลลงในป้ายโดยใช้สัญญาณวิทยุ ภายในเครื่องอ่านประกอบด้วยเสาอากาศเพื่อใช้รับ-ส่งสัญญาณ, ภาครับ-ภาคส่งสัญญาณวิทยุ, วงจรควบคุมการอ่าน-เขียนข้อมูล และส่วนที่ติดต่อกับคอมพิวเตอร์ ในส่วนของป้ายนั้น เครื่องอ่านก็จะมีชนิดและรูปร่างที่แตกต่างกันออกไป ทั้งนี้ขึ้นอยู่กับความต้องการในการใช้งาน เช่น แบทเตอร์มือถือ, แบบติดผนัง เป็นต้น



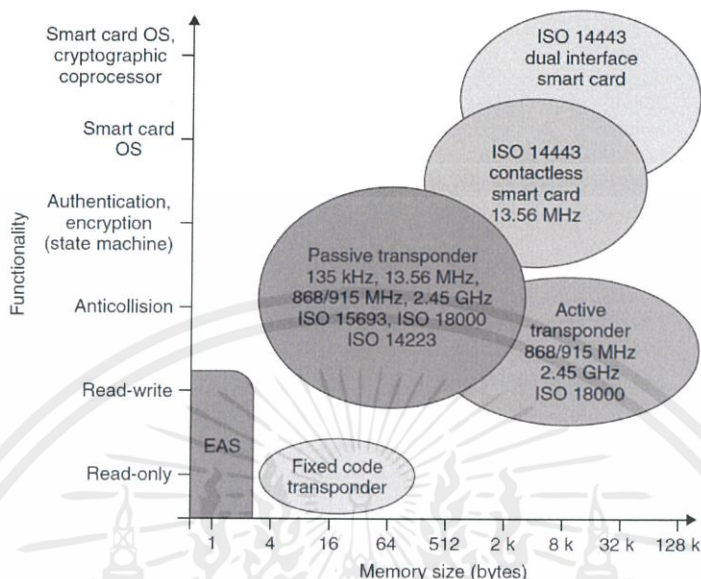
รูปที่ 2.3 แสดงตัวอย่างเครื่องอ่านป้ายชนิดมือถือ



รูปที่ 2.4 แสดงตัวอย่างเครื่องอ่านป้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.3 ฮาร์ดแวร์ หรือ ระบบที่ใช้ประมวลผล ส่วนประมวลผลจากข้อมูลที่ได้รับจากป้าย Tag รวมถึงเป็นส่วนที่ใช้สร้างข้อมูลแล้วส่งไปยังป้าย Tag อีกทั้งยังเป็นที่ใช้เก็บระบบฐานข้อมูล อีกด้วย



รูปที่ 2.5 ภาพแสดงมาตรฐานของ RFID

2.1.2 ประเภทของป้ายอาร์เอฟไอดี (RFID)

2.1.2.1 ชนิด Passive จุดเด่นคือมีขนาดเล็ก เบา และทำงานได้โดยไม่ต้องใช้แหล่งจ่ายไฟจากภายนอกเนื่องจากในตัว Tag นี้มีแหล่งกำเนิดไฟฟ้าจากวงจรกำเนิดไฟฟ้าเหนี่ยวนำอยู่ในตัว ข้อเสียคือ มีระยะการอ่านข้อมูลที่สั้นไม่เกิน 1 เมตร



รูปที่ 2.6 แสดง RFID แบบ Passive ชนิดพวงกุญแจ ชนิดแคปซูล และชนิดบัตร

2.1.2.2 ชนิด Active ข้อดีคือมีระยะการอ่านข้อมูลที่ไกลประมาณ 100 เมตร แต่ข้อเสียคือ ขนาดของเครื่องอ่านมีขนาดใหญ่เนื่องจากแบตเตอรี่ที่มีอายุการใช้งานจำกัดประมาณ 3-7 ปี และการทำงานนั้นต้องอาศัยแหล่งจ่ายไฟจากภายนอก ไม่มีวงจรกำเนิดไฟฟ้าเหนี่ยวนำอยู่ในตัวเหมือนแบบ passive



รูปที่ 2.7 แสดง RFID แบบ Active ชนิดบัตรที่ภายในบรรจุแบตเตอรี่ขนาดเล็ก

2.1.3 การทำงานของ RFID - 1-Bit Transponder (EAS)

ป้าย RFID จะส่งเพียง 0 กับ 1 ไปยังเครื่องอ่านแค่สองสถานะนี้ สำหรับระบุว่า มี Tag หรือไม่มี Tag อยู่ในบริเวณของเครื่องอ่านบัตรเราเรียกการส่งที่มีแค่ 2 สถานะนี้ว่า 1-Bit Transponder หรือ EAS ซึ่งถูกย่อมาจาก Electronic Article Surveillance โดยศูนย์การค้ามักใช้ระบบนี้ติดตั้งตามตู้ต่างๆ เพื่อป้องกันการขโมยสินค้า เนื่องจากป้าย RFID ของระบบ EAS มีขนาดเล็ก นำไปติดตามสินค้าต่างๆ ได้ง่าย



รูปที่ 2.8 ภาพแสดงตัวอย่าง RFID ที่นิยมใช้ในศูนย์การค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

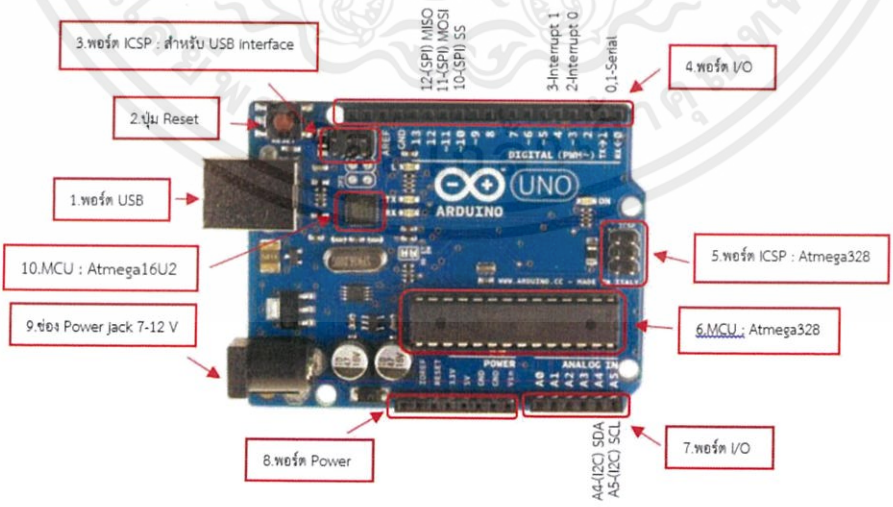
2.1.4 Proximity ID-12 RFID 125 KHz Reader Breakout Board V2 บอร์ด ID-12 พร้อมกับโมดูล RFID Reader ชนิด read-only ใช้ไฟเลี้ยง Vcc 5V ทำงานที่ความถี่ 125KHz โดยไม่จำเป็นต้องต่อเสาอากาศหรือสายอากาศภายนอก (มีเสาอากาศภายในแบบ built-in) สามารถต่อกับ microcontroller ได้โดยตรง อีกทั้งยังมีลำโพง buzzer และไฟ LED เพื่อแสดงสถานะและเป็นการแจ้งเตือนอีกด้วย ข้อเสียคือระยะอ่าน tag ค่อนข้างสั้น ซึ่งอยู่ประมาณ 4-6 เซนติเมตร



รูปที่ 2.9 Proximity ID-12 RFID 125 KHz

2.2 Arduino

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ถูกพัฒนาแบบ Open Source เป็นที่นิยมใช้เนื่องจากใช้งานง่าย ทั้งการดัดแปลงและเพิ่มข้อมูล หรือแม้กระทั่งการพัฒนาต่อยอดของตัวบอร์ดและโปรแกรม ผู้ใช้งานนั้นสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเปรียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย จึงทำให้ Arduino เป็นที่นิยมใช้ และโปรเจกต์นี้ใช้ Arduino Board รุ่น Mega R3 เนื่องจากว่า



