

หุ่นยนต์ G-10

Robot G-10

โดย

นาย ธรรมนูญ เกิดมั่งมี
นาย ภูวิศ รัวิังสรรค์
นาย ศุภวิชญ์ เมณฑกา

โครงการนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2562

กิตติกรรมประกาศ

การดำเนินโครงการ “หุ่นยนต์ G-10 ” จะไม่สามารถสำเร็จลุล่วงไปได้ หากไม่ได้รับความอนุเคราะห์ เป็นอย่างดีจาก พศ.ดร. อุดม จันทร์จรัสสุข ที่ให้คำแนะนำและแนวทางการแก้ปัญหา รวมไปถึงการสนับสนุน เครื่องมือและอุปกรณ์ต่าง ๆ ที่ใช้ในโครงการนี้ ทำให้สามารถทำงานสำเร็จลุล่วงไปได้ด้วยดี

ธรรมนูญ เกิดมั่งมี
ภูวิศ ริวิรังสรรค์
ศุภวิชญ์ เมฆพาก
ผู้จัดทำ

โดย	นาย	ธรรมนูญ	เกิดมีมี	60010441
	นาย	ภูวิศ	รัวิรังสรรค์	60010816
	นาย	ศุภวิชญ์	เมณฑกา	60011006
อาจารย์ที่ปรึกษา		ผศ.ดร. อุดม จันทร์จรัสสุข		

บทคัดย่อ

เทคโนโลยีในปัจจุบันเกี่ยวกับหุ่นยนต์นั้นไม่ใช่เรื่องใหม่สำหรับอุตสาหกรรม ทำให้ในปัจจุบันการพัฒนาหุ่นยนต์ให้เข้ามามีส่วนร่วมในการผลิตในโรงงานมากยิ่งขึ้นเพื่อตอบโจทย์ความต้องการของผู้ผลิตที่ต้องการจะเพิ่มอัตราในการผลิต (Capacity Utilization) และ ใน ณ เดียวกันก็ต้องการเพิ่มผลิตภาพ (Productivity) ให้สูงที่สุดภายใต้เงื่อนไขที่มีการลงทุนต่ำที่สุดเท่าที่จะเป็นไปได้แต่มีกำไรสูง โครงการจึงมีวัตถุประสงค์เพื่อศึกษาการสร้างหุ่นยนต์ให้เป็นพื้นฐานในการต่อยอดการทำหุ่นยนต์เพื่อเพิ่มประสิทธิภาพในการผลิตในอนาคต โดยหุ่นยนต์ควบคุมการทำงานด้วยวงจรไฟฟ้าที่ได้ทำการออกแบบเองกับไมโครคอนโทรลเลอร์ นอกจากนี้หุ่นยนต์ถูกสร้างขึ้นเพื่อใช้ในการแข่งขันลักษณะคล้ายกับการเล่นบอลลูนด้านหรือเล่นเตะ โดยแบ่งเป็นทีมรุกและทีมรับ สถาบันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว จะดำเนินการตามแผนกลยุทธ์รุกและกลยุทธ์รับที่กำหนดไว้ใช้ในการแข่งขัน ซึ่งหุ่นยนต์ที่ใช้ในการแข่งขันครั้งนี้มีความกว้าง 8.8 เซนติเมตร ความยาว 9.1 เซนติเมตรและความสูง 4.6 เซนติเมตร

คำหลัก อัตราในการผลิต, ผลิตภาพ, ไมโครคอนโทรลเลอร์, กลยุทธ์

Abstract

Industry has included robotic technology since the past. Now, robotic technology in industry is being developed and being included more to manufacturing production in order to stratify manufacture demand which is maximizing capacity utilization and productivity with minimizing the cost and maximizing the profit. This project purpose to study about robot building for gain the efficiency in the future production. The robot is controlled by designed electrical circuit and microcontroller. Furthermore, robot is build for competition which is Toey; the game is divided in to two strategy, the first one is passing the obstructs and second is partition, meaning being an obstruct. During the competition, all robot will be divide into defensive (obstruct) and offensive. Each team includes 7 robots. Robot must be 8.8 cm width and 9.1 cm long and 4.6 cm height.

Key words Capacity Utilization, Productivity, Microcontroller, strategy

สารบัญ

หน้า

กิตติกรรมประกาศ

บทคัดย่อ

สารบัญ

สารบัญรูป

สารบัญตาราง

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

1.2 ปัญหา

1.3 วัตถุประสงค์

1.4 ขอบเขตของโครงการ

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ภาษา C/C++

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

2.3 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง

(IR Infrared Obstacle Avoidance Sensor Module)

2.4 เซ็นเซอร์วัดระยะทาง (Ultrasonic Module)

2.5 DC Motor Speed Control

2.6 TCRT5000 Infrared Reflective sensor

2.7 วงจร DC/DC Step-up (แรงดันปรับค่าได้)

2.8 ภาษา C#

บทที่ 3 การออกแบบและการจัดทำโครงงาน

3.1 การออกแบบการทำงานของกลยุทธ์โดยรวม

3.2 การออกแบบการทำงานของกลยุทธ์รุก

3.3 การออกแบบการทำงานของกลยุทธ์รับ

3.4 การออกแบบการทำงานของวงจร

3.5 การออกแบบชิ้นส่วนของหุ่นยนต์

สารบัญ (ต่อ)

หน้า

บทที่ 4

ผลการทดลอง

4.1 การทดสอบ Sensor วัดระยะทางแบบ digital

(IR Infrared Obstacle Avoidance Sensor Module)

4.2 การทดสอบ TCRT5000 Infrared Reflective Sensor

บรรณานุกรม

ภาคผนวก ก คำสั่งในส่วนของไมโครคอนโทรลเลอร์

ภาคผนวก ข เปสบอร์ดและเซนเซอร์ที่ใช้ติดตั้งบนหุ่นยนต์

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

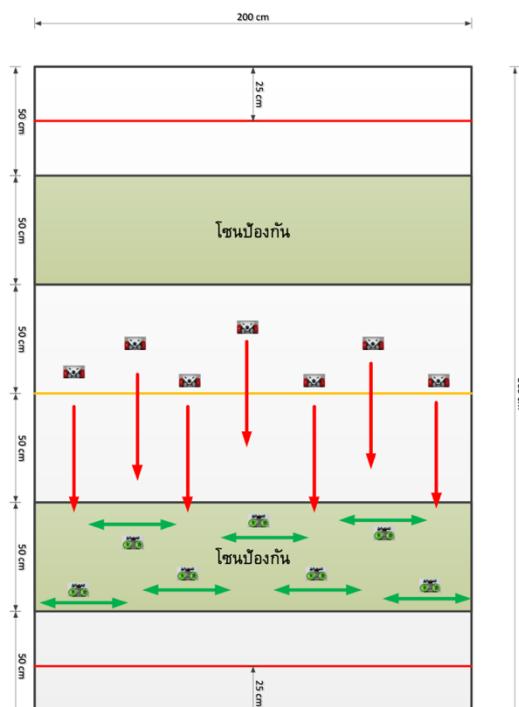
คำว่า “หุ่นยนต์” นั้นแปลความหมายได้หลายอย่าง ในแต่ละบุคคล แต่ใจความสำคัญก็คือ หุ่นยนต์คือ เครื่องจักร ดังนั้นหุ่นยนต์ สามารถโปรแกรมให้มีหน้าที่การทำงานในด้านต่าง ๆ โดยอัตโนมัติหรือตามการควบคุมโดยตรงของมนุษย์ ซึ่งเทคโนโลยีในปัจจุบันเกี่ยวกับหุ่นยนต์นั้นไม่ใช่เรื่องใหม่สำหรับอุตสาหกรรม ทำให้ในปัจจุบันการพัฒนาหุ่นยนต์ให้เข้ามามีส่วนร่วมในการผลิตในโรงงานมากยิ่งขึ้นเพื่อตอบโจทย์ความต้องการของผู้ผลิตที่ต้องการเพิ่มอัตราในการผลิต (Capacity Utilization) และ ใน ณ เดียวกันก็ต้องการเพิ่มผลิตภาพ (Productivity) ให้สูงที่สุดภายใต้เงื่อนไขที่มีการลงทุนต่ำที่สุดเท่าที่จะเป็นไปได้แต่มีกำไรสูง [1]

ปฏิเสธไม่ได้เลยว่าเทคโนโลยีเป็นปัจจัยที่สำคัญต่อชีวิตมนุษย์ในปัจจุบัน และมูลเหตุที่โรงงานอุตสาหกรรมหลาย ๆ แห่ง เริ่มตระหนักถึงประโยชน์ในการใช้หุ่นยนต์เข้ามายึดทบทาในการทำงานมากยิ่งขึ้น ณ ขณะนี้คงมีหลายสาเหตุดังนี้ 1. ประเทศไทยกำลังก้าวสู่สังคมผู้สูงอายุ ทำให้มีแนวโน้มการขาดแคลนจำนวนแรงงานคนในอนาคต ดังนั้น ภาคอุตสาหกรรมจำนวนมากจึงมีความต้องการ ระบบอัตโนมัติมากขึ้น เพื่อเข้ามาช่วยลดแทนแรงงานคนที่เริ่มลดน้อยลงเหล่านั้น 2. อัตราค่าแรงงานขั้นต่ำมีแนวโน้มปรับสูงขึ้น ซึ่งอาจไม่สอดคล้องกับทักษะฝีมือในบางตำแหน่งงาน เช่นคนแบกของหรือขันถ่ายสินค้าดังนั้นในอนาคตหากผู้ประกอบการพิจารณาปรับเปลี่ยนมาใช้ระบบอัตโนมัติ ในการลำเลียงสินค้าทดแทนแรงงานคนอาจจะเกิดความคุ้มค่าและช่วยลดต้นทุนแรงงานได้ในระยะยาว 3. ภาคอุตสาหกรรมไทยมีการใช้หุ่นยนต์ และระบบอัตโนมัติในการบริหารผลิตค่อนข้างน้อย คิดเป็นสัดส่วนเพียงร้อยละ 15 ของโรงงานทั้งหมด จึงแสดงให้เห็นถึงโอกาส และความต้องการอีกจำนวนมาก หากในการปรับเปลี่ยนมาใช้หุ่นยนต์และระบบอัตโนมัติ เพื่อเพิ่มประสิทธิภาพในการทำงานในภาคอุตสาหกรรมการผลิต และธุรกิจบริการ ซึ่งปัจจุบันประเทศไทยมีอัตราการเติบโตของหุ่นยนต์อุตสาหกรรมเฉลี่ยร้อยละ 19 ต่อปี เนื่องจากปี ค.ศ. 2016 ไทยผลิตหุ่นยนต์ได้ 2,646 หน่วย และจะเพิ่มเป็น 5,000 หน่วย ในปี ค.ศ. 2020 [2]

1.2 ปัญหา

การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่นบล็อกด่าน หรือเล่นเตยโดยแบ่งเป็นทีมรุกและทีมรับ สลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้าม จนผ่านเส้นแดงแล้วกลับมาอย่างปลอดภัย (ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้ก็จะเป็นฝ่ายชนะในการแข่งขัน รอบนั้น หุ่นยนต์ที่ถูกจับได้จะถูกตัดออกจากการแข่งขันในรอบนั้น ส่วนทีมรับจะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้นถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจาก การแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มี

หุ่นยนต์ตัวไหน สามารถผ่านด่านได้ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น ภายในตัวหุ่นยนต์ขนาด 10×10 (ไม่กำหนดความสูง) ลูกบาศก์เซนติเมตร ระบบขับเคลื่อนของหุ่นยนต์จะต้องใช้มอเตอร์ที่จัดให้ 2 ตัว ใช้พลังงานจากถ่านชาาร์จที่หาให้เท่านั้น ซึ่งเป็นถ่านชาาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V ที่สามารถเพิ่มอุปกรณ์อื่น ๆ ตามหน้าที่การทำงาน นอกเหนือจากข้อ กำหนดที่กำหนดให้ เช่น sensor ต่าง ๆ ไฟ เสียง จอแสดงผล เป็นต้น และภายในตัวหุ่นยนต์จะมีเซ็นเซอร์ที่สามารถตรวจจับความเคลื่อนไหวในพื้นที่ ความสูงไม่ต่ำกว่า 20 เซนติเมตร ส่วนพื้นจะใช้เทปสีในการแบ่งโซนเส้น เทปมีความกว้าง 3.5 เซนติเมตรสามารถแสดงสนามการแข่งขันได้ดังรูปที่ 1



