



รายงานวิชา Pre-Project รหัสวิชา 01216747

จัดทำโดย

นางสาวปาลิตา บัวแก้ว รหัสนักศึกษา 60010613

นางสาวพิมพ์ลดา กิตติมนิรมย์ รหัสนักศึกษา 60010722

นางสาวภัสวรินทร์ แก้วหา รหัสนักศึกษา 60010774

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

ภาคเรียนที่ 2 ปีการศึกษา 2562

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Pre-Project รหัสวิชา 01216747 ซึ่งประกอบไปด้วยส่วนของแนวคิดในการทำโปรเจค กลยุทธ์ในเชิงรุกและรับ วงจรไฟฟ้า การออกแบบตัวรถ และ Programming Code ในการควบคุมการทำงาน

โดยโปรเจคนี้ใช้โปรแกรม Arduino ในการเขียนโปรแกรมควบคุมการทำงานของตัวรถ ใช้โปรแกรม Solidwork ในการออกแบบตัวรถ และใช้โปรแกรม EdrawMax ในการจำลองวงจรไฟฟ้า โดยทางคณะผู้จัดทำขอขอบคุณ ผศ.ดร.อุดม จันทร์จรัสสุข ในการให้คำปรึกษาและช่วยเหลือในการทำโปรเจคและรายงานฉบับนี้ให้สำเร็จไปได้ด้วยดี

คณะผู้จัดทำหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักศึกษาที่กำลังหาข้อมูลเกี่ยวกับเรื่องนี้อยู่หากมีข้อเสนอแนะหรือเกิดข้อผิดพลาดประการใด คณะผู้จัดทำขอน้อมรับไว้และขอภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทนำ	4-5
ทฤษฎีและหลักการที่เกี่ยวข้อง	6-17
Mechanical Design	18
Programming Codes	19-26
เอกสารอ้างอิง	27

บทที่ 1

บทนำ

1. ความเป็นมา

ในโลกยุคปัจจุบันที่เทคโนโลยีมีความก้าวหน้าและพัฒนาไปอย่างรวดเร็ว ทำให้ชีวิตของมนุษย์ต้องพึ่งพาเทคโนโลยีในการใช้ชีวิตประจำวันอยู่เสมอ และในบางครั้งคนเราก็ไม่รู้ตัวเลยว่ากำลังใช้และพึ่งพาเทคโนโลยี ซึ่งอุปกรณ์เครื่องใช้ต่างๆ ในชีวิตประจำวันของเราในปัจจุบันนั้น ล้วนได้รับการพัฒนามาจากความก้าวหน้าของเทคโนโลยีจนนำมาผลิตเป็นสิ่งของเครื่องใช้ เพื่ออำนวยความสะดวกให้กับมนุษย์มากมาย จนเกิดความเคยชิน

นอกจากสิ่งของเครื่องใช้ต่างๆ ที่ใช้กันในชีวิตประจำวันแล้ว การประยุกต์ใช้งานหุ่นยนต์ ระบบอัตโนมัติและปัญญาประดิษฐ์ ก็เข้ามามีความสำคัญมากยิ่งขึ้น ทั้งในด้านอุตสาหกรรม ด้านการแพทย์ ด้านการเกษตร ด้านโลจิสติกส์และอื่นๆ อีกมากมาย ทำให้หุ่นยนต์อุตสาหกรรม โดรน และมือหุ่นยนต์ที่ติดตั้งเซ็นเซอร์มีความต้องการเพิ่มขึ้นจากที่ผ่านมา ซึ่งมีสาเหตุมาจากหลายอย่าง เช่น ค่าแรงที่พุ่งสูงขึ้น และปัญหาการขาดแคลนแรงงาน ทำให้ผู้ประกอบการต้องใช้ระบบอัตโนมัติและระบบหุ่นยนต์มากขึ้นหรือแม้แต่ความผันผวนของอุปสงค์ในตลาดโลก ซึ่งส่งผลให้ผู้ประกอบการต้องหันมาใช้เทคโนโลยีที่ทั้งช่วยลดต้นทุนการผลิตและเพิ่มประสิทธิภาพในการทำงาน แต่การที่หุ่นยนต์ที่จะถูกนำมาใช้งานทดแทนคนไม่ได้หมายความว่ามั่นคงดีกว่าคนทั้งหมดแน่นอนว่าหุ่นยนต์ก็คือเหล็กที่ถูกนำมาประกอบเข้าด้วยกันและใส่กลไกต่างๆ ให้สามารถเคลื่อนที่ได้ อีกทั้งยังต้องมีระบบในการควบคุมการทำงาน ดังนั้น การที่หุ่นยนต์จะทำงานหรือเคลื่อนที่ได้จะต้องอาศัยทักษะและความรู้จากคนอยู่ดี ทั้งนี้ เพื่อทำหน้าที่ในการป้อนโปรแกรมคำสั่งต่างๆ ให้หุ่นยนต์สามารถทำงานได้ตามที่มนุษย์ต้องการ ความต้องการผู้ที่มีทักษะการเขียนโปรแกรมและผู้มีองค์ความรู้ด้านระบบเซนเซอร์เป็นที่ต้องการตัวมากขึ้น เราจึงต้องมีการเตรียมพร้อมเพื่อให้ทันต่อการเปลี่ยนแปลงที่จะเกิดขึ้นด้วย

2. ปัญหาหรือโจทย์ที่ต้องการแก้

ในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้ามจนผ่านเส้นแดง แล้วกลับมามากอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้ ก็จะเป็นฝ่ายชนะในการแข่งขันรอบนั้น หุ่นยนต์ที่ถูกจับได้จะถูกตัดออก

จากการแข่งขันในรอบนั้น ส่วนทีมรับ จะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

3. กลยุทธ์รุก

1. Sensor 1 : ติดหน้าคันรถ จับวัตถุข้างหน้าในระยะ 15 ซม เมื่อพบวัตถุจะหยุด 5 วิแล้วจึงเดินต่อ
2. Sensor 2 : ติดฝั่งขวา เพื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วเลี้ยวรถไปฝั่งซ้าย
3. Sensor 3 : ติดฝั่งซ้าย เพื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วเลี้ยวรถไปฝั่งขวา
4. Sensor : จับเส้นสีแดง เมื่อพบแล้วไปข้างหน้าต่อ 3 วิแล้วจึงหยุด ติดตั้งไว้ด้านหลังของรถ