รูปที่ 2.10 แสดงตัวอย่าง Lay out และ Pin ของบอร์ด Arduino

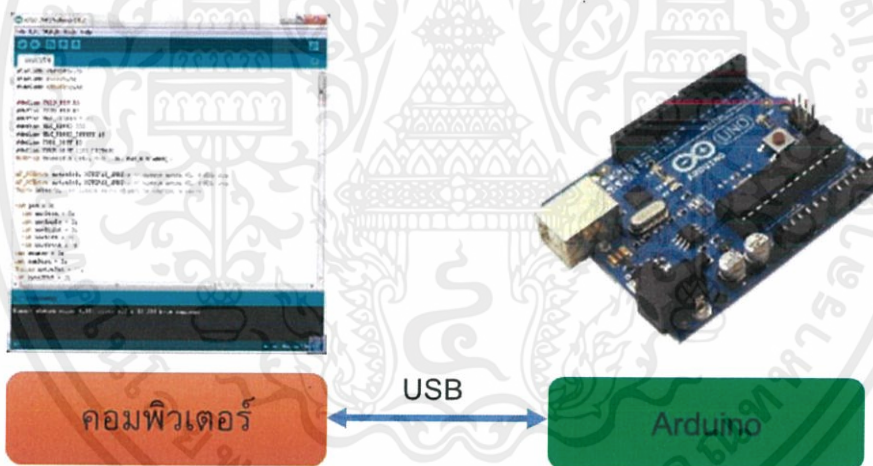
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่เลือกใช้ ข้อดีคือ มี พอร์ต I/O มากกว่า รุ่น Arduino Uno R3 สะดวกกับงานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว อีกทั้งมีความหน่วยความจำแบบ Flash มากกว่ารุ่น Arduino Uno R3 จึงเขียนโค้ดโปรแกรมได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน



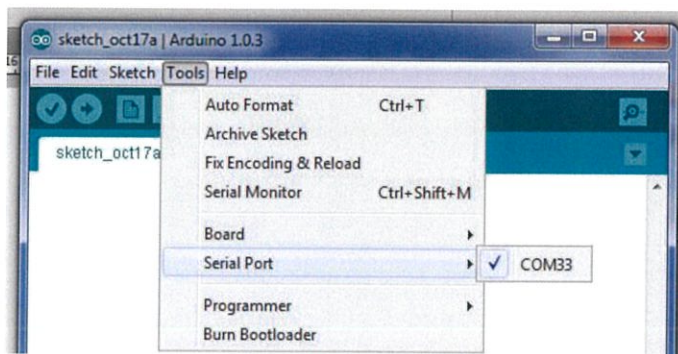
รูปที่ 2.11 บอร์ด Arduino Mega 2560 R3

2.2.2 การเขียนโปรแกรมบน Arduino



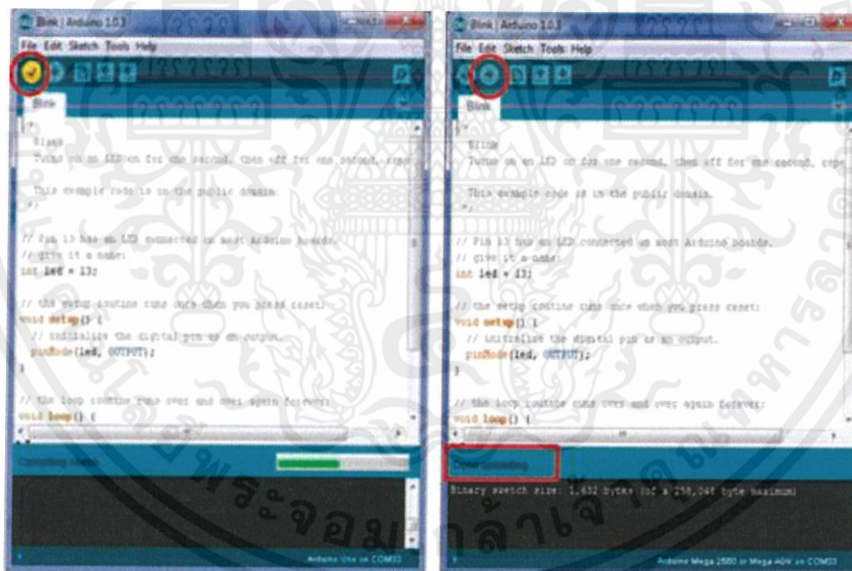
รูปที่ 2.12 แสดงตัวอย่างการใช้งาน Arduino

เมื่อเราเขียนโค้ดแล้วเสร็จบนโปรแกรม ArduinoIDE สิ่งที่ต้องทำต่อไปคือการเลือกรุ่นบอร์ด Arduino ที่เราจะใช้และหมายเลข Com port ที่เราใช้เชื่อมต่อที่เมนู Tools



รูปที่ 2.13 แสดงตัวอย่างการเลือกหมายเลข Comport ของบอร์ด

จากนั้น กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โค้ดโปรแกรม จากนั้นจึงกดปุ่ม Upload เพื่อส่งโค้ดโปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB และเมื่ออัปโหลดเรียบร้อยแล้ว ทางตัวโปรแกรมจะแสดงข้อความว่า “Done uploading” และบอร์ดก็จะสามารถเริ่มทำงานตามที่เราต้องการได้ทันที

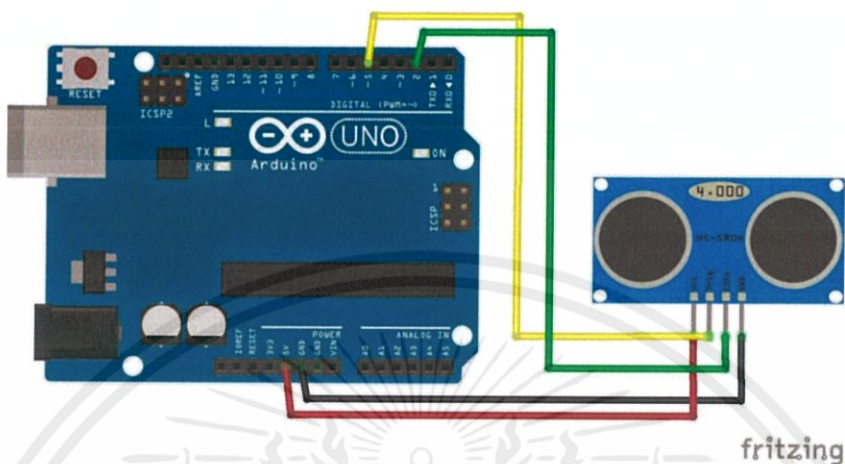


รูปที่ 2.14 แสดงตัวอย่างการอัปโหลดโค้ดและการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