รูปที่ 1 สนามการแข่งขัน

1.3 วัตถุประสงค์

- 1) เพื่อฝึกฝนการทำงานเป็นทีมอย่างเป็นระบบ
- 2) เพื่อชัยชนะในการแข่งขันหุ่นยนต์
- 3) เพื่อศึกษาการสร้างหุ่นยนต์ให้เป็นพื้นฐานในการต่อยอดการทำหุ่นยนต์เพื่อเพิ่มประสิทธิภาพในการผลิตในอนาคต
- 4) เพื่อจำลองจำลองการทำงานของวิศวกรที่ใช้ควบคุมหุ่นยนต์
- 5) เพื่อศึกษาการทำงานและเขียนโปรแกรมไมโครคอนโทรลเลอร์
- 6) เพื่อศึกษาการทำงานของเซ็นเซอร์ชนิดต่าง ๆ

1.4) ขอบเขตโครงการ

ทำการออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลยุทธ์ออกเป็น 2 กลยุทธ์ คือ กลยุทธ์รุกและกลยุทธ์รับ ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ ด้วยขนาดของหุ่นยนต์ ความกว้าง 8.8 เซนติเมตร ความยาว 9.1 เซนติเมตรและความสูง 4.6 เซนติเมตร

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงการ “หุ่นยนต์ G-10” ได้จัดทำขึ้นเพื่อออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลยุทธ์ออกเป็น 2 กลยุทธ์ คือ กลยุทธ์รุกและกลยุทธ์รับ ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ จึงมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ภาษา C/C++

ภาษา Arduino (หรือ ภาษา C/C++) ใช้สำหรับเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ Arduino IDE เป็นเครื่องมือสำหรับเขียนโปรแกรมด้วยภาษา Arduino คอมไพล์โปรแกรม (Compile) และอัปโหลดโปรแกรมลงบอร์ด (Upload)

1) พังก์ชันหลัก (Structure)

เป็นพังก์ชันหลักในการเขียนโปรแกรม จำเป็นต้องมีในทุกโปรแกรม

Setup () คือ พังก์ชันใช้ในการประกาศค่าเริ่มต้น ตำแหน่งพอร์ตที่ใช้งาน รวมถึงพังก์ชันที่อยู่แลบารีที่ใช้งาน เป็นพังก์ชันที่ทำงานเพียงครั้งเดียว จะทำงานทุกครั้งที่มีการรีเซ็ต หรือรีบูตเครื่องใหม่เท่านั้น

Loop () คือ พังก์ชันใช้ในการเขียนโค้ดโปรแกรมการทำงานของ Arduino เป็นพังก์ชันการวนลูปไปเรื่อยๆ

2) ชุดคำสั่งในการควบคุม (Control Structures)

เป็นชุดคำสั่งในการใช้ในการตัดสินใจทางออก เพื่อใช้ในการทำงาน

If คือ คำสั่งในการตัดสินใจแบบตัวเลือกเดียว โดยใช้งานร่วมกับ and, or not, ==, !=, <, > เพื่อใช้ในการตัดสินใจในการหาคำตอบ

If...else คือ คำสั่งในการตัดสินใจแบบหลายตัวเลือก โดยใช้งานร่วมกับ and, or not, ==, !=, <, > เพื่อใช้ในการตัดสินใจในการหาคำตอบ

for คือ คำสั่งกำหนดเงื่อนไขเป็นจำนวนครั้งที่จะทำงานซุดคำสั่งต่าง ๆ ภายในลูป หมายเหตุที่จะใช้กับงานประเภทที่ไม่มีการเปลี่ยนแปลง

switch case คือ คำสั่งที่ใช้ในการจัดการเงื่อนไขหลายเงื่อนไข โดยเฉพาะการใช้งานโครงสร้าง การจำแนกเงื่อนไขมาจำเป็นต้องอาศัยเฉพาะตัวแปรที่เก็บค่าจำนวนเต็มเท่านั้น ข้อมูลแบบอื่นก็สามารถใช้ได้เช่นกัน

while คือ คำสั่งเงื่อนไขที่จะทำการตรวจสอบว่าเป็นจริงหรือเท็จ ชุดคำสั่งก็คือ ส่วนที่ทำงานซ้ำ ๆ โดยจะต้องมีคำสั่งที่จะทำให้เงื่อนไขเป็นเท็จด้วย

#define คือ คำสั่งกำหนดค่านิพจน์ต่าง ๆ ให้กับชื่อของตัวคงที่

#include การกำหนดชื่อไฟล์ตามหลัง include จะใช้เครื่องหมาย <> ซึ่งจะเป็นการอ่านไฟล์จาก ไดเร็กทอรี หรือโฟลเดอร์ที่กำหนดไว้ก่อนแล้ว โดยปกติจะเป็นโฟลเดอร์ include แต่ถ้าใช้เครื่องหมาย “ ” เป็น การอ่านไฟล์จาก โฟลเดอร์ หรือไดเร็กทอรี ที่กำลังติดต่ออยู่และไฟล์ที่จะ include เข้ามานี้จะต้องไม่มีฟังก์ชัน main () โดยมากจะประกอบไปด้วยโปรแกรมย่อย ค่าคงที่ หรือข้อความต่าง ๆ

3) Conversion การแปลงค่า

char () คือ การแปลงค่าข้อมูลให้เป็น character

byte () คือ การแปลงค่าข้อมูลให้เป็น byte

int () คือ การแปลงค่าข้อมูลให้เป็น integer

4) Time

delay () คือ คำสั่งหยุดการทำงานโปรแกรมสำหรับจำนวนของเวลา (ใน milliseconds)

Milliseconds = จำนวนมิลลิวินาทีในการหยุดการทำงานชั่วคราว

2.2.5 Functions

pinMode () ใช้ในกลุ่ม void setup () เพื่อกำหนดหน้าที่ขาของไมโครคอนโทรลเลอร์ให้เป็น ขารับ สัญญาณ INPUT หรือขาส่งสัญญาณ OUTPUT

digitalWrite () คือ การส่งค่าลอจิก HIGH หรือ LOW (เปิด หรือปิด) ไปยังขา digital ที่กำหนด หมายเลขขาไปซึ่งอาจกำหนดเป็นตัวแปรหรือค่าคงที่ (0-13) [3]

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ (Microcontroller) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับ ระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำและพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกันโครงสร้างโดยทั่วไปของ ไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

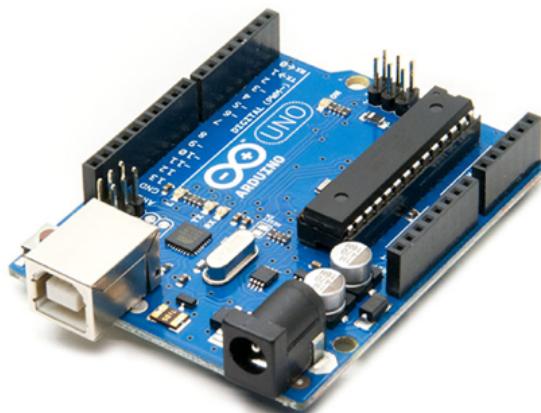
1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)
2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกับกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรง (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรง ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีก

(EEPROM : Electrically Erasable Programmable Read-Only Memory) ซึ่งสามารถเก็บ ข้อมูลได้แม้ไม่มีไฟเลี้ยงก็ตาม

3. ส่วนติดต่อ กับ อุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณ หรือพอร์เตาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผล และส่งไปพอร์เตาต์พุต เพื่อแสดงผล เช่น การติดสว่างของหลอดไฟ เป็นต้น

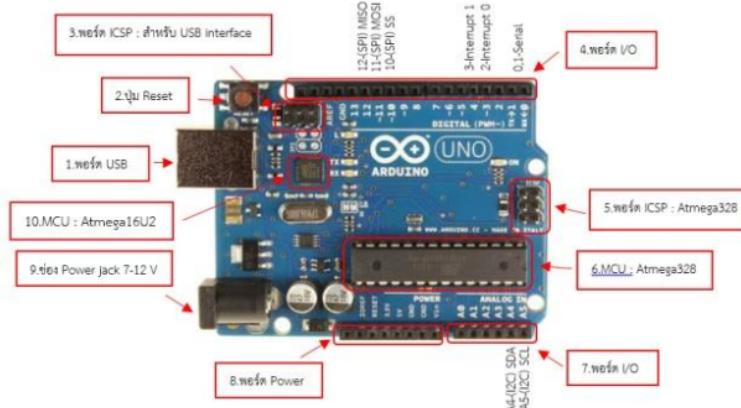
4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ชิปยู หน่วยความจำ และพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และ บัสควบคุม (Control Bus)

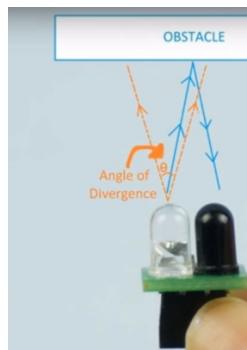
5. วงจรกำเนิดสัญญาณนาฬิกา เป็นองค์ประกอบที่สำคัญมากอีks่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวัดการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วยสำหรับไมโครคอนโทรลเลอร์ที่ผู้จัดทำเลือกใช้ คือ Arduino R3 ATMega 328p U แสดงดังรูปที่ 2 [4]



รูปที่ 2 คือ Arduino R3 ATMega 328p U

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ระดับ AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผย ข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสม สำหรับผู้เริ่มต้นศึกษา ซึ่งส่วนประกอบของ Arduino R3 ATMega 328p U แสดงดังรูปที่ 3 และรายละเอียด แสดงดังตารางที่ 1 [5][6]





รูปที่ 5 การส่ง และรับสัญญาณ infrared

ภายใต้เซ็นเซอร์แบบนี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้มีจำเป็นต้องเดินสายไฟทั้งสองฝั่ง เมื่อเป็นแบบ Opposed Mode ทำให้การติดตั้งใช้งานได้่ายกว่า แต่อย่างไรก็ตาม จำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโพโตเซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสมสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และ ทำให้ทำงานผิดพลาดได้ เช่นเซอร์แบบนี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับตัวแผ่นสะท้อน Reflector อุ่น ตลอดเวลา จะแสดงค่า เป็น 0 หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะอยู่ตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะทำการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้ จะแสดงค่า เป็น 1 ซึ่งจะทำให้ງดresponsive ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาร์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate ซึ่งคุณสมบัติของ IR Infrared Obstacle Avoidance Sensor Module สามารถแสดงได้ดังตารางที่ 2 [7]

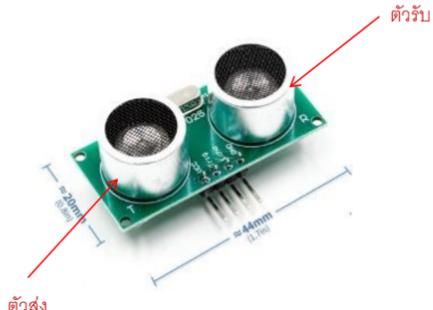
ตารางที่ 2 คุณสมบัติของ IR Infrared Obstacle Avoidance Sensor Module

ไฟเลี้ยง VCC :	3.3-5V
ดิจิตอลเอาต์พุต (0 หรือ 1)	
ระยะตรวจจับ สามารถปรับได้ตั้งแต่ 2-30 cm	
มุมในการตรวจจับ 35 องศา	
ขนาดบอร์ด 3.1 x 1.5 cm	

2.4 เซ็นเซอร์วัดระยะทาง Ultrasonic Module

เซ็นเซอร์วัดระยะทางด้วย Ultrasonic ใช้หลักการ ส่งคลื่นเสียงความถี่ต่ำ Ultrasonic ไปเมื่อคลื่นเสียงกระทบกับวัตถุจะมีการสะท้อนกลับมา เซ็นเซอร์จับเวลาที่ส่งคลื่นเสียงออกไปจนถึงคลื่นเสียงสะท้อนกลับมา เมื่อนำมาคำนวณกับเวลาที่เสียงเดินทางในอากาศ ก็จะได้ระยะทางออกมา

โมดูล Ultrasonic ตรวจจับวัตถุ คำนวณระยะทางโดยใช้คลื่น มีลักษณะเป็นกรวยและไม่มีช่องส่งสัญญาณ จึงเหมาะสมสำหรับใช้ตรวจจับสิ่งกีดขวางด้วย โดยมีส่วนประกอบโมดูล Ultrasonic ดังรูปที่ 6 และแสดงคุณสมบัติดังตารางที่ 3 [8]



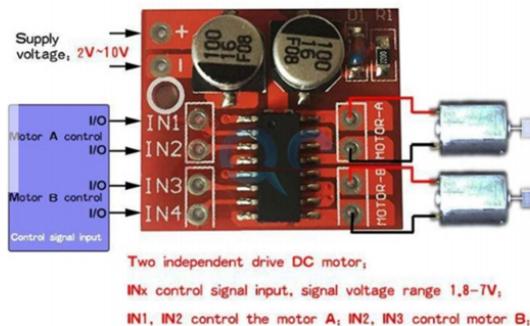
รูปที่ 6 ส่วนประกอบโมดูล Ultrasonic

ตารางที่ 3 คุณสมบัติเซ็นเซอร์วัดระยะทาง Ultrasonic Module

ไฟเลี้ยง VCC :	3.3-5V
ดิจิตอลเอาต์พุต (0 หรือ 1)	
ดิจิตอลอินพุต (0 หรือ 1)	
HC-SR04 :	ระยะตรวจจับ 2-400 cm
US-025 :	ระยะตรวจจับ 2-600 cm