4. กลยุทธ์รับ

โดยกลยุทธ์ในการรับนั้น จะทำการแบ่งพื้นที่และวางแผนกับทุกคนในทีม และจะเดินรถในแนวนอน เพื่อให้ได้พื้นที่ในการป้องกันมากที่สุด เมื่อเซนเซอร์ด้านข้างจับรถคันอื่นได้ หมายความว่ามียอดฝั่งตรงข้ามวิ่งมาชน ให้รถหยุด เพื่อป้องกันไม่ให้ฝ่ายตรงข้ามข้ามผ่านไปได้ เมื่อเซนเซอร์ด้านหน้าหรือหลังจับได้ ให้เดินหน้าเพื่อพุ่งไปจับและป้องกันรถฝั่งตรงข้ามไว้

1. Sensor 1 : ติดหน้าคันรถ จับวัตถุข้างหน้าในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 5 วิแล้วหยุด
2. Sensor 2 : ติดฝั่งขวา เพื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วเลี้ยวรถไปฝั่งขวา
3. Sensor 3 : ติดฝั่งซ้าย เพื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วเลี้ยวรถไปฝั่งซ้าย

4. วัตถุประสงค์

1. เพื่อจำลองการสร้างหุ่นยนต์และสามารถนำไปต่อยอดในอนาคตได้
2. เพื่อศึกษาการเขียนโปรแกรมและการต่อวงจร ช่วยเพิ่มความรู้และประสบการณ์ในด้านต่างๆ
3. เพิ่มความสามารถในการคิดวิเคราะห์ (Critical thinking) ทักษะในการแก้ไขปัญหาที่ซับซ้อน (Complex Problem Solving) การมองภาพรวมและการเชื่อมโยงสิ่งต่างๆได้
4. เพิ่มทักษะในการทำงานกับผู้อื่น ทักษะในการเป็นผู้นำและทักษะในการทำงานเป็นทีมกับผู้อื่น
5. เพื่อเรียนรู้การใช้งานของระบบเซนเซอร์และอุปกรณ์ชนิดอื่นๆ

5. ขอบเขตของโครงการ

โครงการนี้เป็นการทดลองสร้างวงจรบังคับทิศทางของรถขนาดเล็ก โดยนำอุปกรณ์และโมดูลต่างๆ มาประกอบวงจรบวกกับการเขียนซอฟต์แวร์ ซึ่งจะควบคุมทิศทางการเดินรถด้วยเซนเซอร์รูปแบบต่างๆ พร้อมทั้งการเขียนด้วยโปรแกรม Arduino

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

1. ภาษาของการเขียนโปรแกรมใช้งาน Arduino Board ได้แก่ ภาษา C/C++

ฟังก์ชันหลัก(Structure) เป็นฟังก์ชันหลักในการเขียนโปรแกรม

- Setup() คือ ฟังก์ชันใช้ในการประกาศค่าเริ่มต้น ตำแหน่งพอร์ตที่ใช้ใช้งาน รวมถึงฟังก์ชันที่อยู่ไลบรารีที่ใช้งาน เป็นฟังก์ชันที่ทำงานเพียงครั้งเดียว จะทำงานทุกครั้ง ที่มีการรีเซต หรือรีบูตเครื่องใหม่ เท่านั้น
- Loop () คือ ฟังก์ชันใช้ในการเขียนโค้ดโปรแกรมการทำงานของArduinoเป็นฟังก์ชันการวนลูปไปเรื่อย ๆ

ชุดคำสั่งในการควบคุม (Control Structures) เป็นชุดคำสั่งในการใช้ในการตัดสินใจหาทางออก เพื่อใช้ในการทำงาน

- If คือ คำสั่งในการตัดสินใจ แบบตัวเลือกเดียว โดยใช้งานร่วมกับ and, or not, ==, !=, <, > เพื่อใช้ในการตัดสินใจในการหาคำตอบ
- If...else คือ คำสั่งในการตัดสินใจ แบบหลายตัวเลือก โดยใช้งานร่วมกับ And, Or Not, ==, !=, <, > เพื่อใช้ในการตัดสินใจในการหา
- For เป็นคำสั่งกำหนดเงื่อนไขเป็นจำนวนครั้งที่จะทำตามชุดคำสั่งต่าง ๆ ภายใน loop เหมาะที่จะใช้กับงานประเภทที่ไม่มีการ
- Switch case ใช้ในการจัดการเงื่อนไขหลายเงื่อนไขโดยเฉพาะการใช้งานโครงสร้าง การจำแนกเงื่อนไข ไม่จำเป็นต้องอาศัยเฉพาะตัวแปรที่เก็บค่าจำนวนเต็มเท่านั้น ข้อมูลแบบอื่นก็ใช้ได้
- While คือเงื่อนไขที่จะทำการตรวจสอบว่าเป็นจริงหรือเท็จ ชุดคำสั่งก็คือ ส่วนที่ทำงานซ้ำๆ โดยจะต้องมีคำสั่งที่จะทำให้ เงื่อนไข เป็นเท็จด้วย
- Do... while เป็นคำสั่งที่กำหนดให้มีการทำงานวนรอบ คล้าย ๆ คำสั่ง While
- Break เป็นคำสั่งที่ให้โปรแกรมออกจาก loop ทันที โดยไม่ทำคำสั่งที่เหลือต่อ
- Continue ใช้สำหรับสั่งให้กลับไปเริ่มต้นที่จุดเริ่มต้นใหม่ ใช้ร่วมกับคำสั่งการวนลูปต่างๆจะต่างกับคำสั่ง คำสั่ง break นั้นจะเป็นคำสั่งเพื่อออกจาก loop ส่วนคำสั่ง continue นั้นจะเป็นคำสั่งเพื่อไปยังต้น
- Return คือ คำสั่งที่ส่งค่าอะไรก็ได้กลับออกไปจากฟังก์ชัน
- Goto เป็นคำสั่งที่ทำให้ กระโดดไปทำบรรทัดนั้น
- #define คือ คำสั่งกำหนดค่านิพจน์ต่าง ๆ ให้กับชื่อของตัวคงที่

- #include การกำหนดชื่อไฟล์ตามหลัง include จะใช้เครื่องหมาย <> ซึ่งจะเป็นการอ่านไฟล์จากโฟลเดอร์ที่กำหนดไว้

การแปลงค่า (Conversion)