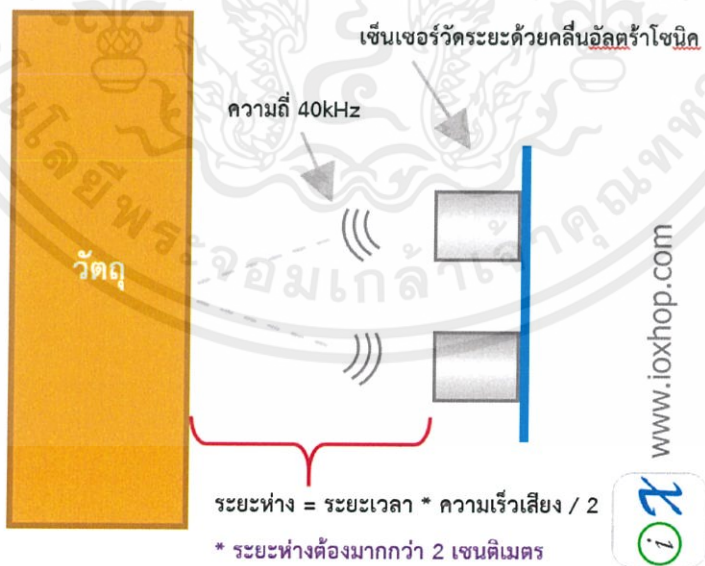
2.3 Ultrasonic Sensor

เป็นการใช้โมดูล Ultrasonic มาใช้งานเป็นเซนเซอร์ที่ควบคุมด้วย Arduino Uno



รูปที่ 2.15 ตัวอย่างการเชื่อมต่อของโมดูล ultrasonic Sensor และ Arduino Board

เริ่มด้วยเราจะกำหนดให้ตัวเรดาร์มีการสแกนทั้งจากขาไปและขากลับ สามารถคำนวณระยะทางจาก ultrasonic ได้โดยการดูว่า คลื่นเสียงที่ถูกส่งออกจากขา Trig ไปกระทบกับวัตถุ และกลับมายังขา echo นั้น เป็นระยะเวลาเท่าไร จากนั้น เราจะนำระยะทางนี้ไปหาระยะทาง



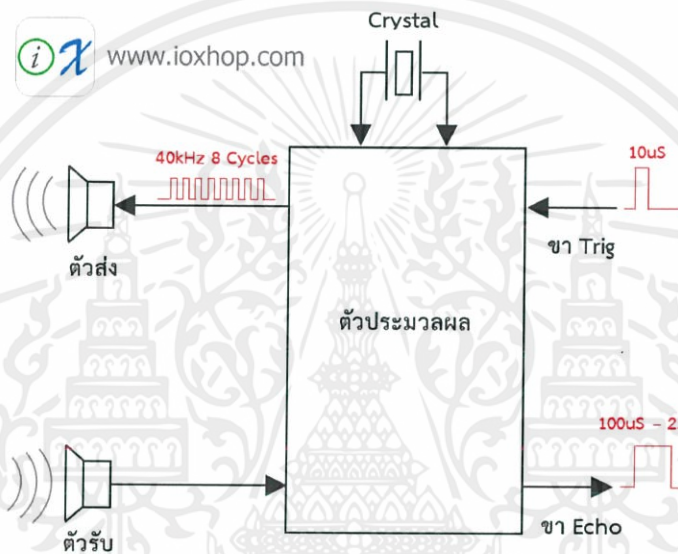
รูปที่ 2.16 แสดงการวัดระยะทางและสูตรการหาระยะทางระหว่างตัวโมดูลและวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการดังกล่าว จะมีตัวแปรหนึ่งที่เปลี่ยนไปตามอุณหภูมิ ซึ่งก็คือ อัตราเร็วของเสียง เราจึงสามารถหาอัตราเร็วของเสียงในอากาศเพื่อนำไปใช้ในสมการการหาระยะทางได้ดังนี้

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

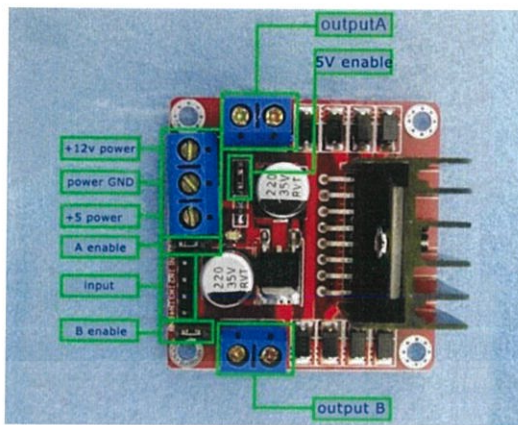
รูปที่ 2.17 แสดงสมการการหาอัตราเร็วของเสียงในอากาศ



รูปที่ 2.18 โค้ดและการทำงานของเซนเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

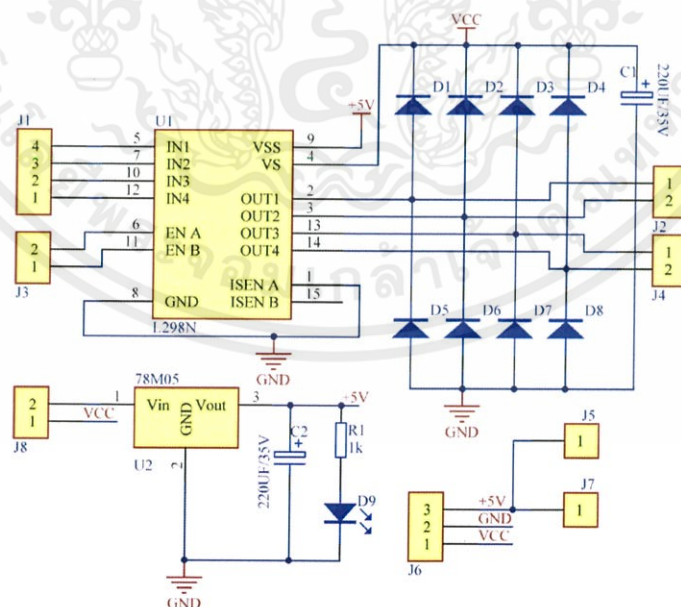
2.4 Motor Drive Module L298N

L298N เป็นชุดขับมอเตอร์ชนิด H-Bridge ซึ่งจำเป็นต่อการขับเคลื่อนชิ้นงานที่เป็นแบบหุ่นยนต์ ส่วนใหญ่จะถูกนำไปใช้ในการควบคุมทิศทางและความเร็วของมอเตอร์ ซึ่งสามารถควบคุมมอเตอร์ได้ทั้งหมด 2 Channel แบบแยกอิสระ จึงง่ายต่อการควบคุมล้อเมื่อต้องการเลี้ยวซ้าย-ขวา โมดูลนี้ใช้แรงดัน 5 โวลต์ สามารถรับแรงดันอินพุตได้ 7-35 โวลต์โดยมีวงจรถูกเรกูเลตในตัว ขับกระแสสูงสุด ช้างละ 2A



รูปที่ 2.19 แสดงรายละเอียดของตัวโมดูล L298N

- ENA: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ A
- IN1: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A
- IN2: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A
- IN3: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B
- IN4: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B
- ENB: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ B

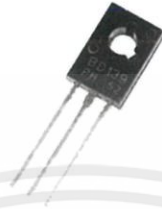


รูปที่ 2.20 แสดงวงจร H-Bridge ภายในตัวโมดูล L298N

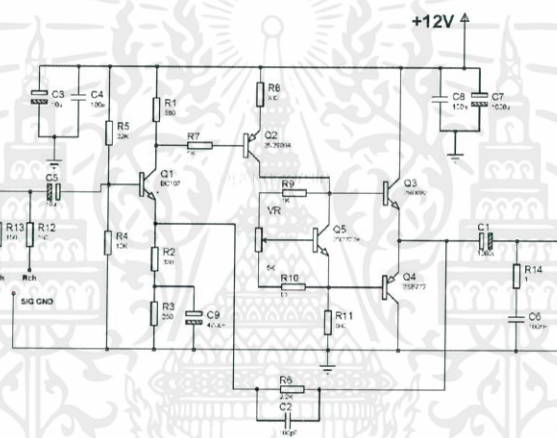
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Transistor BD139