2.5 DC Motor Speed Control

ประกอบด้วย 2 ส่วนใหญ่ ๆ คือ 1) H-bridge Driver และ 2) Pulse-width modulation (PWM) สิ่งที่เป็นพลังขับเคลื่อนหลักให้กับหุ่นยนต์นั้น ก็คงจะไม่พ้น มอเตอร์ ซึ่งต้องการควบคุม จาก Motor Driver ที่จะมาควบคุม ทิศทาง และ ความเร็ว ของมอเตอร์ซึ่งทางผู้จัดทำได้เลือกใช้ L298N Dual H-Bridge Motor Controller ซึ่งสามารถแสดงได้ดังรูปที่ 7 และ คุณสมบัติของ L298N Dual H-Bridge Motor Controller ดังตารางที่ 4 โดยหลักการทำงานของ H-Bridge ของ L298N จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วยลอกิจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ PWM (Pulse Width Modulation) เป็นวิธีการควบคุมการจ่ายกำลังโดยการปรับความกว้างของ สัญญาณ Pulse ด้วยความถี่สูงเพื่อให้ได้กำลังเฉลี่ยเป็นไปตามส่วนที่ต้องการ ซึ่งต้องมีการปรับความถี่ให้เหมาะสมกับเป็นพารามิเตอร์ที่ใช้กำหนดสัดส่วนการทำงาน (ON) ของ Load (มอเตอร์) [9][10]



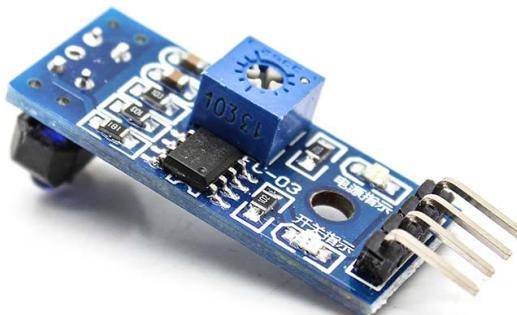
รูปที่ 7 ส่วนประกอบของ L298N Dual H-Bridge Motor Controller

ตารางที่ 4 คุณสมบัติ L298N Dual H-Bridge Motor Controller

Supply voltage :	2-10V
Signal input voltage:	1.8-7V
Max output current :	3A (1.5A*2)
Control signal:	PWM

2.6 TCRT5000 Infrared Reflective sensor

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็กสะดวกในการนำไปใช้ติดตั้งกับงานจำพวกหุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น ซึ่งสามารถแสดงได้ดังรูปที่ 8 และคุณสมบัติตั้งตารางที่ 5 โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณเป็นแสงอินฟราเรด ออกไปต่อกกระหบวัตถุที่ ตรวจพบในระยะ และทำการสะท้อนกลับมาอย่างตัวหลอดไฟต้องได้รับแสงอินฟราเรด โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal แต่สำหรับบางโมดูลอาจจะรองรับ output แบบ Analog signal ด้วย ส่วนตัว R ปรับค่านี้ใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจวัตถุของตัวเซนเซอร์



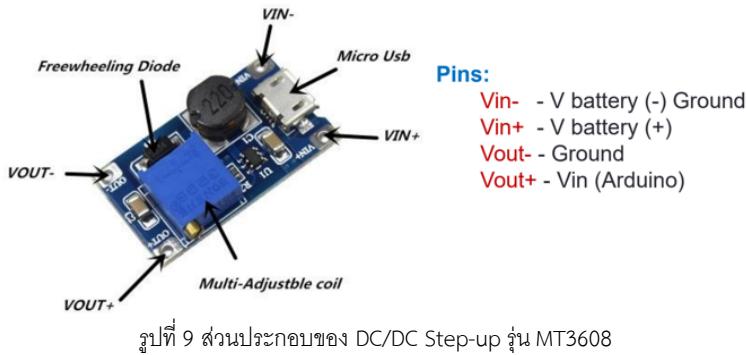
รูปที่ 8 ส่วนประกอบของ TCRT5000 Infrared Reflective sensor

ตารางที่ 4 คุณสมบัติเซ็นเซอร์วัดระยะทาง TCRT5000 Infrared Reflective sensor

ไฟเลี้ยง VCC :	3.3-5V
ดิจิตอลเอาต์พุต (0 หรือ 1)	
อนาล็อกเอาต์พุต	

2.7 วงจร DC/DC Step-up (แรงดันปรับค่าได้)

เป็นวงจรที่ทำหน้าที่เพิ่มแรงดันไฟฟ้า โดยที่ทางผู้จัดทำเลือกใช้เป็นรุ่น MT3608 สามารถแสดงส่วนประกอบได้ดังภาพที่ 9 และสามารถและคุณสมบัติตั้งตารางที่ 6 [11]



รูปที่ 9 ส่วนประกอบของ DC/DC Step-up รุ่น MT3608

ตารางที่ 6 คุณสมบัติของ DC/DC Step-up รุ่น MT3608

กระแสไฟขาออกสูงสุด(I max) :	2 A
แรงดันไฟฟ้าเข้า (V in):	2 V ~ 24 V
แรงดันขาออกสูงสุด(V Out max)	28 V
ประสิทธิภาพ (%Eff)	> 93%

โดยกฎของโอห์มกระแสไฟฟ้านั่นจะไฟฟ้านั้น จะแปรผัน ตรงกับ แรงดันของแหล่งจ่ายไฟแต่จะแปรผกผันกับค่า ความต้านทานในวงจรไฟฟ้า” ดังสมการ [12]

$$I = \frac{V}{R} \quad (1)$$

เมื่อ I = กระแสไฟฟ้ามีหน่วยเป็น แอมป์แพร์ (A)
 V = แรงดันไฟฟ้ามีหน่วยเป็นโวลต์ (V)
 R = ความต้านทานมีหน่วยเป็น โอห์ม (Ω)

ข้อควรระวังเมื่อแรงดันเพิ่ม กระแสไฟฟ้าจะต้องลดลงดังตัวอย่างที่แสดงในรูปที่ 10



ถ้าประสิทธิภาพอยู่ที่ 80% กระแส output จะเหลือ 0.4A

รูปที่ 10 ตัวอย่างความสัมพันธ์ระหว่างแรงดันเพิ่มกับกระแสไฟฟ้าคง

2.8 ภาษา C#

ภาษา C# (C-Sharp) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่ใช้สำหรับเขียนโปรแกรมคอมพิวเตอร์ ซึ่งได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ในปัจจุบัน และเป็นภาษาโปรแกรมคอมพิวเตอร์ที่เหมาะสมสำหรับผู้ที่เริ่มต้นสนใจที่จะเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างยิ่ง ซึ่งภาษา C# ถูกพัฒนามาจากภาษา C++ และมีโครงสร้างแบบเชิงวัตถุ (object-oriented programming) โดยใช้ Visual Studio เป็นเครื่องมือสำหรับพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่ง Visual Studio เป็นเครื่องมือที่ค่อยอำนวยความสะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมคอมพิวเตอร์ได้ไม่ยากนัก

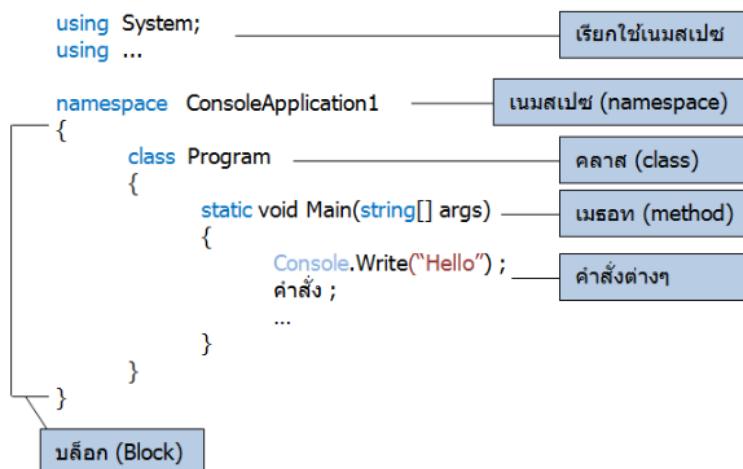
ภาษา C# ได้รวมรวมข้อดีของภาษาต่าง ๆ เข้าไว้ด้วยกัน ไม่ว่าจะเป็นภาษา Java ภาษา C และภาษา C++ โดยมีข้อดีดังนี้

1) เป็นภาษาที่เขียนง่าย ไม่ซับซ้อนและเรียบง่าย เพราะคล้ายภาษา Java ภาษา C และภาษา C++ ทำให้หลายคนเข้าใจได้ง่าย

2) เป็นภาษาโปรแกรมคอมพิวเตอร์ยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์ ภายใต้แนวคิด .NET Framework ซึ่งเป็นแนวคิดที่ได้รับความนิยมสูงที่สุดในปัจจุบัน

3) เป็นภาษาที่ถูกออกแบบมาให้ทำงานบน .NET Framework โดย .NET Framework เป็นรูปแบบในการพัฒนาโปรแกรมคอมพิวเตอร์สมัยใหม่ ซึ่งบริษัทไมโครซอฟท์เป็นผู้พัฒนา คุณสมบัติที่สำคัญของ .NET Framework ก็คือ ผู้ใช้งานสามารถใช้งานบนระบบฮาร์ดแวร์ (Hardware) หรือ ระบบปฏิบัติการ (Operating System) ที่แตกต่างกันได้อย่างไม่มีปัญหา ดังนั้นผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมคอมพิวเตอร์ใหม่ๆ ได้โดยง่าย รวดเร็ว และไม่ต้องติดข้อจำกัดต่าง ๆ อย่างเช่นการเขียนโปรแกรมคอมพิวเตอร์ในสมัยก่อนอีกต่อไป

4) เป็นภาษาที่แข็งแกร่ง เพราะเป็นภาษาที่ได้มีการแก้ไขข้อบกพร่องบางอย่างของภาษา Java ภาษา C และภาษา C++ เหล่านั้น ทำให้ ภาษา C# เป็นภาษาที่มีความสมบูรณ์ตามแบบฉบับของโครงสร้างแบบเชิงวัตถุ (object-oriented programming) [13] ซึ่งโครงสร้างคำสั่งของโปรแกรมภาษา C# แสดงในรูปที่ 11



รูปที่ 11 โครงสร้างคำสั่งของโปรแกรมภาษา C#

Method ในภาษา C# เมธอดนี้เป็นส่วนหนึ่งของโปรแกรมเพื่อให้ดำเนินการบางอย่าง กล่าวอีกนัยหนึ่ง คือ ส่วนย่อยของโปรแกรมที่ถูกเรียกโดยโปรแกรมหลัก และสามารถนำกลับมาใช้ซ้ำได้ การสร้างเมธอดจะช่วยให้ลดจำนวนของชอสโค้ดที่เขียนขึ้น ช่วยการนำกลับมาใช้ใหม่ และทำให้ง่ายในการแก้ไขโปรแกรมในภายหลัง แต่ เมธอด ในอีกคำหนึ่งที่รู้จัก คือ พังก์ชัน แต่คำว่าเมธอดใช้สำหรับพังก์ชันที่อยู่ในคลาส (การเขียนโปรแกรมเชิงวัตถุ) [14] องค์ประกอบพื้นฐานที่สำคัญตามรูปแบบของ C# ที่สำคัญมีดังนี้

1) Block { ... }

ใช้รูปแบบของภาษา เช่นเดียวกับภาษา C/C++ คือ จะใช้บล็อก { ... } ในการกำหนดจุดเริ่มต้น และ จุดสิ้นสุดของการทำงานในแต่ละส่วน ซึ่งภายในบล็อกอาจมีบล็อกอยู่อีก ๆ ข้อนลงไปได้อีกตามลักษณะของงาน เช่น

```
if (.....)
{
    for (.....)
    {
        .....
    }
    .....
}
```

ทั้งนี้เครื่องหมาย “ { ” (open brace) กับ “ } ” (close brace) นั้นต้องครบถ้วนพอดี และ โดยทั่วไปเรานิยมวาง “ { ” ไว้ให้ตรงกับคำสั่งเริ่มต้นบล็อก เช่น จากรหัสคำสั่งตัวอย่างที่เรา妄ไว้ตรงกับคำสั่ง if หรือ for เป็นต้น ทั้งนี้เพื่อให้พิจารณาขอบเขตของบล็อกได้ง่าย แต่ไม่ใช่กฎข้อบังคับแต่อย่างใด สามารถเลือกวิธีในแบบที่คนดักกีดี