- Char () แปลงค่าข้อมูลให้เป็น char
- Byte () แปลงค่าข้อมูลให้เป็น byte
- Int () แปลงค่าข้อมูลให้เป็น int
- Word () แปลงค่าข้อมูลให้เป็น word
- Long () แปลงค่าข้อมูลให้เป็น long
- Float () แปลงค่าข้อมูลให้เป็น float

Time

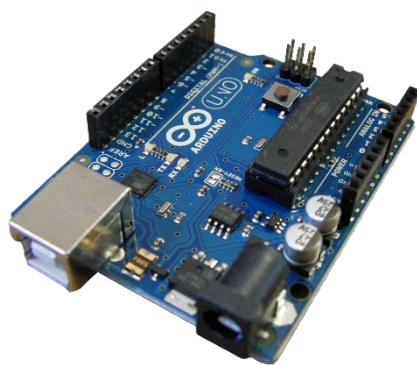
- delay () คือ คำสั่งหยุดการทำงานของโปรแกรมสำหรับจำนวนของเวลา (ใน milliseconds)
Milliseconds = จำนวน มิลลิวินาทีในการหยุดการทำงานชั่วคราว

Functions

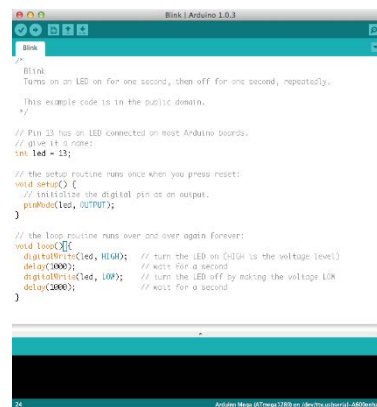
- pinMode () ใช้ในกลุ่ม void setup () เพื่อกำหนดหน้าที่ขาของไมโครคอนโทรลเลอร์ให้เป็นขารับสัญญาณ INPUT หรือขาส่งสัญญาณ OUTPUT
- digitalWrite () คือ การส่งค่าลอจิก HIGH หรือ LOW (เปิด หรือปิด) ไปยังขา digital ที่กำหนดหมายเลขขาไอซีอาจ กำหนดเป็นตัวแปรหรือค่าคงที่ (0-13)
- Digital Read () อ่านค่าจาก ขาไอซีที่ถูกกำหนดให้เป็น digital pin ซึ่งจะได้ผลลัพธ์เป็น HIGH หรือ LOW หมายเลขขาไอซีอาจกำหนดเป็นตัวแปรหรือค่าคงที่ (0-13)

Analog I/O

- Analog Read () คำสั่งนี้อ่านค่าจากขา Analog จะได้ค่า 10-bit คำสั่งนี้จะทำงานกับขา analog input (0-5) เท่านั้น และได้ผลลัพธ์เป็นเลขจำนวนเต็มค่า 0 – 1023
- AnalogWrite() เป็น คำสั่งเขียนค่า analog เทียมโดยใช้ hardware enabled pulse width modulation(PWM) ไปยังขา outputที่สามารถทำ PWM ได้ ใน Arduinoรุ่นใหม่ที่ใช้ชิพ Atmega168 คำสั่งนี้จะทำงานกับขา 3, 5, 6, 9, 10, และ 11 ส่วน Arduinoรุ่นเก่าที่ใช้ Atmega8 จะรองรับเพียงขา 9, 10 และ 11 ค่าที่เขียนสามารถใช้เป็นตัวแปรหรือค่าคงที่จาก 0 – 255
- Delay () หยุดการทำงานของโปรแกรมสำหรับจำนวนของเวลา (ใน milliseconds) ระบุเป็นพารามิเตอร์ (มี 1,000 มิลลิวินาทีในทั้งสองเป็น.) Milliseconds = จำนวนมิลลิวินาทีในการหยุดการทำงานชั่วคราว



รูปที่ 1 Arduino Board

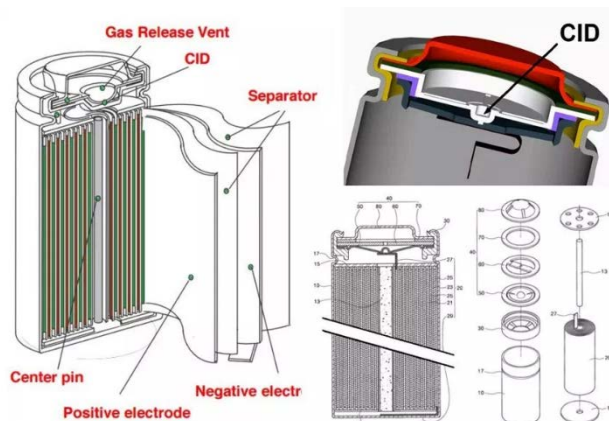


รูปที่ 2 Arduino programming

2. Battery (1865 Li-ion) - 3.7 V 3400 mAh



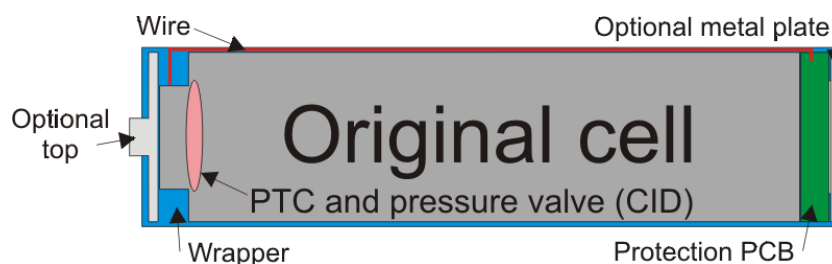
รูปที่ 3 Battery Case



รูปที่ 4 โครงสร้างภายใน 18650 Li-ion Battery

ชื่อแบตเตอรี่ 18650 นั้นมาจากขนาดของตัวมันเอง (เส้นผ่านศูนย์กลาง 18mm x ความยาว 65mm) ซึ่งสามารถจ่ายกระแสได้สูงกว่าแบตเตอรี่ขนาด AA และให้ความจุที่มากกว่า จึงเป็นที่นิยมกันอย่างแพร่หลายในการใช้กับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เช่น ไฟฉายแรงสูง คอมพิวเตอร์ เครื่องดูดฝุ่น โซล่าเซลล์ มอเตอร์ไซค์ไฟฟ้า แบตเตอรี่สำรอง ฯลฯ แต่ด้วยแบตเตอรี่ชนิดลิเธียม 18650 ไม่เหมือนแบตเตอรี่ทั่วไป NiMH หรือแบตเตอรี่แห้ง ดังนั้นจึงมีการออกแบบโครงสร้างภายในให้พิเศษกว่า ในรูป Separators (ตัวกั้น) ก็คือ แผ่นฉนวนพลาสติกบางๆที่ใช้สำหรับแยก ขั้วแอโนด (+) ออกจากแคโทด (-) มันต้องมีรูพรุนเพียงพอเพื่อให้ไอออนเคลื่อนย้ายไปมา ขนาดของรูต้องเล็กพอ เพื่อป้องกันอิเล็กโตไลต์จากการโยกย้ายไปมา [4]