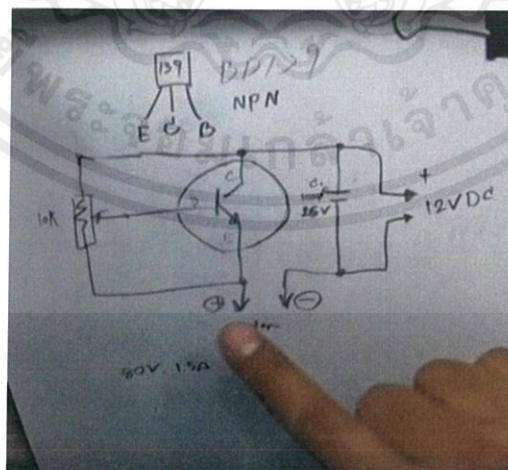
เป็น transistor ที่ใช้สำหรับควบคุมแรงดันให้คงที่และเพื่อไม่ให้มีกระแสมากเกินไป สามารถปรับค่าแรงดันเอาต์พุตได้ในเรนจ์ที่ค่อนข้างกว้าง โดยรับแรงดันอินพุตได้สูงสุดที่ 80V ทนกระแสได้สูงสุด 1.5A และมีความผิดพลาดน้อย



รูปที่ 2.21 แสดงลักษณะ BD139



รูปที่ 2.22 แสดงลักษณะวงจรภายในของ BD139



รูปที่ 2.23 แสดงการต่อวงจรเพื่อใช้งาน BD139

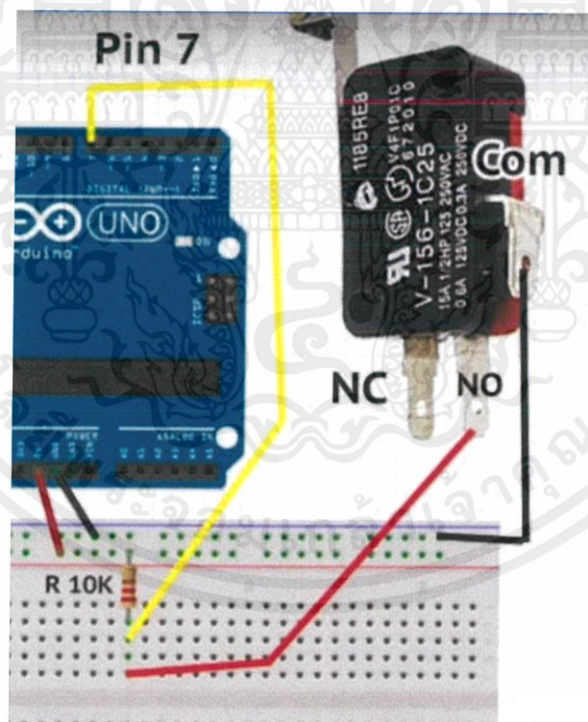
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Limit Switch

เป็นสวิตช์ที่จำกัดระยะทาง ทำงานโดยอาศัยแรงกดภายนอกมากระทำ และเป็นผลทำให้หน้าสัมผัสที่อยู่กับก้านชน เปิด-ปิด ตามจังหวะของการชน ซึ่งเปิด-ปิดสวิตช์นี้เป็นสัญญาณทางดิจิทัล คือ High-Low, On-Off หรือ 0-1 โดยแบ่งตามแรงดันไฟ ซึ่ง เมื่อทำงานร่วมกับ Arduino จะแบ่งโดยใช้ High (5V) กับ Low (0V)



รูปที่ 2.24 Limit Switch ประเภทหนึ่ง เรียกว่า Micro Switch



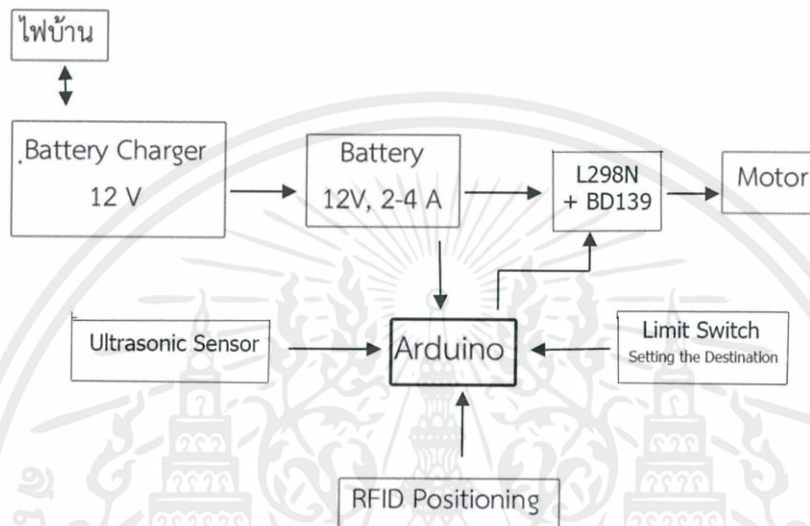
รูปที่ 2.25 ตัวอย่างการต่อ Micro Switch กับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ

3.1 ขั้นตอนการดำเนินงาน



รูปที่ 3.1 Block Diagram แสดง ขั้นตอนการทำงานของหุ่นยนต์

3.2 ออกแบบการทดลอง

3.2.1 ทดสอบตัวมอเตอร์ขับเคลื่อนล้อ โดยการป้อนไฟแบบ $+/+$, $+/-$, $-/+$ และ $-/-$ ทั้งจากแบตเตอรี่โดยตรงและคำสั่งจาก L298N

3.2.2 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board เมื่อนำทุกอุปกรณ์มาเชื่อมต่อเข้าด้วยกัน

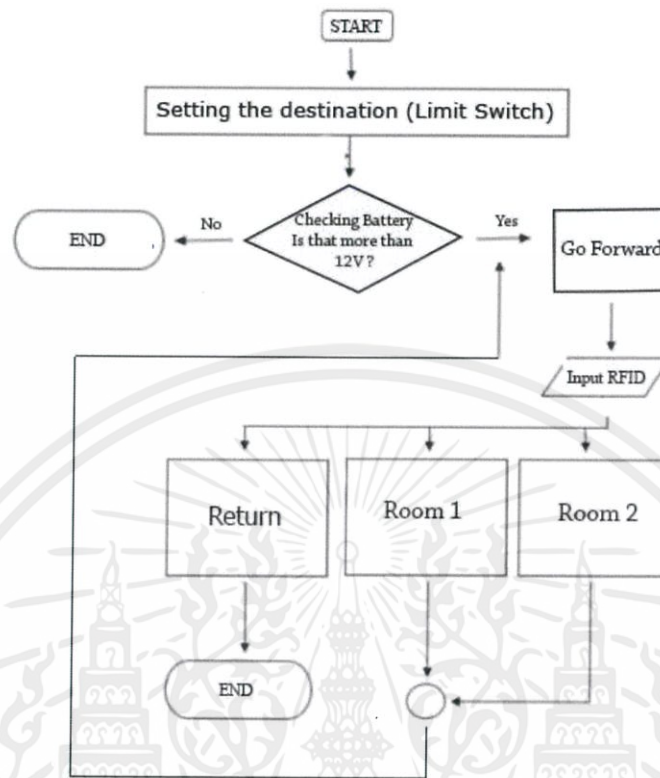
3.2.3 นำหุ่นไปทดสอบจริงตามที่กำหนดไว้ โดยแยกเป็น

3.2.3.1 ทดสอบการเคลื่อนที่ของหุ่นยนต์ในการเคลื่อนที่ไปตามทางตรง โดยทดสอบจากหน้าห้องธุรการภาควิชาวิศวกรรมอิเล็กทรอนิกส์ไปจนถึงห้องสุดทางเดิน

3.2.3.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์ เมื่อตัวเซนเซอร์สามารถตรวจจับสิ่งกีดขวางได้