2) เครื่องหมายสิ้นสุดคำสั่ง (;)

ใน C# จะใช้เครื่องหมายเซมิโคลอน (semicolon) “ ; ” เป็นตัวแสดงจุดสิ้นสุดของแต่ละคำสั่ง หากไม่ใส่เครื่องหมายนี้เพื่อคุณระหว่างคำสั่ง โปรแกรมจะถือว่าเป็นคำสั่งเดียวกันไปตลอดแม้ว่าจะอยู่คนละบรรทัด ก็ตาม เช่น จากตัวอย่างมีคำสั่งทั้งหมด 3 คำสั่ง

```
x = 10;
y = "xxx";
z = x + 10;
```

เมื่อเครื่องหมาย ; เป็นตัวบ่งบอกจุดสิ้นสุดของคำสั่ง จึงสามารถมีคำสั่งอยู่ในบรรทัดเดียวกันมากกว่า 1 คำสั่งก็ได้ เช่น จากตัวอย่างมีคำสั่งทั้งหมด 3 คำสั่ง

```
x = 10; y = "xxx"; z = x + 10;
```

การเขียนรหัสคำสั่งในลักษณะนี้ ไม่ค่อนนิยมทำกัน เนื่องจากจะทำให้เราโปรแกรมได้ยาก รหัสโปรแกรมดูไม่เป็นระเบียบ แต่ก็อาจนำมาใช้ในบางกรณีได้ เช่นกัน

บางคำสั่งไม่สามารถทำให้จบภายในบรรทัดเดียวได้ เพราะบางคำสั่งจะมีคำสั่งย่ออยู่ อุปกรณ์ภายในอีก ก็ได้ ดังนั้นคำสั่งเหล่านี้จะมีบล็อกของตนเอง เช่น คำสั่ง if, คำสั่ง for และ คำสั่ง while เป็นต้น เมื่อคำสั่งใดมี

บล็อกอยู่แล้วก็ไม่จำเป็นต้องใส่เครื่องหมาย ; ต่อท้ายเครื่องหมายบล็อกอีก แต่คำสั่งภายในต้องมีเครื่องหมาย ;
ตามปกติ เช่น

```
if (เงื่อนไข)
```

```
{
```

```
    คำสั่ง A;  
    คำสั่ง B;
```

```
}
```

3) การเขียนคำอธิบายประกอบแทรกไว้ในรหัสโปรแกรม

คำอธิบาย (Comment) หมายถึง การเขียนข้อความใดๆ ที่ไม่ใช่คำสั่งประปันกันไปกับ คำสั่งอื่นๆ เพื่อ
อธิบายเรื่องใดเรื่องหนึ่งเอาไว้เพื่อความเข้าใจในรหัสโปรแกรมตรงส่วนนั้น ดังนั้นเพื่อให้โปรแกรมไม่สับสนว่า
ส่วนใดเป็นคำสั่ง ส่วนใดเป็นเพียงคำอธิบายไม่เกี่ยวข้องกับการทำงานของโปรแกรม จึงต้องใช้เครื่องหมายมา
เป็นตัวช่วยในการแยกแยะ การแทรกคำอธิบายสามารถทำได้ 2 ลักษณะดังนี้ คือ

รูปแบบ // คำอธิบาย

ใช้สำหรับคำอธิบายแบบบรรทัดเดียว (Single Line comment) โดยโปรแกรมจะถือว่าตั้งแต่
สัญลักษณ์ // เป็นต้นไปจนถึงสิ้นสุดบรรทัดเป็นคำอธิบายทั้งหมด จะไม่นำมาพิจารณาในการประมวลผล เช่น
// สูตรคำนวนหาขนาดพื้นที่สี่เหลี่ยมผืนผ้า

```
RectangleArea=width*length;
```

หรือ

```
RectangleArea=width*length; // สูตรคำนวนหาขนาดพื้นที่สี่เหลี่ยมผืนผ้า
```

รูปแบบ /* คำอธิบาย */

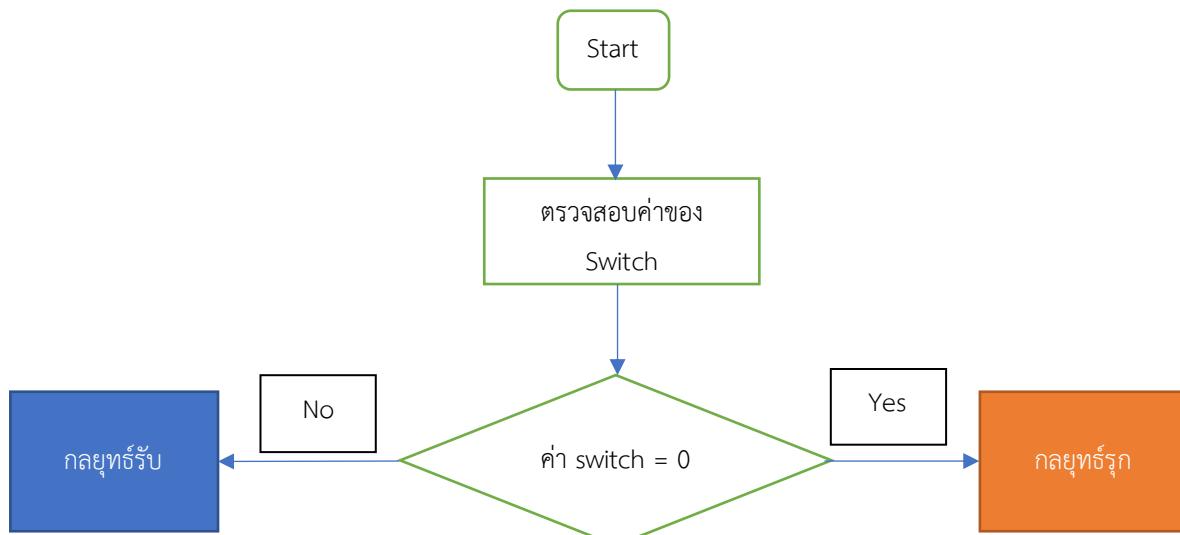
กรณีที่คำอธิบายของเราค่อนข้างยาวจำเป็นต้องเขียนหลายบรรทัด การใช้ // หลายๆ ครั้งอาจไม่
สะดวกนัก เรายังสามารถใช้ /*...*/ ครอบคำอธิบายนั้นแทน โดยโปรแกรมจะถือว่าตั้งแต่สัญลักษณ์ /* เป็นต้น
ไปจะเป็นคำอธิบายทั้งหมดจนกว่าจะเจอสัญลักษณ์ */ จึงจะถือว่าเป็นการสิ้นสุดคำอธิบาย เช่น /* นี่คือส่วน
ของคำอธิบายที่ไม่มีผลต่อการทำงานของโปรแกรมไว้เพื่อช่วยให้ทำความเข้าใจรหัสคำสั่งได้ง่ายขึ้น */ [15]

บทที่ 3

การออกแบบและการจัดทำโครงงาน

3.1 การออกแบบการทำงานของกลยุทธ์โดยรวม

ในงานวิจัยนี้ทางผู้จัดได้แนวคิดในการแก้ไขปัญหาแบ่งออกเป็น 2 กลยุทธ์ คือกลยุทธ์รุกและกลยุทธ์รับ ซึ่งสามารถแสดงแผนภาพการทำงานได้ดังรูปที่ 12 ซึ่งทางผู้จัดทำได้ออกแบบให้หุ่นยนต์มีสวิตช์ 1 ตัวที่อยู่ทำหน้าที่สัลักกลยุทธ์ไปมา โดยกำหนดให้กลยุทธ์รุกนั้นมีค่าของ Switch เท่ากับ 0 และกลยุทธ์รับนั้น มีค่า Switch เท่ากับ 1 ก่อนเริ่มทำการแข่งขันทางผู้จัดทำจะตรวจสอบค่าของ Switch ก่อนเสมอ



รูปที่ 12 Flow Chart กลยุทธ์รุกและกลยุทธ์รับ

3.2 การออกแบบการทำงานของกลยุทธ์รุก

ทางผู้จัดทำได้วางแผนการทำงานไว้ดังรูปที่ 13 โดยมีหลักการทำงานดังต่อไปนี้ เมื่อทางผู้จัดทำมั่นใจแล้วว่าทีมเป็นฝ่ายรุกทางผู้จัดทำจะทำการกด Switch เพื่อปรับเป็นกลยุทธ์รุก โดยจะเริ่มให้หุ่นยนต์ตรวจสอบค่าของสนามกล่าวคือ ให้หุ่นยนต์ตรวจสอบเซ็นเซอร์ตรวจสอบทั้ง 3 ตัวที่ทางผู้จัดทำได้ติดตั้งไว้มี ด้านหน้า (Field=1) ด้านซ้าย (Field=2) และด้านขวา (Field=3) ว่ามีวัตถุเข้ามายกเลี้ยงหรือชนอะไรหรือไม่ (Field=0) จะสามารถแบ่งออกเป็น 2 กรณีดังนี้

ในกรณีที่ 1. ใช่ (ไม่พบอะไรเลย Field=0) หุ่นยนต์จะตรวจสอบค่าของ Ultrasonic กล่าวคือ ตรวจสอบว่าสิ่งกีดขวางที่อยู่ด้านหน้าด้านขวาหรือไม่ ในกรณีที่ไม่พบ (No) จะให้หุ่นยนต์วิ่งไปด้านหน้าเรื่อยๆ จนกว่าจะหุ่นยนต์จะตรวจพบ ในกรณีที่ เจอ (Yes) จะให้อ่านค่าของ Sensor ตรวจจับวัตถุ กล่าวคือให้ทำการตรวจสอบเซ็นเซอร์ทั้ง 2 ตัว โดยเริ่มจาก Sensor A กล่าวคือการตรวจสอบเซ็นเซอร์ทางซ้ายในกรณีที่ ใช่ (พบวัตถุทางซ้าย) จะให้เคลื่อนที่ไปทางซ้ายแล้วไปตรวจสอบค่าของ Ultrasonic จนกว่าจะพบว่า ไม่ แล้วจึงเคลื่อนที่ไปต่อไปข้างหน้า ในกรณีตรวจสอบเซ็นเซอร์ทางซ้ายแล้วพบว่า ไม่ (ไม่พบวัตถุทางซ้าย) ก็จะตรวจสอบ

Sensor B กล่าวคือ เช็นเซอร์ทางขวาต่อไป เมื่อเช็นเซอร์ตรวจสอบเช็นเซอร์ทางขวาแล้วพบว่า ใช่ (พบวัตถุทางขวา) จะให้เคลื่อนที่ไปทางขวาแล้วไปตรวจสอบค่าของ Ultrasonic จนกว่าจะพบว่า ไม่ แล้วจึงเคลื่อนที่ไปต่อไปข้างหน้าต่อ ในการณ์ที่ตรวจสอบเช็นเซอร์ทางขวาพบว่า ไม่ (ไม่พบวัตถุทางขวา) จะทำการหยุดแล้วจะตรวจสอบค่าของสนามตั้งแต่เริ่มต้นต่อไป

ในการณ์ที่ 2 ไม่ (พบวัตถุหรือสามารถตรวจสอบค่าได้) หุ่นยนต์จะทำการตรวจสอบค่าของสนามกล่าวคือ หุ่นยนต์ตรวจสอบเช็นเซอร์ตรวจสอบทั้ง 3 ตัวที่ทางผู้จัดทำได้ติดตั้งไว้มี ด้านหน้า ด้านซ้าย และด้านขวา โดยจะเริ่มจากการตรวจสอบสีน้ำเงิน (Field=1) ในกรณี ใช่ จะให้หุ่นยนต์อยู่หลัง 1 วินาที วนกลับ 180 องศา แล้วให้ตรวจสอบค่าของสนามใหม่ตั้งแต่เริ่มต้นอีกรั้ง ในการณ์ ไม่ หุ่นยนต์จะทำการตรวจสอบเช็นเซอร์ด้านขวาต่อไป เมื่อเช็นเซอร์ตรวจสอบทางขวา (Field=2) และพบว่า ใช่ จะให้หุ่นยนต์ทำการเคลื่อนที่ไปทางซ้ายแล้วตรวจสอบค่าของสนามใหม่ตั้งแต่เริ่มต้นอีกรั้ง ในการณ์ตรวจสอบเช็นเซอร์ทางซ้ายแล้วพบว่า ไม่ หุ่นยนต์จะทำการหยุดแล้วทำการตรวจสอบค่าสนามใหม่ตั้งแต่เริ่มอีกรั้ง