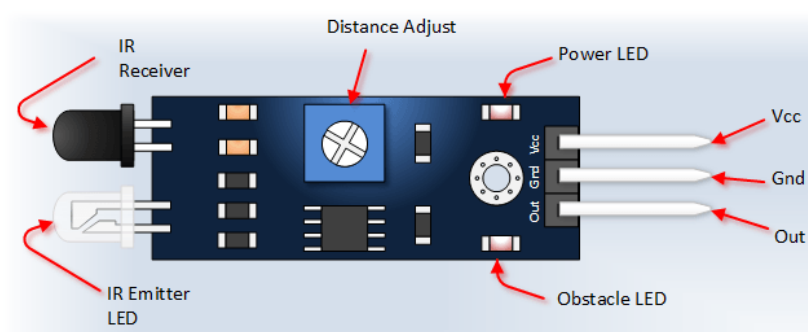
โครงสร้างภายในของแบตเตอรี่ 18650



รูปที่ 5 โครงสร้างภายใน 18650 Li-ion Battery

1. PCT จะเป็นตัวป้องกันความร้อนของ Cell เกินพิกัด และตัดการทำงานของแบตเตอรี่ลง โดยจะสามารถกลับมาใช้งานได้เมื่ออุณหภูมิเข้าสู่สภาวะปกติ
2. CID นั้นจะเป็นวาล์วป้องกันความดันภายใน Cell เกินพิกัดจนอาจนำไปสู่การระเบิดได้ โดยวาล์วตัวนี้จะทำหน้าที่ตัดการทำงานของ Cell ถาวร ไม่สามารถคืนสภาวะกลับมาใช้ใหม่ได้อีก หากสังเกตที่ขั้วของแบตเตอรี่จะพบรูเล็กๆ ที่ถูกออกแบบไว้สำหรับระบายแก๊สหากมีแรงดันผิดปกติภายใน cell นั้นเอง
3. Protected PCB หรือเรียกกันว่า 18650 แบบมีวงจร ซึ่งเป็นวงจรอิเล็กทรอนิกส์ขนาดเล็กที่ฝังไว้ภายในชุดของตัวแบตเตอรี่ ทำหน้าที่คอยป้องกันการใช้กระแสเกิน (Over Current) ป้องกันแรงดันชาร์ตเกิน (Over Charge Voltage) และป้องกันการใช้ไฟในระดับโวลต์ที่ต่ำกว่ากำหนด (Over Discharge) โดยภายในจะมี IC ที่คอย ตรวจสอบอยู่ตลอดเวลาโดย IC ถูกออกแบบให้มีการกินกระแสน้อยมากๆ ในระดับไมโครแอมป์ ซึ่งแทบจะไม่ส่งผลกับปริมาณแบตเตอรี่เลย

3. IR Infrared Obstacle Avoidance Sensor Module

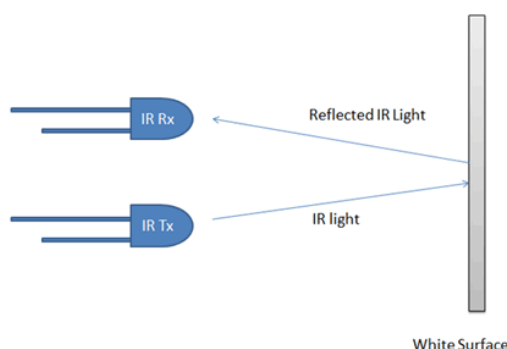


รูปที่ 6 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง

หุ่นยนต์และมนุษย์มีหลักการทำงานที่เหมือนกัน คือ หน่วยรับข้อมูลเข้า (Input Unit) หน่วยประมวลผล (Process Unit) และหน่วยแสดงผล (Output Unit) ดังนั้นการที่หุ่นยนต์จะเคลื่อนที่ไปให้ตรง

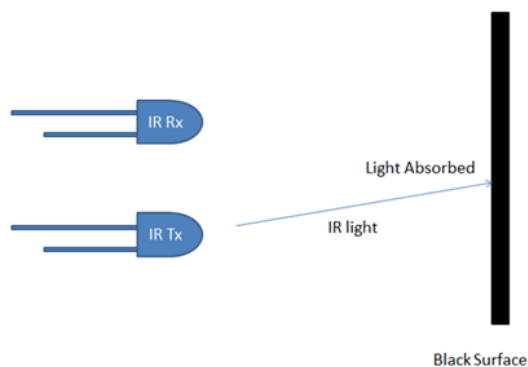
เป้าหมาย หุ่นยนต์จะต้องมีอุปกรณ์ที่จะตรวจสอบตำแหน่งและส่งข้อมูลที่ได้ไปยังหน่วยประมวลผล เพื่อให้มอเตอร์ทำการแสดงผลโดยการไปยังเป้าหมายต่อไป อุปกรณ์พื้นฐานที่ใช้ในการตรวจสอบตำแหน่งนั้น คือ โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง (IR Infrared Obstacle Avoidance Sensor Module) โดยโมดูลนี้ จะมีตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ (สีขาว) infrared จะส่งสัญญาณออกมา และเมื่อมีวัตถุมาบัง คลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้อนกลับไปเข้าตัวรับสัญญาณ (สีดำ) สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้ ภายตัวเซ็นเซอร์แบบนี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่ง เหมือนแบบ Opposed Mode ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสมสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อนและ ทำให้ทำงานผิดพลาดได้

เซ็นเซอร์แบบนี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลา จะแสดงค่า เป็น 0