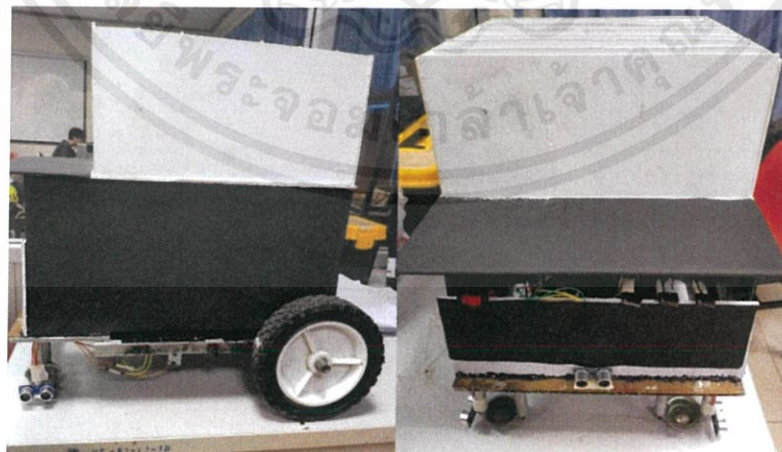
3.2.3.3 ทดสอบการระบุจุดหมายของหุ่นยนต์ จาก ID12 และ RFID Tag Card

โดยตัวหุ่นยนต์มีลำดับการทำงานดังนี้



รูปที่ 3.2 แสดงลำดับการทำงาน (Flow Chart) ของตัวหุ่นยนต์

และตัวหุ่นยนต์ที่ประกอบแล้วเสร็จสมบูรณ์มีรูปลักษณะดังภาพ



รูปที่ 3.3 รูปสำเร็จของ Messenger Robot เมื่อตกแต่งเสร็จแล้ว

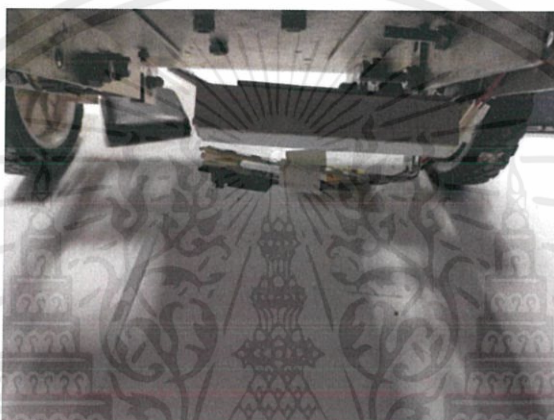
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดสอบตัวมอเตอร์ขับเคลื่อนล้อ โดยการป้อนไฟแบบ +/+, +/-, -/+ และ -/-

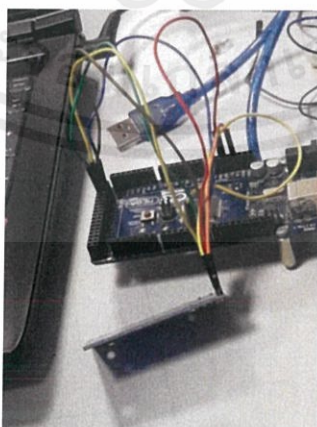
ตัวมอเตอร์ทั้งสองสามารถขับเคลื่อนล้อให้หมุนได้ตาม input ที่กำหนด ทั้งจากแบตเตอรี่โดยตรง และ จากตัว L298N โดยการยกท้าย (ยังไม่ต้องให้หุ่นยนต์เคลื่อนที่) ผลที่ได้คือ ล้อเคลื่อนที่ไปตามทิศทางที่ถูกกำหนดได้โดยไม่มีข้อผิดพลาด ทั้งข้างหน้า, ด้านหลัง, เลี้ยวซ้าย และเลี้ยวขวา



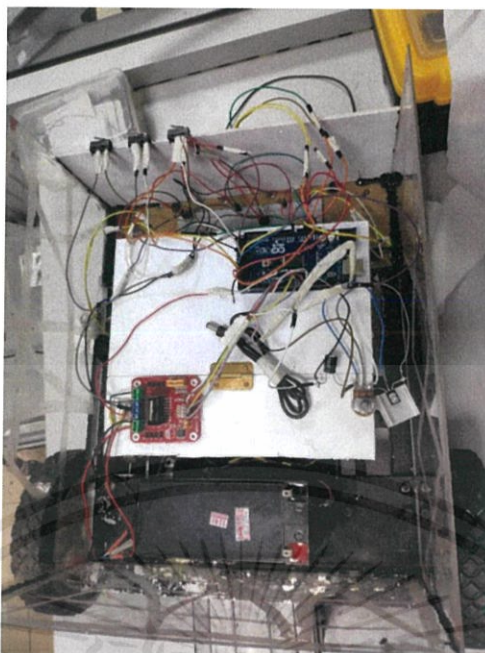
รูปที่ 4.1 มอเตอร์ขับเคลื่อนล้อ

4.2 นำอุปกรณ์มาต่อและการทำงานร่วมกันหลังจากเขียนโปรแกรม Arduino แล้วเสร็จ

4.2.1 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board เมื่อนำทุกอุปกรณ์มาเชื่อมต่อเข้าด้วยกัน พบว่าทุกอุปกรณ์สามารถทำงานด้วยกันได้ตามหน้าที่เมื่อนำมาเชื่อมต่อกันแล้ว



รูปที่ 4.2 ทดสอบการเชื่อมต่อและการทำงานของ Arduino Board



รูปที่ 4.3 ตัวหุ่นยนต์เมื่อเชื่อมต่อทุกอุปกรณ์เข้าด้วยกันแล้วเสร็จ

4.3 นำหุ่นไปทดสอบจริงตามที่กำหนดไว้ โดยแยกเป็น

4.3.1 ทดสอบการเคลื่อนที่ของหุ่นยนต์ในการเคลื่อนที่ไปตามทางตรง โดยทดสอบจากหน้าห้อง
 ธุรการภาควิชาวิศวกรรมอิเล็กทรอนิกส์ไปจนถึงห้องสุดทางเดิน ผลการทดสอบคือ ถึงแม้ว่าหุ่นยนต์จะมีการ
 เบี่ยงไปด้านข้างบ้าง แต่ก็สามารถเคลื่อนที่ไปตามทางตรงได้ตามกำหนด



รูปที่ 4.4 พื้นที่การทดสอบการเคลื่อนที่ทางตรงของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงการทดสอบการเคลื่อนที่ของหุ่นยนต์

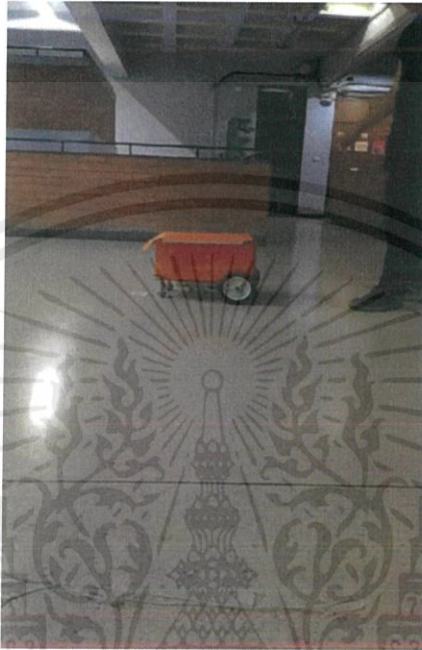
4.3.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์ เมื่อตัวเซนเซอร์สามารถตรวจจับสิ่งกีดขวางได้ พบว่า เมื่อเซนเซอร์ที่อยู่ด้านหน้า และด้านข้างทั้งสองของหุ่นยนต์ตรวจพบสิ่งกีดขวาง หุ่นยนต์จะเคลื่อนที่เบนไปทางฝั่งที่เซนเซอร์ตรวจไม่พบสิ่งกีดขวาง และถ้าสิ่งกีดขวางอยู่ที่ด้านหน้า ตัวหุ่นยนต์จะถอยหลังเล็กน้อย และเบี่ยงเลี้ยวไปที่นิตไปยังด้านที่เซนเซอร์ตรวจไม่พบสิ่งกีดขวาง และจะกลับไปเดินทางเป็นเส้นตรงอีกครั้งเมื่อเซนเซอร์ทั้งสามด้านตรวจไม่พบสิ่งกีดขวางเลย