3.3 การออกแบบการทำงานของกลยุทธ์รับ

ทางผู้จัดทำได้วางแผนการทำงานไว้ดังรูปที่ 14 โดยมีหลักการทำงานดังต่อไปนี้ เมื่อทางผู้จัดทำมั่นใจแล้วว่าทีมเป็นฝ่ายรับทางผู้จัดทำจะทำการกด Switch เพื่อปรับเป็นกลยุทธ์รับโดยจะเริ่มให้หุ่นยนต์ตรวจสอบค่าของสนามกล่าวคือ ให้หุ่นยนต์ตรวจสอบเช็นเซอร์ตรวจสอบทั้ง 3 ตัวที่ทางผู้จัดทำได้ติดตั้งไว้มี ด้านหน้า (Field=1) ด้านซ้าย (Field=2) และด้านขวา (Field=3) ว่ามีวัตถุเข้ามาใกล้หรือชนอะไรหรือไม่ (Field=0) จะสามารถแบ่งออกเป็น 2 กรณีดังนี้ว่ามีวัตถุเข้ามาใกล้หรือชนอะไรหรือไม่ หากวัตถุไม่ชนกับขอบสนามด้านใดเลย (Field=0) จะสามารถแบ่งออกเป็น 2 กรณีดังนี้

ในการณ์ที่ 1. ใช่ (ไม่พบอะไรเลย) หุ่นยนต์จะตรวจสอบค่าของ Ultrasonic กล่าวคือ ตรวจสอบว่าสิ่งกีดขวางที่อยู่ด้านหน้าด้านหน้าหรือไม่ ในการณ์ที่ เจอ (Yes) จะให้หุ่นยนต์เคลื่อนที่ไปข้างหน้าเพื่อขวางฝ่ายรุก เมื่อไม่พบหุ่นยนต์ของฝ่ายรุกแล้วตรวจสอบค่าของสนามตั้งแต่เริ่มอีกรั้ง ในการณ์ ที่ไม่พบ (No) จะให้อ่านค่า Sensor ตรวจจับวัตถุ กล่าวคือหุ่นยนต์ตรวจสอบเช็นเซอร์ทั้ง 2 ตัวต่อไป โดยเริ่มจาก Sensor A กล่าวคือการตรวจสอบเช็นเซอร์ทางซ้ายในกรณีที่ ใช่ (พบวัตถุทางซ้าย) จะให้เคลื่อนที่ไปทางซ้ายเพื่อหยุดหุ่นยนต์ของฝ่ายรุกแล้วตรวจสอบค่าของ Ultrasonic อีกรั้งจนกว่าจะพบว่า ไม่แล้วจึงตรวจสอบเช็นเซอร์อีนต่อไป ในการณ์ตรวจสอบเช็นเซอร์ทางซ้ายแล้วพบว่า ไม่ (ไม่พบวัตถุทางซ้าย) ก็จะตรวจสอบ Sensor B กล่าวคือ เช็นเซอร์ทางขวาต่อไป เมื่อเช็นเซอร์ตรวจสอบเช็นเซอร์ทางขวาแล้วพบว่า ใช่ (พบวัตถุทางขวา) จะให้เคลื่อนที่ไปทางขวาเพื่อหยุดหุ่นยนต์ของฝ่ายรุกแล้วตรวจสอบค่าของ Ultrasonic อีกรั้งจนกว่าจะพบว่า ไม่ แล้วจึงตรวจสอบเช็นเซอร์อีนต่อไปในกรณีที่ตรวจสอบเช็นเซอร์ทางขวาพบว่า ไม่ (ไม่พบวัตถุทางขวา) จะใช้ฟังก์ชันคันหา จะเกินขึ้นในกรณีที่ไม่เจออะไรเลยโดยจะให้หุ่นยนต์เคลื่อนที่ไปขวาสุดก่อน ถ้าเจออะไรก็ให้หยุด แต่เมื่อไม่เจอ

อะไรแล้วก็ให้เคลื่อนที่ต่อไปจนสุดขอบสนาม ให้หุ่นยนต์อยู่หลังกลับ 1 วินาที วนกลับ 180 องศา และให้เคลื่อนที่ต่อไปทางซ้ายสุดถ้าเจอะอะไรก็ให้หยุด แต่เมื่อไม่เจอะอะไรแล้วก็ให้เคลื่อนที่ต่อไปจนสุดขอบสนาม

ในกรณีที่ 2 ไม่ (พบวัตถุหรือสามารถตรวจสอบค่าได้) หุ่นยนต์จะทำการตรวจสอบค่าของสนาม กล่าวคือ หุ่นยนต์ตรวจสอบเซ็นเซอร์ตรวจสอบทั้ง 3 ตัวที่ทางผู้จัดทำได้ติดตั้งไว้มี ด้านหน้า ด้านซ้าย และ ด้านขวา โดยจะเริ่มจากจากการตรวจสอบทางขวา (Field=2) แล้วพบว่า ใช่ จะให้หุ่นยนต์ทำการเคลื่อนที่ไปทางซ้ายแล้วตรวจสอบค่าของสนามใหม่ตั้งแต่เริ่มต้นอีกรั้ง ในกรณีตรวจสอบเซ็นเซอร์ทางขวาแล้วพบว่า ไม่ หุ่นยนต์จะทำการตรวจสอบเซ็นเซอร์ด้านซ้ายต่อไป เมื่อเซ็นเซอร์ตรวจสอบทางซ้าย (Field=3) แล้วพบว่า ใช่ จะให้หุ่นยนต์ทำการเคลื่อนที่ไปทางขวาแล้วตรวจสอบค่าของสนามใหม่ตั้งแต่เริ่มต้นอีกรั้ง ในกรณีตรวจสอบเซ็นเซอร์ทางซ้ายแล้วพบว่า ไม่ หุ่นยนต์จะทำการหยุดแล้วทำการตรวจสอบค่าสนามใหม่ตั้งแต่เริ่มอีกรั้ง

3.4 การออกแบบการทำงานของวงจร

ทางผู้จัดทำได้ทำการออกแบบวงจรเพื่อเชื่อมต่ออุปกรณ์ต่าง ๆ ภายในหุ่นยนต์ดังรูปที่ 15 สามารถ อธิบายหลักการทำงานได้ดังนี้ จาก ถ่านชาร์จ Li-ion 18650 สามารถให้แรงดันที่ 3.7-4.2V ซึ่งไม่เพียงพอ กับ ความต้องการของไมโครคอนโทรลเลอร์ (บอร์ด Arduino) ที่ทางผู้จัดทำเลือกใช้จึงต้องทำการต่อวงจร DC/DC Step-up Converter เพื่อปรับแรงดันให้เป็น 5V เมื่อไมโครคอนโทรลเลอร์ (บอร์ด Arduino) มี แรงดันเพียงพอแล้วจะทำหน้าที่เป็นตัวจ่ายแรงดันไฟฟ้า (Output) ไปยัง มอเตอร์ เซ็นเซอร์ต่าง ๆ ต่อไป โดย มีรายละเอียดในการจ่ายแรงดันดังนี้

1. ขา D2-D5 ที่ต่ออยู่กับไมโครคอนโทรลเลอร์ (บอร์ด Arduino) จะเป็นตัวจ่ายแรงดันไฟฟ้า (Output) จะเข้าไปขา IN1-IN4 ที่เป็นตัวรับแรงดันไฟฟ้า (Input) และจะส่งแรงดันไฟฟ้าผ่าน D/C Motor Driver Module ที่อยู่ทำหน้าที่แปลงค่าความต้านทาน (R) แรงดันไฟฟ้า (V) และกระแสไฟฟ้า (I) ให้ เหมาะสมกับมอเตอร์ A และ มอเตอร์ B

2. ขา D8-D9 ต่ออยู่กับไมโครคอนโทรลเลอร์ (บอร์ด Arduino) จะเป็นตัวรับแรงดันไฟฟ้า (Input) จะ ทำหน้าที่รับค่าจากการตรวจจับว่าพวตถุหรือไม่จากเซ็นเซอร์ A และ B

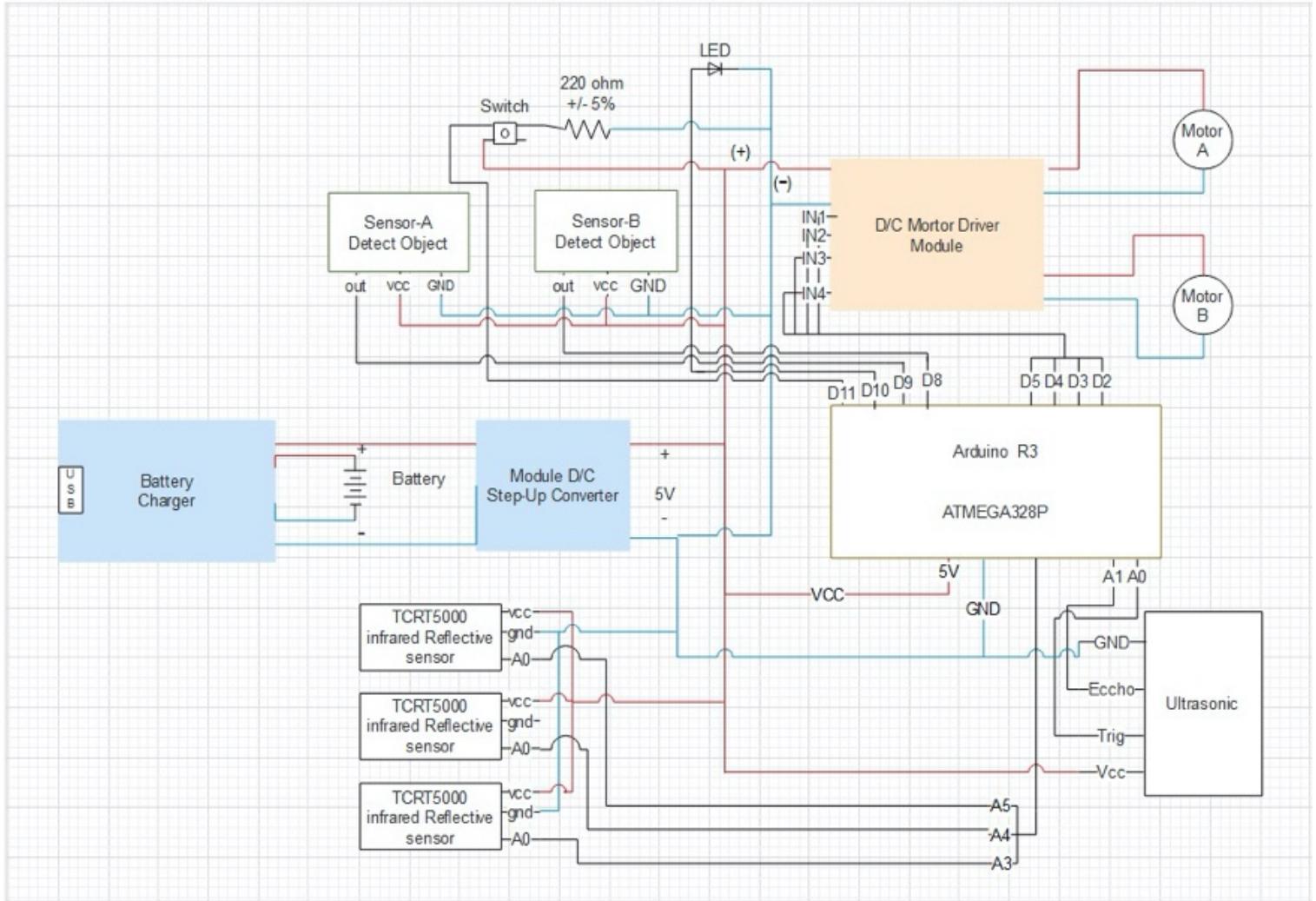
3. ขา D11 เป็นตัวรับแรงดันไฟฟ้า (Input) ที่รับค่ามาจาก Switch และเมื่อแสงเกิดขึ้น ถ้าขา D11 อ่านได้ว่ามีกระแสไฟฟ้า จะส่งให้ ขา D10 ที่เป็นตัวจ่ายแรงดันไฟฟ้า (Output) และ LED ก็จะสว่างขึ้น ซึ่งทาง ผู้จัดทำได้ทำมีความประสงค์ที่จะออกแบบจรนีขึ้นเพื่อให้รู้ว่า เมื่อหลอดไฟ LED ติด จะหมายถึงหุ่นยนต์ได้ ปรับเป็นกลยุทธ์รับเรียบร้อยแล้ว

4. ขา A0-A1 จะเป็นทั้งตัวรับและตัวจ่ายแรงดันไฟฟ้า (Input & Output) โดยจะสั่งให้ทำงานตาม ไลบรารีของ Ultrasonic (ตัวมันเอง) ตามที่ผู้พัฒนาหรือผู้ผลิตได้กำหนดเอาไว้ ซึ่งทางผู้จัดทำไม่แน่ใจว่าได้ เป็นขาส่งขาได้เป็นขารับ