รูปที่ 7 การทำงานของเซ็นเซอร์

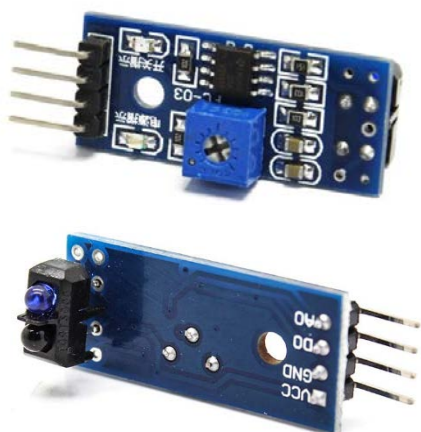
หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับแสงที่จะสะท้อนกลับมาได้ จะแสดงค่า เป็น 1 ซึ่งจะทำให้วงจรภายในรับรู้ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาต์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate



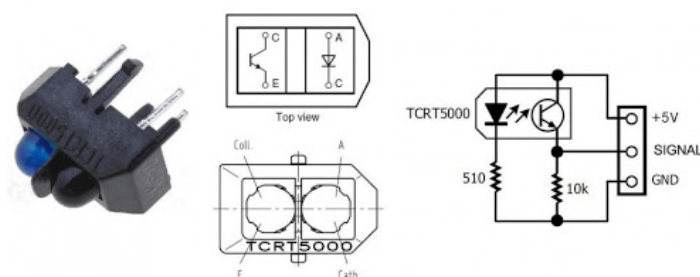
รูปที่ 8 การทำงานของเซนเซอร์เมื่อเจอสิ่งกีดขวาง

4. TCRT5000 Infrared Reflectance Obstacle Avoidance Line Tracking Sensor

เป็นโมดูลตรวจจับวัตถุระยะใกล้ ซึ่งจะอ่านค่าสะท้อนกลับของแสง ใช้ไฟ 3.3 - 5V เหมาะสำหรับใช้กับ Arduino โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณ เป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัวหลอดโฟโตไดโอดที่ทำหน้าที่รับแสงอินฟราเรด โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal ซึ่งสามารถปรับค่าที่ต้องการได้ เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการก็จะส่งค่า 1 ออกมา ถ้ายังไม่ถึงระดับก็จะส่งค่า 0 ออกมา แต่สำหรับบางโมดูลอาจจะรองรับ output แบบ Analog signal ด้วย โดยอ่านค่าได้เป็นตัวเลข 0-1023 หรือสัญญาณไฟในช่วง 0-5 ส่วนตัว R ปรับค่านั้นใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจพบวัตถุของตัวเซนเซอร์



รูปที่ 9 TCRT5000



รูปที่ 10 วงจรของ TCRT5000

5. DC motor

มอเตอร์กระแสตรง (DC Motor) คืออุปกรณ์ทางไฟฟ้าที่แปลงพลังงานไฟฟ้าเป็นพลังงานกล หลักการทำงานของมอเตอร์เกิดจากการทำงานร่วมกันระหว่างสนามแม่เหล็กของแม่เหล็กในตัวมอเตอร์ และสนามแม่เหล็กที่เกิดจากกระแสในขดลวดทำให้เกิดแรงดูดและแรงผลักของสนามแม่เหล็กทั้งสอง จึงทำให้มอเตอร์หมุน สำหรับมอเตอร์รุ่นต่างๆจะมีคุณสมบัติที่แตกต่างกัน เช่น ขนาดแรงบิดของมอเตอร์ อัตราทดของเกียร์ งานที่นำไปใช้ โดย DC motor ประกอบด้วย 2 ส่วนหลักๆ ได้แก่ โรเตอร์และสเตเตอร์ ส่วนของโรเตอร์คือ ส่วนที่หมุน ส่วนของสเตเตอร์ คือ ส่วนที่เป็นขดลวดที่สร้างสนามแม่เหล็ก นอกจากนี้ยังมีแปรงถ่าน (Brush) ซึ่งเป็นส่วนเชื่อมต่อเพื่อรับพลังงานไฟฟ้าภายนอกไปยังขดลวดของมอเตอร์ เมื่อขดลวดได้รับไฟฟ้ากระแสตรง จะมีถูกเหนี่ยวนำให้เกิดสนามแม่เหล็กรอบๆ รอบขดลวด

ในการใช้งานมอเตอร์กระแสตรงกับบอร์ดไมโครคอนโทรลเลอร์ไม่สามารถต่อใช้งานกันโดยตรงได้ เพราะว่าขา GPIO ของตัวบอร์ดไมโครคอนโทรลเลอร์นั้นไม่สามารถจ่ายกระแสออกมาขนาดที่จะทำให้มอเตอร์ทำงานได้ ต้องใช้งานบอร์ดขับมอเตอร์และแหล่งจ่ายไฟเลี้ยงที่เหมาะสมจากภายนอกเพิ่มเข้ามาด้วย โดยปกติแล้วบอร์ดขับมอเตอร์ 1 บอร์ดสามารถควบคุมมอเตอร์ได้ 1 ตัว หรือ 2 ตัวตามสเปคของบอร์ดขับมอเตอร์ตัวนั้นๆ และการในการควบคุมมอเตอร์ 1 ตัวจะใช้ต้องสายสัญญาณ 3 สายในการควบคุมมอเตอร์ [9]



รูปที่ 11 DC Motor



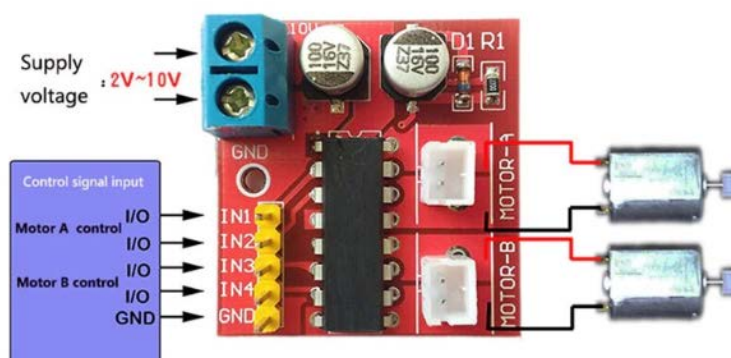
รูปที่ 12 DC Motor เมื่อต่อกับล้อ

6. โมดูลขับมอเตอร์ (DC Motor Speed Control)