รูปที่ 4.6 แสดงการทดสอบการตอบสนองเซนเซอร์ของหุ่นยนต์กับสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ทดสอบการระบุจุดหมายของหุ่นยนต์จาก ID12 และ RFID Tag Card พบว่าตัวหุ่นยนต์จะหยุดด้วยตัวมันเอง เมื่อ ID12 ที่อยู่ด้านล่างของตัวหุ่นยนต์ตรวจพบ RFID Tag Card ของจุดหมายปลายทางที่ Match กับ Input จาก Limit Switch ที่เราได้ระบุว่าการให้จุดหมายปลายทางเป็นห้องไหน (เมื่อ Limit Switch ตัวที่ 1 หมายถึงห้องที่ 1 และ Limit Switch ตัวที่ 2 หมายถึงห้องที่ 2)



รูปที่ 4.7 แสดงการทดสอบการระบุจุดหมายจาก Limit Switch กับ RFID Tag card ของจุดหมายปลายทาง



รูปที่ 4.8 ทดสอบการหยุดของหุ่นยนต์เมื่อตรวจพบ RFID Tag Card ของจุดหมายปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

สรุปผลการทดลอง

จากการทดลองที่ผ่านมา ผลลัพธ์ของการทดลองในแต่ละส่วนนั้น ไม่มีความผิดพลาดเกิดขึ้น แต่ละอุปกรณ์ทำงานตามคำสั่งที่ได้จากการเขียนโค้ดโปรแกรม Arduino และเมื่อนำมาประกอบรวมกันแล้ว หุ่นยนต์สามารถเคลื่อนที่จากจุดเริ่มต้น ซึ่งก็คือจากห้องธุรการภาควิชาวิศวกรรมอิเล็กทรอนิกส์ เป็นทางตรง ไปยังห้องโพรเจกต์ที่อยู่สุดทางเดิน สามารถหลบหลีกสิ่งกีดขวางจากเซนเซอร์ทั้งสามตัวที่ถูกติดตั้งอยู่ด้านหน้า และทั้งสองข้างของตัวหุ่นได้ รวมถึงสามารถหยุดได้ด้วยตัวมันเองเมื่อ ID12 ที่อยู่ด้านล่างของหุ่นยนต์ได้ตรวจพบ RFID Tag Card ของจุดหมายปลายทางว่าปลายทางคือห้องไหน โดย RFID Tag Card ที่ถูกตรวจพบนั้น สอดคล้องกับ Limit Switch ซึ่งถูกตั้งให้เป็น input ของตัวหุ่นยนต์นี้

ข้อเสนอแนะ

สิ่งที่เกิดขึ้นคือ ระยะทางของ Ultrasonic sensor นั้นค่อนข้างสั้น เมื่อมีสิ่งกีดขวาง จึงต้องเป็นระยะที่อยู่ค่อนข้างใกล้จึงจะสามารถจับได้ อีกทั้งตัวหุ่นยนต์นี้ยังไม่สามารถใช้ในโรงงานใหญ่ที่มีพื้นที่กว้างไกลได้ เพราะระยะจาก ID12 กับ RFID Tag Card ของจุดหมายปลายทางนั้นต้องอยู่ค่อนข้างใกล้จึงจะสามารถตรวจจับได้ ทางแก้คือใช้อุปกรณ์ที่มีคุณภาพสูงกว่านี้ แต่อุปกรณ์ที่คุณภาพสูงนั้นมีราคาที่สูงกว่าแพง ตัวหุ่นยนต์จึงสามารถใช้งานได้เฉพาะเส้นทางที่ไม่ใช่ที่โล่งกว้างและเคลื่อนที่ได้ในความเร็วที่ค่อนข้างช้าเท่านั้น อีกทั้ง เนื่องจากล้อหน้าทั้งสองของหุ่นยนต์นั้นเป็นลูกกลิ้ง จึงมีการสั่นไถลในบางครั้ง ทำให้การเดินทางเป็นเส้นตรงของหุ่นยนต์ค่อนข้างมีเบี้ยวไปบ้าง แต่เส้นทางการเคลื่อนที่ของหุ่นยนต์ก็ยังถือว่าอยู่ในเส้นทาง

เอกสารอ้างอิง

[1] Arduino DC Motor Control Tutorial

ชื่อเว็บไซต์: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

จาก howtomechatronics [16 ตุลาคม 2561]

[2] Arduino RFID ID-12/20 Library

ชื่อเว็บไซต์: <https://www.youtube.com/watch?v=iSEPRW64EtE&t=147s>

จาก youtube [8 พฤศจิกายน 2561]

[3] Arduino tutorial with code: Innovations ID-12/ID-20 RFID Reader

ชื่อเว็บไซต์: <https://www.youtube.com/watch?v=936vSO6nDHg>

จาก youtube [21 มีนาคม 2562]

[4] Arduino uno ultrasonic radar

ชื่อเว็บไซต์: <https://brain2com.blogspot.com/2018/01/arduino-uno-ultrasonic-radar.html>

จาก brain2com [15 พฤศจิกายน 2561]

[5] Automatic object collector robot

จากหนังสือ ปรินญาณินพนธ์ ปีการศึกษา 2558

โดย กนกพรรณ รัตนสิริเวช, กรวรัญ ดำรงค์ฤทธิ์ และ กฤษติยา ชมพูเงินแสงสกุล [12 มีนาคม 2562]

[6] Document sending robot

จากหนังสือ: ปรินญาณินพนธ์ ปีการศึกษา 2558

โดย ณรงค์เกียรติ นิระเคนม, ณัฐกุล แต่งสกุล และ ณัฐวุฒิ ชนะทวี [12 มีนาคม 2562]

[7] Motor Drive Module L298N

ชื่อเว็บไซต์: <https://www.arduinoall.com/product/125/motor-drive-module-l298n>

จาก Arduinoall [15 พฤศจิกายน 2561]

[8] RFID in Robot-Assisted Indoor Navigation for the Visually Impaired

ชื่อเว็บไซต์: <https://ieeexplore.ieee.org/document/1389688/>

จาก ieeexplore [27 ตุลาคม 2561]

[9] RFID Card Reader/Detector Module Kit

ชื่อเว็บไซต์: <https://www.arduinoall.com/product/102/rfid-card-reader-detector-module-kit-rc522/>

จาก Arduinoall [13 พฤศจิกายน 2561]

[10] RFID Knowledge Communication Reader Tag

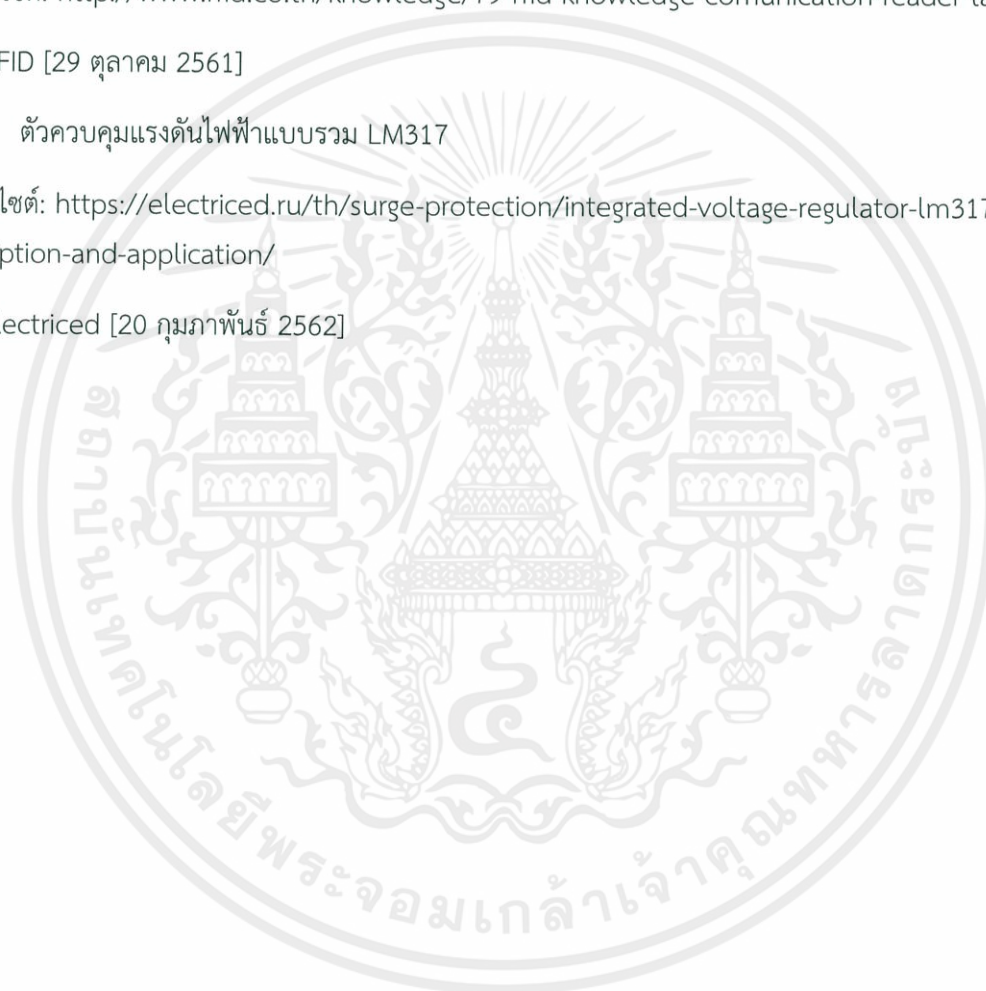
ชื่อเว็บไซต์: <http://www.rfid.co.th/knowledge/79-rfid-knowledge-comunication-reader-tag>