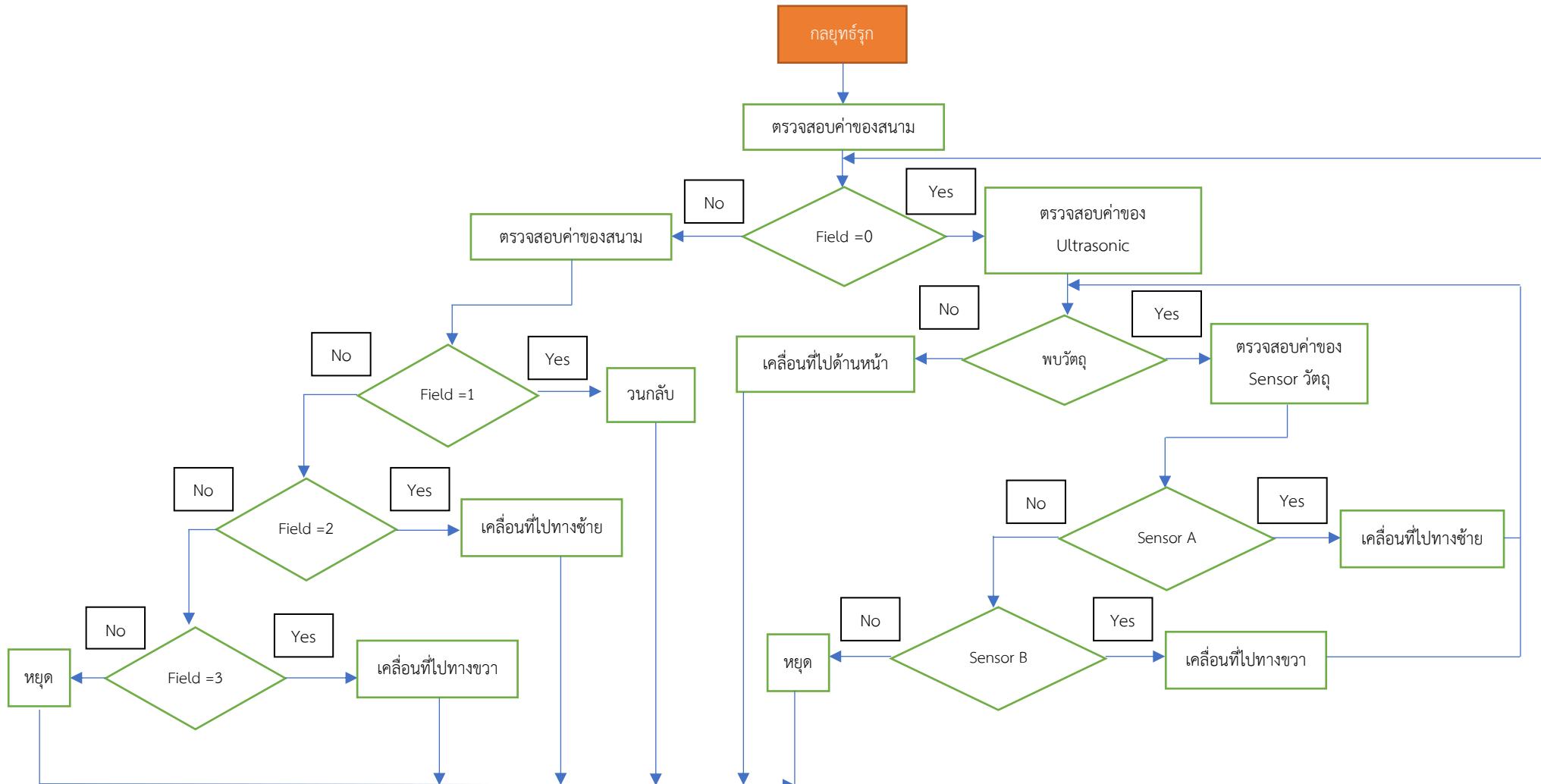
5. ขา A3 จะเป็นตัวรับค่าแรงดันไฟฟ้า (Input) โดยจะทำหน้าที่ตรวจค่าสีของสนามที่เป็นสีน้ำเงิน

6. ขา A4-A5 จะเป็นตัวรับค่าแรงดันไฟฟ้า (Input) โดยจะทำหน้าที่ตรวจค่าที่จับได้ทางซ้ายและขวา

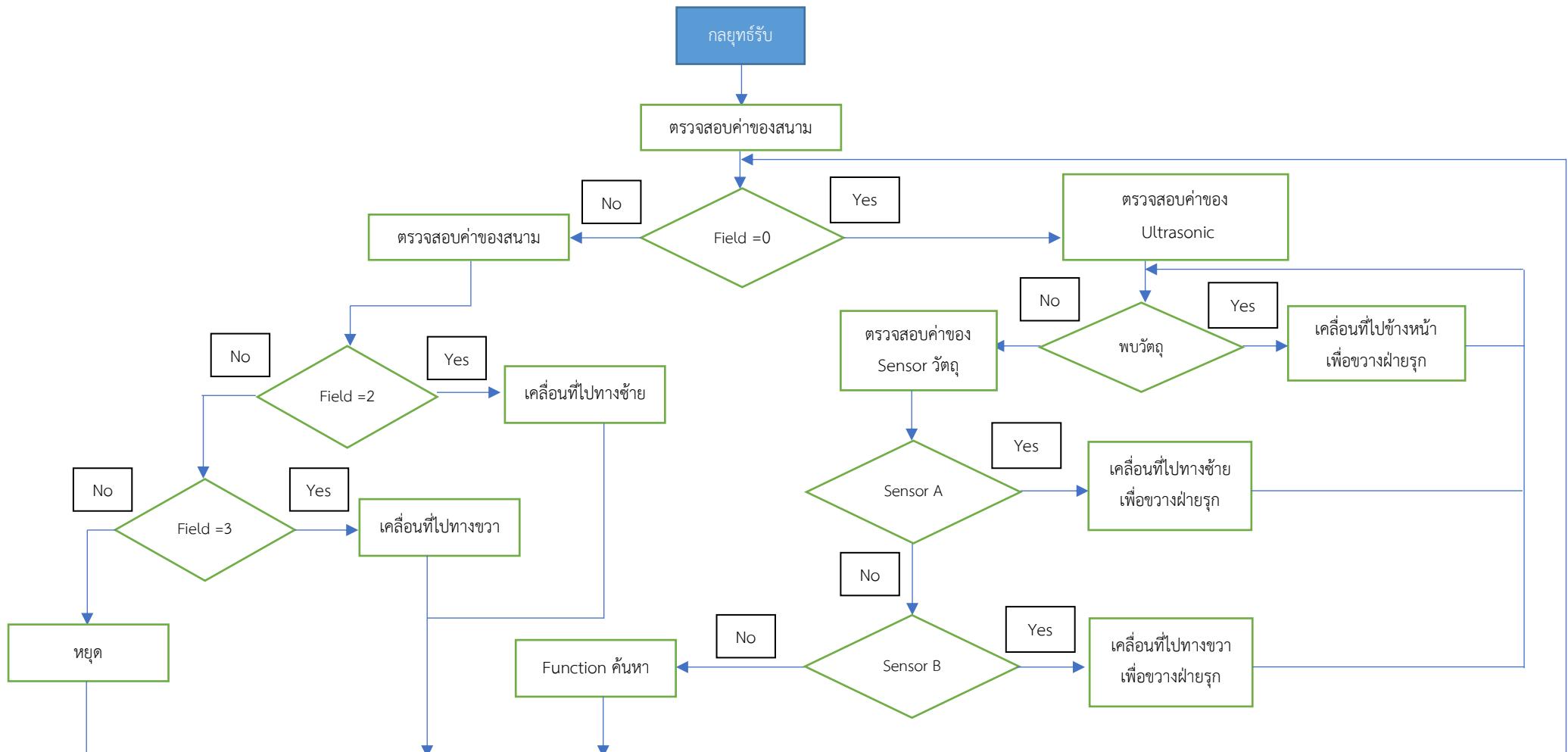
หมายเหตุ A0 ในเซ็นเซอร์ TCRT5000 เป็นตัวจ่ายแรงดันไฟฟ้า (Output) ที่ออกจากการตัวเซ็นเซอร์เองซึ่งค่าที่ออกมามี 2 แบบคือ A0 และ D0 ทางผู้จัดทำได้เลือกค่าที่เป็น A0 เพราะว่าต้องการข้อมูลที่ได้เป็นแบบ Analog (A) มีความหลากหลายในการใช้งานไม่ใช่แค่เพียงค่า 0 กับ 1 แบบ Digital (D)



รูปที่ 15 การออกแบบวงจรเพื่อเข้มต่ออุปกรณ์ต่าง ๆ ภายในหุ่นยนต์



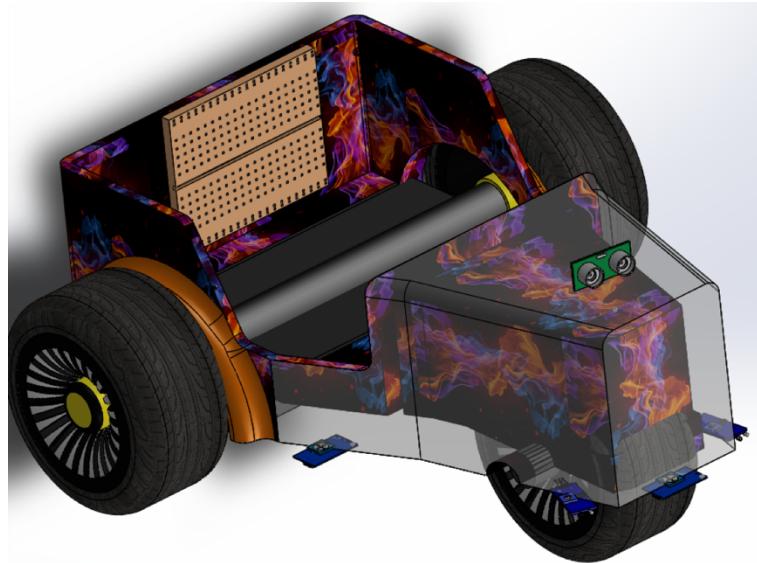
รูปที่ 13 Flow Chart กลยุทธ์รุก



รูปที่ 14 Flow Chart กลยุทธ์รับ

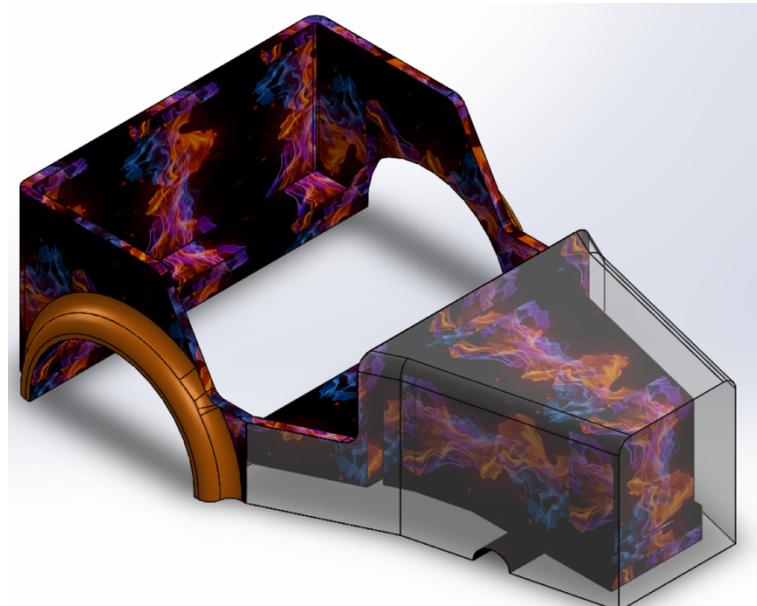
3.5 การออกแบบหุ่นยนต์

ทางผู้จัดทำได้ทำการออกแบบหุ่นยนต์มีโครงสร้างทั้งหมดมีความกว้าง 8.8 เซนติเมตร ความยาว 9.1 เซนติเมตรและความสูง 4.6 เซนติเมตรสามารถแสดงได้ดังรูปที่ 15 โดยทางผู้จัดทำได้ทำการแบ่งส่วนประกอบของหุ่นยนต์ออกเป็น 4 ส่วนประกอบใหญ่ ๆ ดังนี้