บอร์ดขับมอเตอร์ ควบคุมความเร็วได้ สำหรับควบคุมมอเตอร์ 2 ตัว กำลังขับต่อเนื่อง 1.5A สูงสุดที่ 2.5A มีวงป้องกันโหลดและความร้อนเกิน ประหยัดพลังงาน ความร้อนขณะทำงานต่ำ

โดยคณะผู้จัดทำได้เลือกใช้ คือ **Mini-298N 2-Channel PWM Motor Driver** ประกอบด้วย H-Bridge เป็นโมดูลที่ใช้ในการควบคุมความเร็วและทิศทางของมอเตอร์ และยังสามารถนำไปประยุกต์ใช้กับ Project

อื่นได้อีกด้วย เป็นวงจรที่สามารถใช้ควบคุมกระแสได้ทั้งชั่วคราวและลบด้วยการควบคุม pulse width modulation (PWM) เป็นการควบคุมแบบ digital ที่มีการนำมาใช้กันมากโดยส่วนมากเพื่อเป็นการประหยัดพลังงานและ สามารถควบคุม Output ได้ โดยมีการกระตุ้นอย่างต่อเนื่อง ซึ่งทำให้เกิดการสูญเสียพลังงานน้อยมาก แสดงดังรูปที่ 13



รูปที่ 13 โมดูลขับมอเตอร์

ตารางที่ 1 แสดงรายละเอียด Mini-298N 2-Channel PWM Motor Driver

Supply Voltage	2 – 10 V
Signal Input	1.8 – 7 V
Max Output current	3 A (1.5*2 A)
Control signal	PWM

7. เซ็นเซอร์แทรกตามเส้น (TCRT5000 Infrared Reflective sensor)

เซนเซอร์ใช้คลื่นกีดขวาง เส้นขาวดำ นับจำนวน เป็นโมดูลอ่านค่าสะท้อนกลับของแสง ใช้ไฟ 3.3-5V เหมาะสำหรับใช้กับ Arduino ให้เอาต์พุตออกมา 2 แบบ คือ แบบdigital สามารถปรับค่าที่ต้องการได้ เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการก็จะส่งค่า 1 ออกมา ถ้ายังไม่ถึงระดับก็จะส่งค่า 0 ออกมา และอีกแบบคือ เอาต์พุตแบบanalog อ่านค่าได้เป็นตัวเลข 0-1023 หรือสัญญาณไฟในช่วง 0-5V สามารถนำไปประยุกต์กับงานได้หลายแบบ เช่น ใช้เป็นเซนเซอร์หลีกเลี่ยงการชน แสดงดังรูปที่ 14



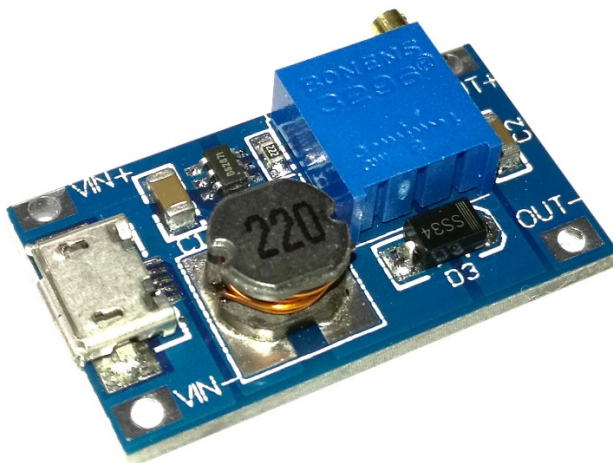
รูปที่ 5 เซ็นเซอร์แทรกตามเส้น

ตารางที่ 2 แสดงรายละเอียด TCRT5000 Infrared Reflective sensor

ไฟเลี้ยง VCC	3.3 – 5 V
ดิจิตอลเอาต์พุต	0 หรือ 1
อนาล็อกเอาต์พุต	0 - 1023

8. วงจร DC/DC Step-up

เป็นแผงวงจรที่เพิ่มแรงดันแบบปรับค่าได้ แสดงดังรูปที่ 15



รูปที่ 15 วงจร DC/DC Step-up รุ่น MT3608

ตารางที่ 3 แสดงรายละเอียด วงจร DC/DC Step-up รุ่น MT3608

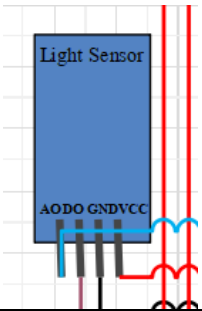
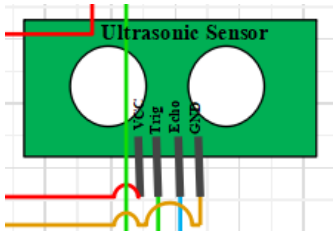
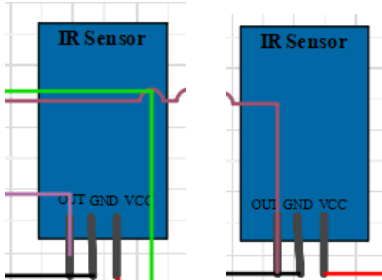

กระแสไฟขาออกสูงสุด	2 A
แรงดันไฟเข้า	2 – 24 V
แรงดันขาออกสูงสุด	28 V
% efficiency	> 93 %

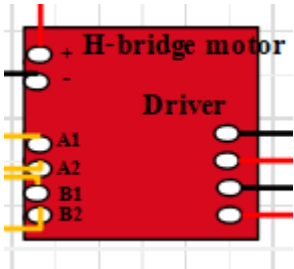
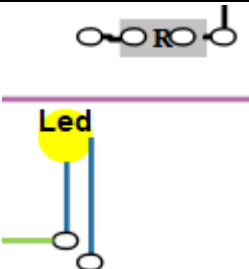
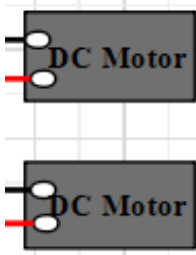
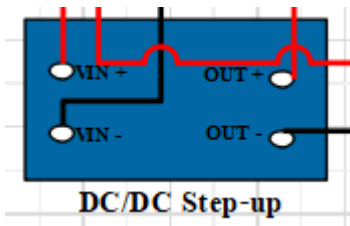
9. เครื่องมือและอุปกรณ์ที่จำเป็น