จาก RFID [29 ตุลาคม 2561]

[11] ตัวควบคุมแรงดันไฟฟ้าแบบรวม LM317

ชื่อเว็บไซต์: <https://electriced.ru/th/surge-protection/integrated-voltage-regulator-lm317-description-and-application/>

จาก electriced [20 กุมภาพันธ์ 2562]





เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Code

```
#include <SoftwareSerial.h>

SoftwareSerial rfidSerial(51,50); //(RX, TX)

//Register your RFID tags here

char tag1[13] = "2100332AA098";

char tag2[13] = "21003355CE89";

char tag3[13] = "1E0072085C38";

char tag4[13] = "1E0072051D74";

char tag5[13] = "1E00716D3C3E";

char tag6[13] = "1E0071A830F7";

char tag7[13] = "1E0071F5841E";

char tag8[13] = "1E007186D831";

char tag9[13] = "1E007175EFF5";

char tag10[13] = "1E00717B6F7B";

char tag11[13] = "1E007189A345";

char tag12[13] = "1E0070FE0D9D";

char tag13[13] = "1E0071430F23";

char tag14[13] = "1E0072352970";

char tag15[13] = "1E0071285512";

char tag16[13] = "1E0072265C16";

char tag17[13] = "1E00719C7D8E";

char tag18[13] = "1E007182BF52";
```

```
char tag19[13] = "1E0071888760";
```

```
char tag20[13] = "1E007131E6B8";
```

```
char tagString[13];
```

```
int i=0;
```

```
int j=0;
```

```
int k=0;
```

```
int l=0;
```

```
int z=0;
```

```
int val=0;
```

```
int index = 0;
```

```
boolean reading = false;
```

```
int enableA = 3;
```

```
int pinA1 = 6;
```

```
int pinA2 = 7;
```

```
int enableB = 5;
```

```
int pinB1 = 8;
```

```
int pinB2 = 9;
```

```
//-----ultrasonics-----
```

```
const int trigPin = 24;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
const int echoPin = 26;
```

```
const int trigPin1 = 28;
```

```
const int echoPin1 = 30;
```

```
const int trigPin2 = 32;
```

```
const int echoPin2 = 34;
```

```
long duration;
```

```
int distance;
```

```
long duration1;
```

```
int distance1;
```

```
long duration2;
```

```
int distance2;
```

```
//-----Setup-----
```

```
void setup() {
```

```
    pinMode( 44 , INPUT );
```

```
    pinMode( 42 , INPUT );
```

```
    pinMode( 40 , INPUT );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(trigPin, OUTPUT);  
pinMode(echoPin, INPUT);  
pinMode(trigPin1, OUTPUT);  
pinMode(echoPin1, INPUT);  
pinMode(trigPin2, OUTPUT);  
pinMode(echoPin2, INPUT);
```

```
pinMode(enableA, OUTPUT);  
pinMode(pinA1, OUTPUT);  
pinMode(pinA2, OUTPUT);
```

```
pinMode(enableB, OUTPUT);  
pinMode(pinB1, OUTPUT);  
pinMode(pinB2, OUTPUT);
```

```
Serial.begin(9600);  
rfidSerial.begin(9600);
```

```
}
```

```

//-----//loop/////-----
void loop() {
  if ( digitalRead(44) == LOW && z == 0){
    i=1; //SET UP FOR ROOM1
  }
  if ( digitalRead(42) == LOW && z == 0){
    j=1; //SET UP FOR ROOM 2
  }
  if ( digitalRead(40) == LOW ){
    val = analogRead(A0);
    enableMotors();
    k=0;
    z=1; //START WORKING
  }
  /////-----forward-----
  if (z == 1 && k == 0 && val > 900){
    forward(10);
    distance = calculateDistance();
    distance1 = calculateDistance1();
    distance2 = calculateDistance2();
    if(distance <= 5){
      backward(1000);
    }
  }
}

```

```

    }

else if(distance1 <= 30){

    turnRight(200);

    while(rfidSerial.available()){

int readByte = rfidSerial.read(); //read next available byte

delay(1);

if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){

//store the tag

tagString[index] = readByte;

index ++;

}

if((readByte == 2) && (reading == false)) reading = true; //begin of tag

if((readByte == 3) && (reading == true)) reading = false; //end of tag

}

if(reading == false){

checkTag(tagString);

clearTag(tagString);

index = 0;

}

}

else if(distance2 <= 30){

    turnLeft(200);

    while(rfidSerial.available()){

```

```

int readByte = rfidSerial.read(); //read next available byte

delay(1);

if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){

//store the tag

tagString[index] = readByte;

index ++;

}

if((readByte == 2) && (reading == false)) reading = true; //begin of tag
if((readByte == 3) && (reading == true)) reading = false; //end of tag
}

if(reading == false){
checkTag(tagString);
clearTag(tagString);
index = 0;
}

}

else{
while(rfidSerial.available()){
int readByte = rfidSerial.read(); //read next available byte

delay(1);

if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){

//store the tag

tagString[index] = readByte;

```

```

index ++;
}

if((readByte == 2) && (reading == false)) reading = true; //begin of tag
if((readByte == 3) && (reading == true)) reading = false; //end of tag
}

if(reading == false){
checkTag(tagString);
clearTag(tagString);
index = 0;
}
}
}
}

//-----////checktag1/////-----

void checkTag(char tag[]){
if(strlen(tag) == 0) return; //empty, no need to continue
if(i == 1 && compareTag(tag, tag1)){ // if matched tag1, do this
while(i == 1){
disableMotors();

if ( digitalRead(44) == LOW ){

i=0; //arriving ROOM1

enableMotors();