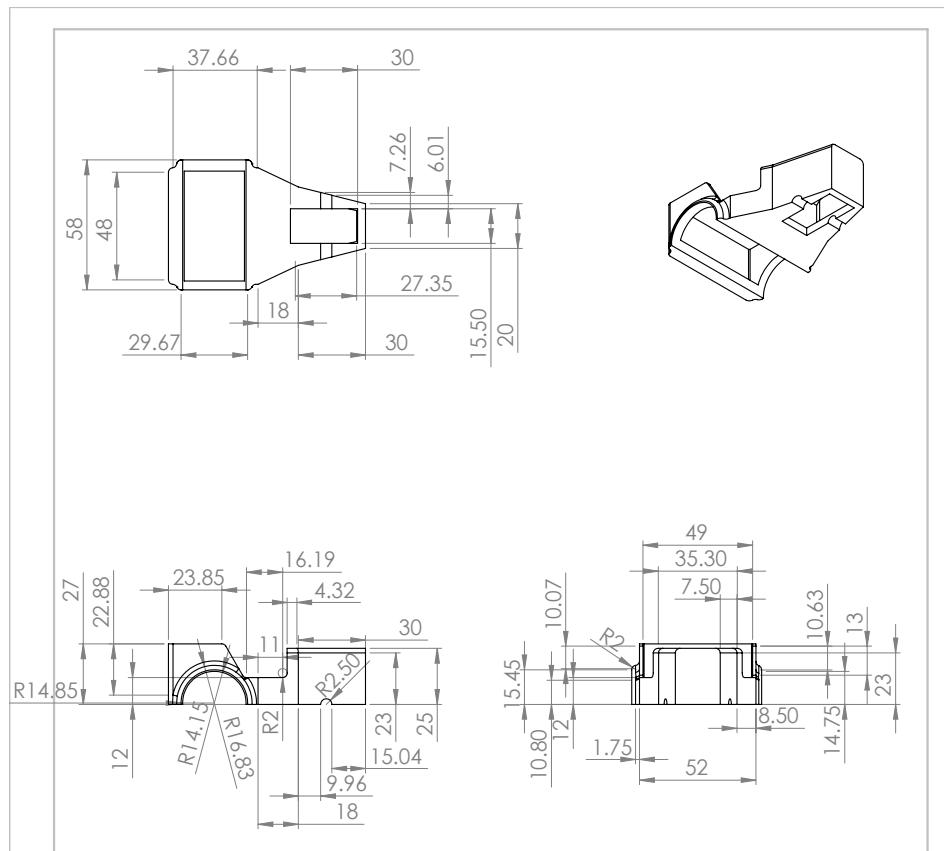


รูปที่ 15 โครงสร้างทั้งหมดของหุ่นยนต์

1. การออกแบบโครงหุ่นยนต์หลักที่มีความกว้าง 5.8 เซนติเมตร ความยาว 8.9 เซนติเมตรและความสูง 2.7 เซนติเมตร สามารถแสดงได้ดังรูปที่ 16 และสามารถอธิบายรายละเอียดของส่วนประกอบโครงหุ่นยนต์หลักได้ดังรูปที่ 17



รูปที่ 16 โครงหุ่นยนต์หลัก

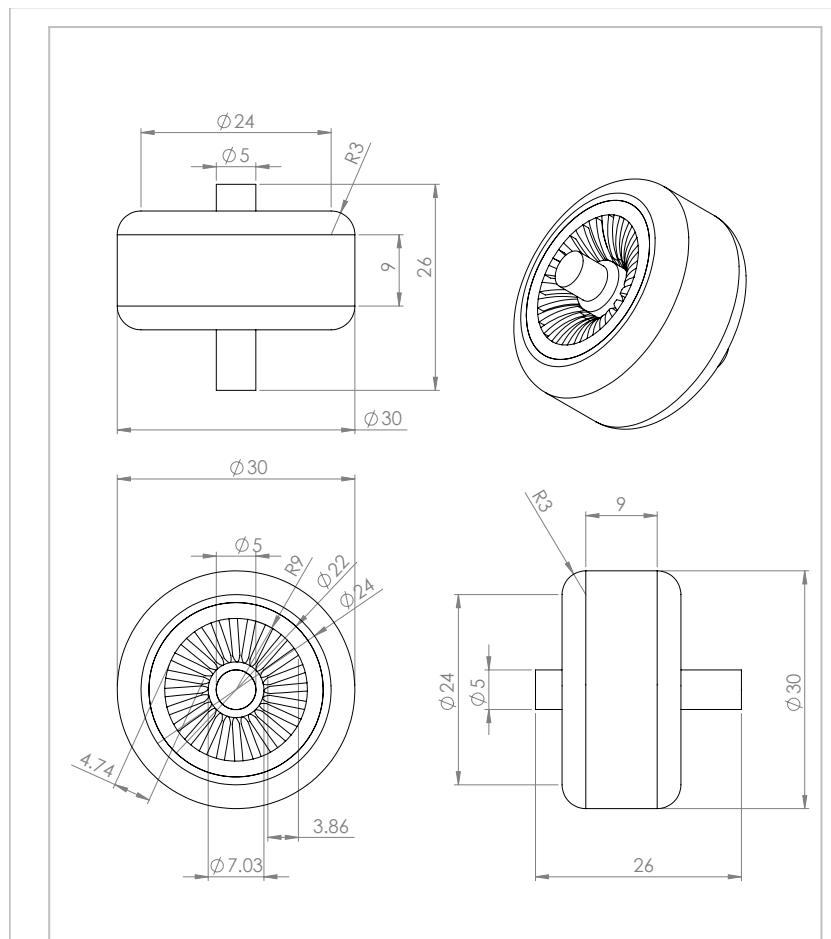


รูปที่ 17 รายละเอียดของโครงหุ้นยนต์หลัก

2. การออกแบบล้อหน้าและแกนล้อหน้าที่มีเส้นผ่านศูนย์กลางของล้อหน้า 3 เซนติเมตร ความยาวของแกนอยู่ที่ 2.6 เซนติเมตร เส้นผ่านศูนย์กลางของแกน 0.5 เซนติเมตร และเส้นผ่านศูนย์กลางของล้อหน้าข้างใน 2.4 เซนติเมตร สามารถแสดงได้ดังรูปที่ 18 และสามารถอธิบายรายละเอียดของล้อหน้าและแกนล้อหน้าได้ดังรูปที่ 19

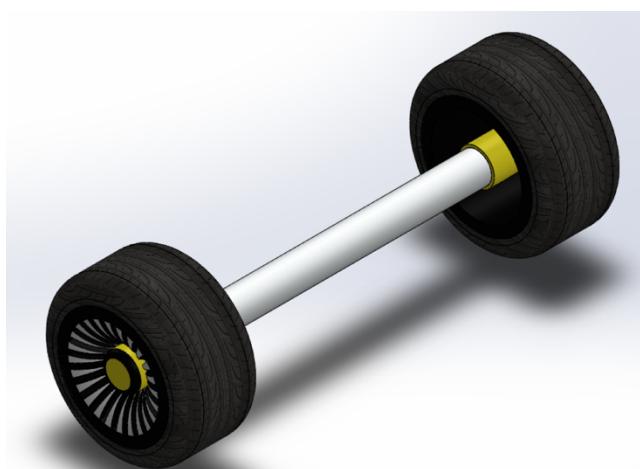


รูปที่ 18 ล้อหน้าและแกนล้อหน้า

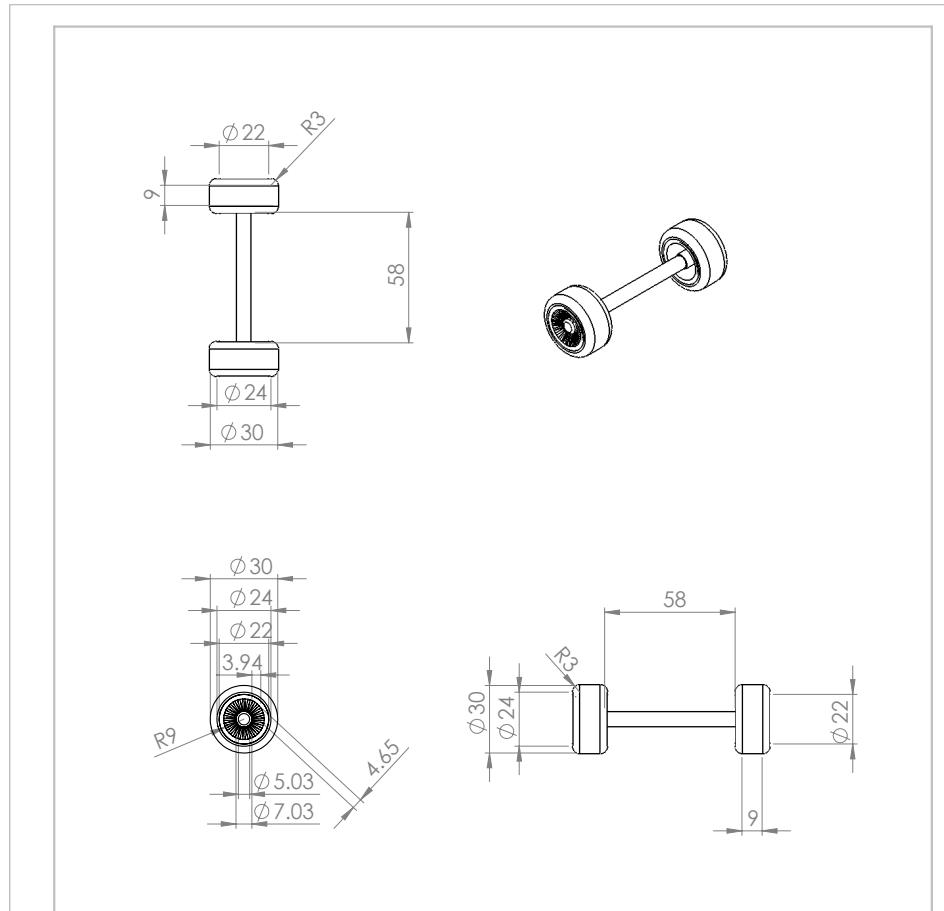


รูปที่ 19 รายละเอียดล้อหน้าและแกนล้อหน้า

3. การออกแบบล้อหลังและแกนล้อหลังที่มีเส้นผ่านศูนย์กลางของล้อหลัง 3 เซนติเมตร ความยาวของแกนอยู่ที่ 7.6 เซนติเมตร เส้นผ่านศูนย์กลางของแกนตรงกลาง 0.7 เซนติเมตร เส้นผ่านศูนย์กลางของแกนที่เสียบไปในล้อ 0.5 เซนติเมตร เส้นผ่านศูนย์กลางของล้อหลังข้างใน 2.4 เซนติเมตรและแกนระหว่างล้ออยู่ 5.8 เซนติเมตร สามารถแสดงได้ดังรูปที่ 20 และสามารถอธิบายรายละเอียดของล้อหลังและแกนล้อหลังได้ดังรูปที่ 21

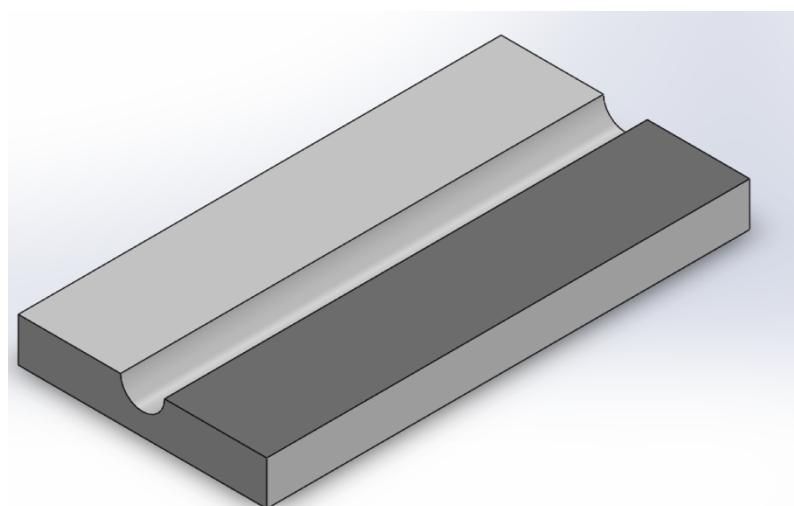


รูปที่ 20 ล้อหลังและแกนล้อหลัง

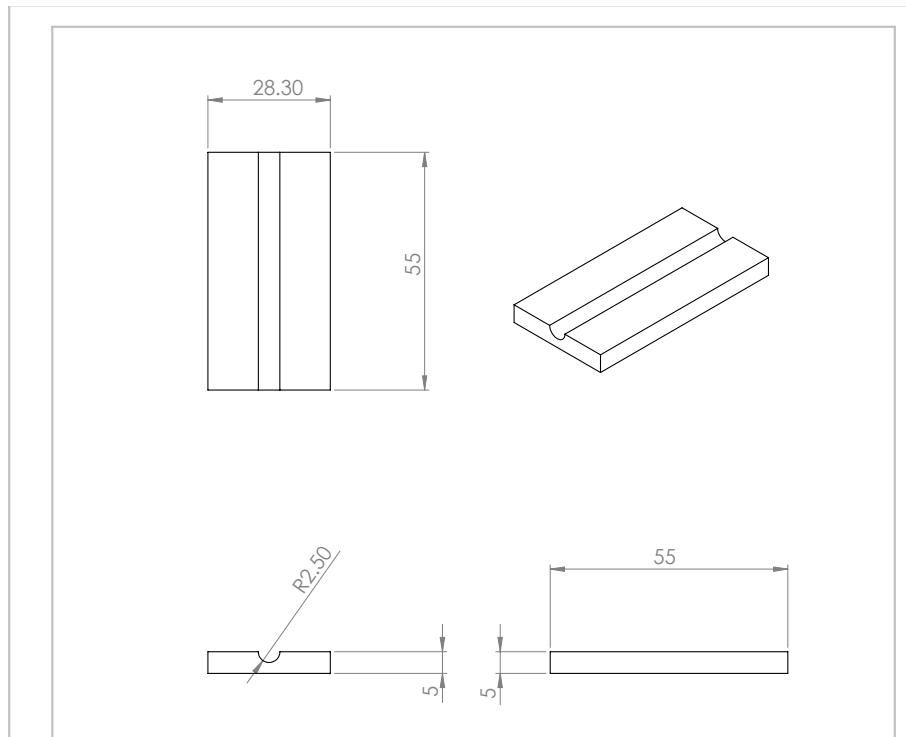


รูปที่ 21 รายละเอียดล้อหลังและแกนล้อหลัง

4. ฐานล้อจะติดตั้งบริเวณล้อและแกนล้อด้านหลังที่มีความกว้าง 2.83 เซนติเมตร ความยาว 5.5 เซนติเมตรและความสูง 0.5 เซนติเมตร สามารถแสดงได้ดังรูปที่ 22 และสามารถอธิบายรายละเอียดของฐานล้อได้ดังรูปที่ 23