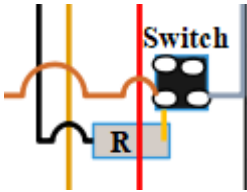
1. Arduino board (LGT8F328P) จ ำนวน 1 ชิ้น
2. 18650 Li-ion battery, Li-ion Charging module, Battery case จำนวนอย่างละ 1 ชิ้น
3. DC Geared-Motors 2 ก้อน
4. H-bridge Driver 1 ชิ้น
5. Breadboard 1 ชิ้น

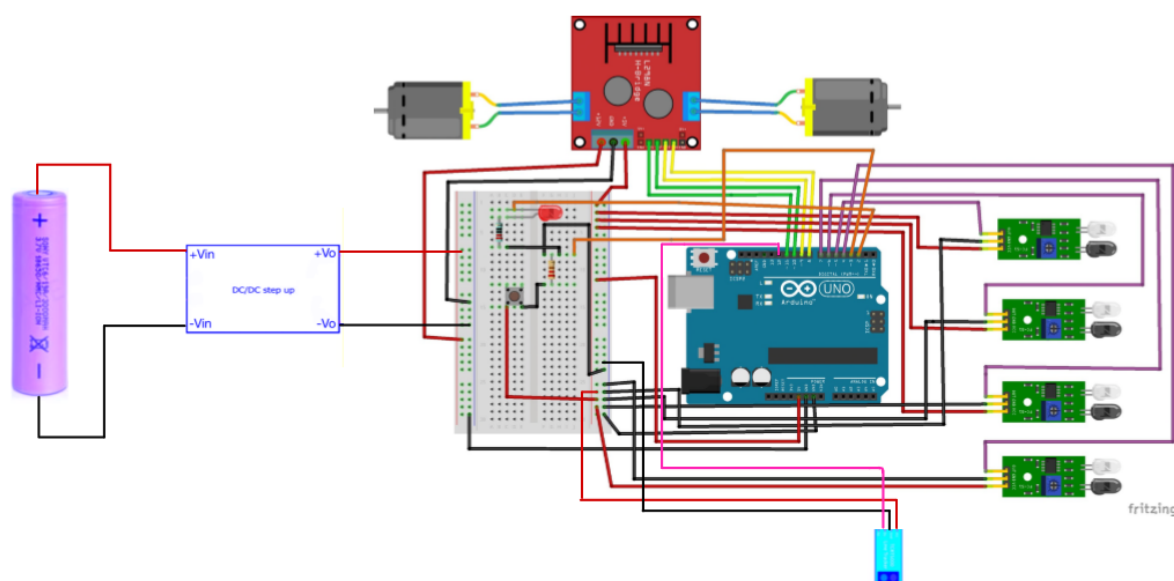
6. DC/DC Step-up Converter 1 ชิ้น
7. TCRT5000 Infrared Reflective sensor 1 ชิ้น
8. IR Infrared Obstacle Avoidance Sensor 2 ชิ้น
9. Ultrasonic Sensor 1 ชิ้น
10. LED และตัวต้านทาน อย่างละ 1 ชิ้น

โดยมีรายละเอียดการต่อวงจรดังนี้

อุปกรณ์	รูปภาพ	การต่อ
TCRT5000 Infrared Reflective sensor		VCC: 5V GND: Ground D0: Digital output (0/1): D13 A0: Analog output: A0
Ultrasonic Sensor		VCC: 5V Trig: Digital output (0/1): D7 Echo: Digital Input (0/1): D8 GND: Ground
IR Infrared Obstacle Avoidance Sensor		OUT: Digital Input (0/1): D4, D6 GND: Ground VCC: 5V
18650 Li-ion battery, Battery case, Li-ion Charging		ขั้วบวก: VIN+ ของ DC/DC Step-up Converter และบวกของ H-bridge Driver

		<p>ขั้วลบ: VIN- ของ DC/DC Step-up Converter</p>
H-bridge Driver		<p>บวก: ขั้วบวกของ 18650 Li-ion battery</p> <p>ลบ: Ground</p> <p>A1: D12</p> <p>A2: D3</p> <p>B1: D10</p> <p>B2: D9</p> <p>4ช่องด้านขวา: Motor 1 and 2</p>
LED และตัวต้านทาน		<p>ขาบวกของ LED: D11</p> <p>ขาลบของ LED: ขาบวกตัวต้านทาน</p> <p>ขาบวกตัวต้านทาน: ขาลบของ LED</p> <p>ขาลบตัวต้านทาน: Ground</p>
DC Geared-Motors		<p>ต่อเข้ากับ H-bridge Driver</p>
DC/DC Step-up Converter		<p>Vin: V battery (-) Ground</p> <p>Vin+: V battery (+)</p> <p>Out-: Ground</p> <p>Out+: VIN (Arduino)</p>

Switchและตัว ต้านทาน		<p>ขาต้านซ้ายของswitch: D2และตัว ต้านทาน</p> <p>ขาต้านขวาของSwitch: 5V</p> <p>ขาบวกของตัวต้านทาน: ขาต้านซ้าย ของswitch</p> <p>ขาลบของตัวต้านทาน: Ground</p>
-------------------------	---	---



รูปที่ 16 แสดงการต่อวงจร

บทที่ 3

Mechanical design

บทที่ 4

Programming Codes

```
#define ia1 25
#define ia2 16
#define ib1 26
#define ib2 17
#define ic1 27
#define ic2 14
#define id1 12
#define id2 13
#define sensor1 18 //left
#define sensor2 19 //front
#define sensor3 23 //back
#define sensor4 05 //right
#define color 36 //จับสี
#define switch 39 // สวิตช์
#define maxSpd 255
int speed = maxSpd;
int led;
int t=0;
int r=0;

void setup() {
  pinMode(ia1, OUTPUT);
  pinMode(ia2, OUTPUT);
  pinMode(ib1, OUTPUT);
  pinMode(ib2, OUTPUT);
  pinMode(ic1, OUTPUT);
  pinMode(ic2, OUTPUT);
  pinMode(id1, OUTPUT);
  pinMode(id2, OUTPUT);
```

```

pinMode(swith, INPUT);
pinMode(color,INPUT);
pinMode(sensor1,INPUT);
pinMode(sensor2,INPUT);
pinMode(sensor3,INPUT);
pinMode(sensor4,INPUT);
Serial.begin(9600);
}

void loop() {
  if (led==HIGH){
    Challenger();
  }
  if(led == LOW){
    defend();
  }
}

void defend(){
  Dsensor1();
  Dsensor2();
  Dsensor3();
}

void Challenger(){
  Csensor2();
  Csensor1();
  Csensor4();
  SensorcolorR();
  Csensor3();
}

void Forward(int speed) //goforward
{
  digitalWrite(ia1, LOW);
  analogWrite(ia2, speed);
}