```

```

}
}
}
if(i == 1 && compareTag(tag, tag2)){ // if matched tag1, do this
while(i == 1){
  disableMotors();
  if ( digitalRead(44) == LOW ){
    i=0; //arriving ROOM1
    enableMotors();
  }
}
}
if(i == 1 && compareTag(tag, tag3)){ // if matched tag1, do this
while(i == 1){
  disableMotors();
  if ( digitalRead(44) == LOW ){
    i=0; //arriving ROOM1
    enableMotors();
  }
}
}
if(i == 1 && compareTag(tag, tag4)){ // if matched tag1, do this
while(i == 1){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

disableMotors();

if ( digitalRead(44) == LOW ){

i=0; //arriving ROOM1

enableMotors();

}

}

}

if(i == 1 && compareTag(tag, tag5)){ // if matched tag1, do this
while(i == 1){
disableMotors();
if ( digitalRead(44) == LOW ){
i=0;
enableMotors();
}
}
}

if(j == 1 && compareTag(tag, tag6)){ // if matched tag1, do this
while(j == 1){
disableMotors();
if ( digitalRead(42) == LOW ){
j=0;
enableMotors();
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
if(j == 1 && compareTag(tag, tag7)){ // if matched tag1, do this
while(j == 1){
disableMotors();
if ( digitalRead(42) == LOW ){
j=0;
enableMotors();
}
}
}
if(j == 1 && compareTag(tag, tag8)){ // if matched tag1, do this
while(j == 1){
disableMotors();
if ( digitalRead(42) == LOW ){
j=0;
enableMotors();
}
}
}
if(compareTag(tag, tag9)){
while(j == 1){
disableMotors();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( digitalRead(42) == LOW ){
j=0;
enableMotors();
}
}
}
else if(compareTag(tag, tag10)){
while(j == 1){
disableMotors();
if ( digitalRead(42) == LOW ){
j=0;
enableMotors();
}
}
}
else if(compareTag(tag, tag11)){
turnRight(3000);
}
else if(compareTag(tag, tag12)){
turnRight(3000);

}
else if(compareTag(tag, tag13)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

turnRight(3000);

}

else if(compareTag(tag, tag14)){
turnRight(3000);

}

else if(compareTag(tag, tag15)){
turnRight(3000);

}

if(compareTag(tag, tag16)){
k=0; //preparing for new order
z=0; //clear z value stop loop
disableMotors();
}

else if(compareTag(tag, tag17)){
k=0; //preparing for new order
z=0; //clear z value stop loop
disableMotors();
}

else if(compareTag(tag, tag18)){
k=0; //preparing for new order

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

z=0; //clear z value stop loop
disableMotors();
}
else if(compareTag(tag, tag19)){
k=0; //preparing for new order
z=0; //clear z value stop loop
disableMotors();
}
else if(compareTag(tag, tag20)){
k=0; //preparing for new order
z=0; //clear z value stop loop
disableMotors();
}
} //End checkTag()

//-----//clear tag//-----
void clearTag(char one[]){ //clear the char array by filling with null - ASCII 0
//Will think same tag has been read otherwise
for(int n = 0; n < strlen(one); n++){
one[n] = 0;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----//compar//////////-----
boolean compareTag(char one[], char two[]) {
//compare two value to see if same, strcmp not working 100% so we do this
if(strlen(one) == 0) return false; //empty
for(int n = 0; n < 12; n++){
if(one[n] != two[n]) return false;
}
return true; //no mismatches
}

//-----//distance//////-----//
int calculateDistance(){
while(rfidSerial.available()){
int readByte = rfidSerial.read(); //read next available byte
delay(1);
if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){
//store the tag
tagString[index] = readByte;
index ++;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((readByte == 2) && (reading == false)) reading = true; //begin of tag

    if((readByte == 3) && (reading == true)) reading = false; //end of tag

    }

    if(reading == false){

    checkTag(tagString);

    clearTag(tagString);

    index = 0;

    }

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance= duration*0.034/2;
    return distance;
}

//-----distance1-----

int calculateDistance1(){
    while(rfidSerial.available()){

        int readByte = rfidSerial.read(); //read next available byte

```

```

delay(1);

if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){

//store the tag

tagString[index] = readByte;

index ++;

}

if((readByte == 2) && (reading == false)) reading = true; //begin of tag
if((readByte == 3) && (reading == true)) reading = false; //end of tag
}

if(reading == false){
checkTag(tagString);
clearTag(tagString);
index = 0;
}

digitalWrite(trigPin1, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

duration1 = pulseIn(echoPin1, HIGH);

distance1= duration1*0.034/2;

return distance1;

```

```

}

//-----//distance1//-----

int calculateDistance2(){
    while(rfidSerial.available()){
        int readByte = rfidSerial.read(); //read next available byte

        delay(1);

        if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){
            //store the tag
            tagString[index] = readByte;
            index ++;
        }
        if((readByte == 2) && (reading == false)) reading = true; //begin of tag
        if((readByte == 3) && (reading == true)) reading = false; //end of tag
        }
        if(reading == false){
            checkTag(tagString);
            clearTag(tagString);
            index = 0;
        }

        digitalWrite(trigPin2, LOW);

        delayMicroseconds(2);

        // Sets the trigPin on HIGH state for 10 micro seconds
        digitalWrite(trigPin2, HIGH);

```

```

delayMicroseconds(10);

digitalWrite(trigPin2, LOW);

duration2 = pulseIn(echoPin2, HIGH);

distance2= duration2*0.034/2;

return distance2;

}

```

```

//-----//-----//motor//////////-----

```

```

void enableMotors()
{
motorAOn();
motorBOn();
}

```

```

void disableMotors()
{
motorAOff();
motorBOff();
}

```

```

void forward(int time)
{
while(rfidSerial.available()){

```

```

int readByte = rfidSerial.read(); //read next available byte

delay(1);

if((reading == true)&&(readByte != 3)&&(readByte != 10)&&(readByte != 13)){

//store the tag

tagString[index] = readByte;

index ++;

}

if((readByte == 2) && (reading == false)) reading = true; //begin of tag
if((readByte == 3) && (reading == true)) reading = false; //end of tag
}

if(reading == false){
checkTag(tagString);
clearTag(tagString);
index = 0;
}

motorAForward();
motorBForward();
delay(time);
}

void backward(int time)
{
motorABackward();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
motorBBackward();  
delay(time);  
}
```

```
void turnLeft(int time)  
{  
motorABackward();  
motorBForward();  
delay(time);  
}
```

```
void turnRight(int time)  
{  
motorAForward();  
motorBBackward();  
delay(time);  
}
```

```
void coast(int time)  
{  
motorACoast();  
motorBCoast();  
delay(time);  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
void brake(int time)
```

```
{
```

```
motorABrake();
```

```
motorBBrake();
```

```
delay(time);
```

```
}
```

```
//Define low-level H-bridge commands
```

```
//enable motors
```

```
void motorAOn()
```

```
{
```

```
digitalWrite(enableA, HIGH);
```

```
}
```

```
void motorBOn()
```

```
{
```

```
digitalWrite(enableB, HIGH);
```

```
}
```

```
//disable motors
```

```
void motorAOff()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{  
digitalWrite(enableB, LOW);  
}
```

```
void motorBOff()
```

```
{  
digitalWrite(enableA, LOW);  
}
```

```
//motor A controls
```

```
void motorAForward()
```

```
{  
digitalWrite(pinA1, HIGH);  
digitalWrite(pinA2, LOW);  
}
```

```
void motorABackward()
```

```
{  
digitalWrite(pinA1, LOW);  
digitalWrite(pinA2, HIGH);  
}
```

```
//motor B controls
```

```
void motorBForward()  
{  
digitalWrite(pinB1, HIGH);  
digitalWrite(pinB2, LOW);  
}
```

```
void motorBBackward()  
{  
digitalWrite(pinB1, LOW);  
digitalWrite(pinB2, HIGH);  
}
```

//coasting and braking

```
void motorACoast()  
{  
digitalWrite(pinA1, LOW);  
digitalWrite(pinA2, LOW);  
}
```

```
void motorABrake()  
{  
digitalWrite(pinA1, HIGH);  
digitalWrite(pinA2, HIGH);  
}
```

```
}
```

```
void motorBCoast()
```

```
{
```

```
digitalWrite(pinB1, LOW);
```

```
digitalWrite(pinB2, LOW);
```

```
}
```

```
void motorBBrake()
```

```
{
```

```
digitalWrite(pinB1, HIGH);
```

```
digitalWrite(pinB2, HIGH);
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้