```

```
digitalWrite(ib1, LOW);
analogWrite(ib2, speed);
digitalWrite(ic1, LOW);
analogWrite(ic2, speed);
digitalWrite(id1, LOW);
analogWrite(id2, speed);
}

void Reward(int speed) //goReward
{
    digitalWrite(ia2, LOW);
    analogWrite(ia1, speed);
    digitalWrite(ib2, LOW);
    analogWrite(ib1, speed);
    digitalWrite(ic2, LOW);
    analogWrite(ic1, speed);
    digitalWrite(id2, LOW);
    analogWrite(id1, speed);
}

void TurnLeft(int speed) //Turnleft
{
    digitalWrite(ia1, LOW);
    analogWrite(ia2, 0.8*speed);
    digitalWrite(ib1, LOW);
    analogWrite(ib2, 0.3*speed);
    digitalWrite(ic1, LOW);
    analogWrite(ic2, 0.8*speed);
    digitalWrite(id1, LOW);
    analogWrite(id2, 0.3*speed);
}

void TurnRight(int speed) //TurnRight
{
    digitalWrite(ia1, LOW);
```

```

    analogWrite(ia2, 0.3*speed);
    digitalWrite(ib1, LOW);
    analogWrite(ib2, 0.8*speed);
    digitalWrite(ic1, LOW);
    analogWrite(ic2, 0.3*speed);
    digitalWrite(id1, LOW);
    analogWrite(id2, 0.8*speed);
}

void Break() // motor break
{
    digitalWrite(ia1, HIGH);
    digitalWrite(ia2, HIGH);
    digitalWrite(ib1, HIGH);
    digitalWrite(ib2, HIGH);
    digitalWrite(ic1, HIGH);
    digitalWrite(ic2, HIGH);
    digitalWrite(id1, HIGH);
    digitalWrite(id2, HIGH);
}

void Dsensor1(){ //sensor1+defend
    int speed = maxSpd;
    if(digitalRead(sensor1)==LOW){
        delay(20);
        if(t<=300){
            Forward(speed);
            delay(10);
            t++;
        }
        if(t==300 && r<=300){
            Reward(speed);
            delay(10);
            r++;
        }
    }
}

```

```

    }
    if(t==300 && r==300){
        t=0;
        r=0;
    }
}
else{
    Break();
}
}

void Dsensor2(){
    int speed = maxSpd;
    int sensorvalue = analogRead(A0);
    if(digitalRead(sensor2)==LOW){
        delay(20);
        Forward(speed);
    }
    if(digitalRead(sensor2)==LOW && sensorvalue >=3600&&sensorvalue <=3900){
        delay(20);
        Reward(speed);
    }
    if(digitalRead(sensor2)==HIGH){
        delay(20);
        Break();
    }
}

void Dsensor3(){
    int speed = maxSpd;
    int sensorvalue = analogRead(A0);
    if(digitalRead(sensor3)==LOW){
        delay(20);
        Reward(speed);
    }
}

```

```

}
if(digitalRead(sensor3)==LOW && sensorvalue >=3600&&sensorvalue <=3900){
    delay(20);
    Forward(speed);
}
if(digitalRead(sensor3)==HIGH){
    delay(20);
    Break();
}
}
void Csensor2(){
    int speed = maxSpd;
    if(digitalRead(sensor2)==LOW){
        delay(20);
        Reward(speed);
    }
    else{
        Forward(speed);
    }
}
void Csensor1(){
    int speed = maxSpd;
    if(digitalRead(sensor1)==LOW){
        delay(20);
        TurnRight(speed);
    }
    else {
        Forward(speed);
    }
}
void Csensor4(){
    int speed = maxSpd;

```



```

if(digitalRead(sensor4)==LOW){
    delay(20);
    TurnLeft(speed);
}
else {
    Forward(speed);
}
}

void SensorcolorR(){
    int speed = maxSpd;
    int sensorvalue = analogRead(A0);
    if(sensorvalue>=2200&&sensorvalue<=2500){
        delay(20);
        Reward(speed);
    }
    else {
        Forward(speed);
    }
}

void Csensor3(){
    int speed = maxSpd;
    int sensorvalue = analogRead(A0);
    if(digitalRead(sensor3)==LOW){
        delay(20);
        Reward(speed);
    }
    if(digitalRead(sensor3)==HIGH){
        delay(20);
        Forward(speed);
    }
    if(digitalRead(sensor3)==HIGH &&sensorvalue>=2200&&sensorvalue<=2500){
        delay(20);

```

```
        Break();  
    }  
}
```

เอกสารอ้างอิง

- [1] <https://www.applicadthai.com/editor-talks/เทคโนโลยีกับชีวิต>
- [2] <https://www.mmthailand.com/พื้นฐานหุ่นยนต์-อุตสาหกรรม-01/>
- [3] <http://krunisit.rwb.ac.th/robot.html>
- [4] <https://arduinothing.blogspot.com/2016/04/arduino-cc.html>
- [5] <http://www.robotsiam.com/product/118/wemos-ttgo-esp32-wifi-bluetooth-uno-d1-r32>
- [6] <https://www.rkbbattery.com/2019/03/28/lithium-ion-battery-18650/>
- [7] <https://www.thaicconverter.com/article/8/มาทำความรู้จักกับแบตเตอรี่-18650>
- [8] <https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html>
- [9] <http://www.arduino.codemobiles.com/product/8/infrared-reflectance-sensor-tcrt5000-ใช้เซ็นเซอร์ตรวจจับวัตถุ-หรือ-นับจำนวน>
- [10] <https://www.arduitronics.com/product/1295/ir-reflective-obstacle-avoidance-line-tracking-sensor-tcrt5000>
- [11] <https://www.thaieasyelec.com/article-wiki/embedded-electronics-application/14-espino32-motor.html>
- [12] <http://www.se-edstemeducation.com/รถบังคับขับเคลื่อน-2-ล้อควบคุมด้วยโทรศัพท์มือถือ-android/>
- [13] <http://aimagin.com/blog/motor/?lang=th>