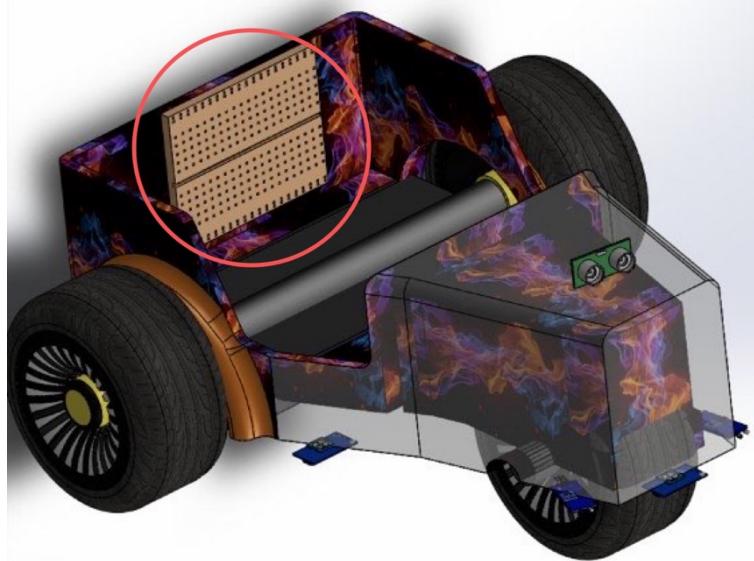
รูปที่ 22 ฐานล้อ



รูปที่ 23 รายละเอียดฐานล้อ

5. อุปกรณ์ที่ใช้ทั้งหมดภายในหุ่นยนต์มีดังนี้

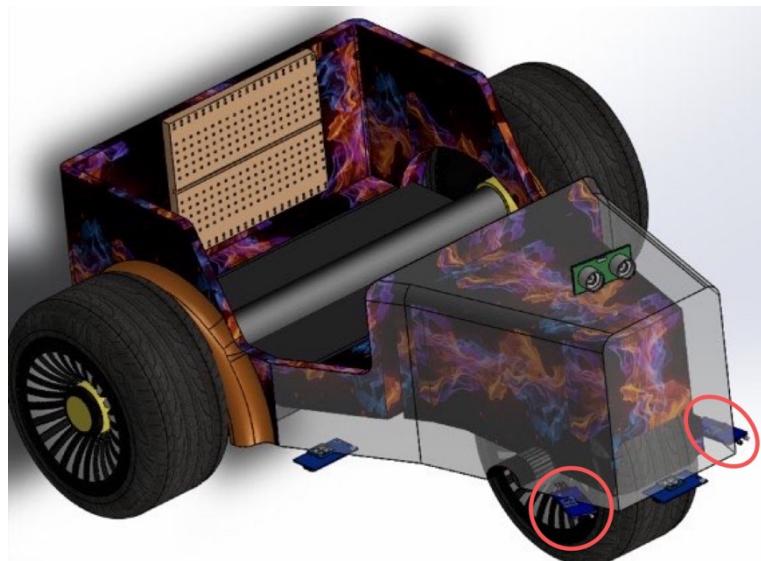
5.1 เบสบอร์ด 400 ช่องเสียบ ใช้จำนวน 2 ชิ้นติดตั้งบริเวณด้านท้ายของรถ ดังรูปที่ 24



รูปที่ 24 เบสบอร์ด ที่ติดตั้งกับหุ่นยนต์

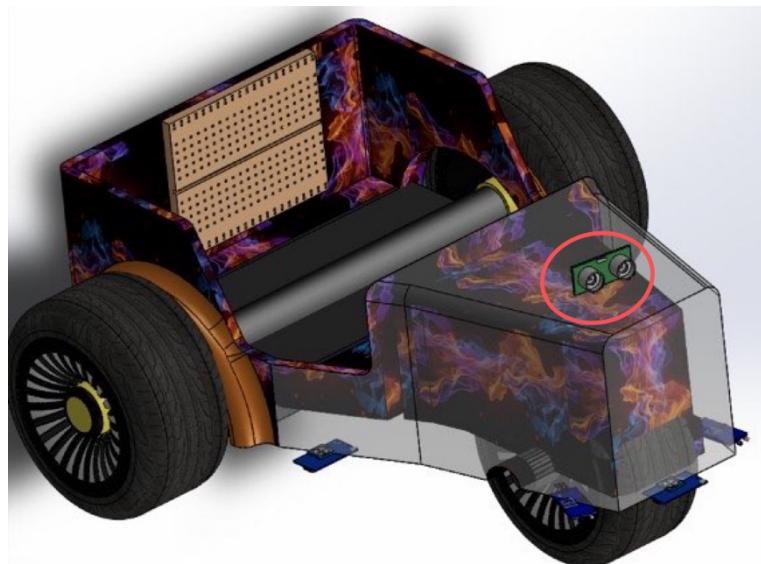
5.2 ไมโครคอนโทรลเลอร์ รุ่น Arduino R3 ATMega 328p U ที่ทำให้ควบคุมการทำงานของหุ่นยนต์ทั้งหมดติดตั้งบริเวณเบสบอร์ด

5.3 IR Infrared Obstacle Avoidance Sensor Module ใช้จำนวน 2 ชิ้น ติดตั้งบริเวณหน้ารถทางซ้ายและทางขวา ดังรูปที่ 25



รูปที่ 25 IR Infrared Obstacle Avoidance Sensor Module ที่ติดตั้งกับหุ่นยนต์

5.4 Ultrasonic Module ใช้จำนวน 1 ชิ้น ติดตั้งบริเวณด้านบน ดังรูปที่ 26

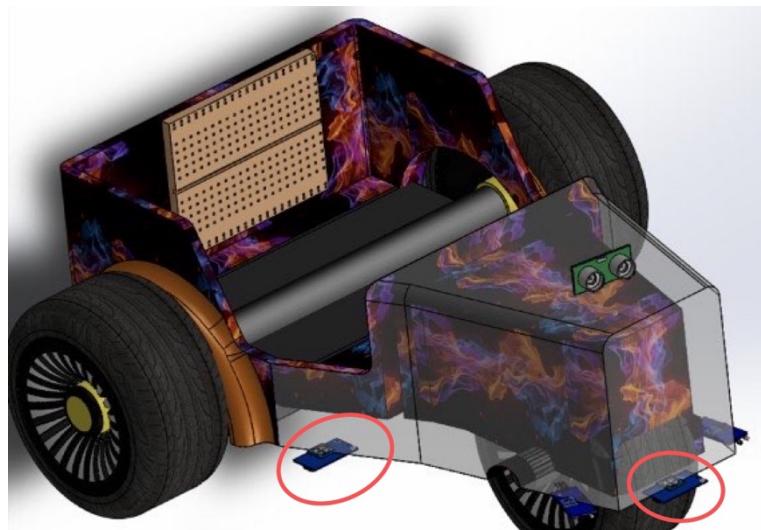


ที่ 26 Ultrasonic Module ที่ติดตั้งกับหุ่นยนต์

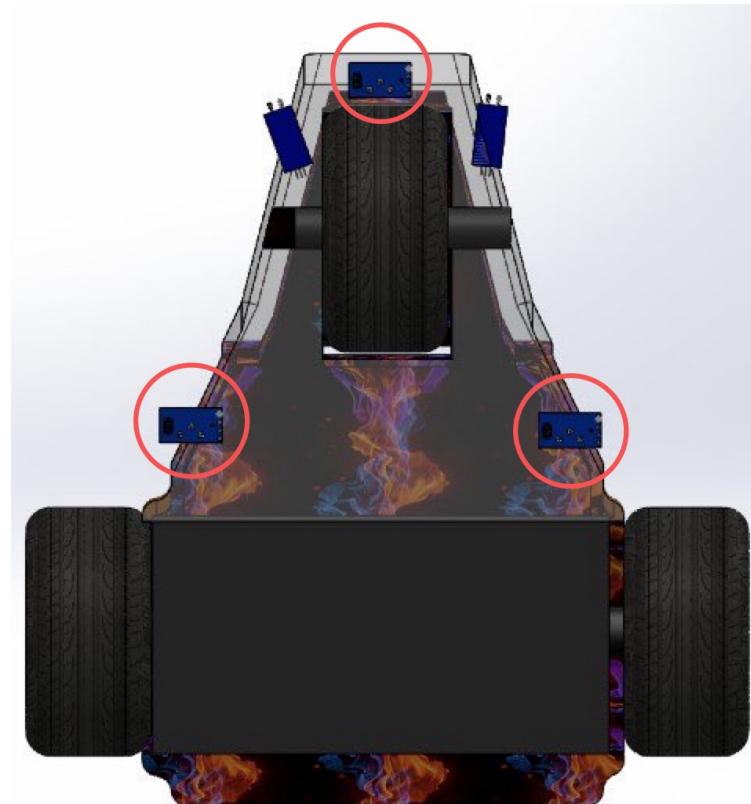
5.5 L298N Dual H-Bridge Motor Controller ที่ทำหน้าที่ควบคุมการทำงานของ Motor ใช้จำนวน 1 ชิ้นติดตั้งบริเวณแบบอิริวาร์

5.6 Motor ใช้จำนวน 2 ชิ้น ติดตั้งบริเวณล้อด้านหลังทั้ง 2 ข้าง

5.7 TCRT5000 Infrared Reflective sensor ใช้ทั้งหมด 3 ชิ้น ติดทั้งบริเวณด้านหน้าทางซ้ายและทางขวา ดังรูปที่ 27 และรูปที่ 28



ที่ 27 TCRT5000 Infrared Reflective sensor ที่ติดตั้งกับหุ่นยนต์ทางด้านหน้าและทางด้านขวา



ที่ 28 TCRT5000 Infrared Reflective sensor ที่ติดตั้งกับหุ่นยนต์ทางด้านหน้า ทางด้านขวาและทางซ้าย

5.8 วัตต์ DC/DC Step-up ใช้จำนวน 1 ชิ้น ติดตั้งบริเวณเบสเบอร์ดเพื่อปรับแรงดันให้ล่างถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V ให้มีขนาด 5V

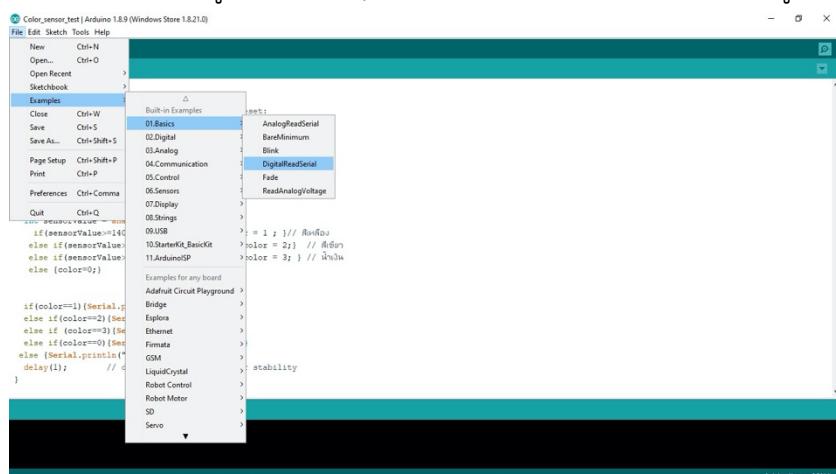
บทที่ 4

การทดสอบอุปกรณ์

4.1 การทดสอบ Sensor วัดระยะทางแบบ digital

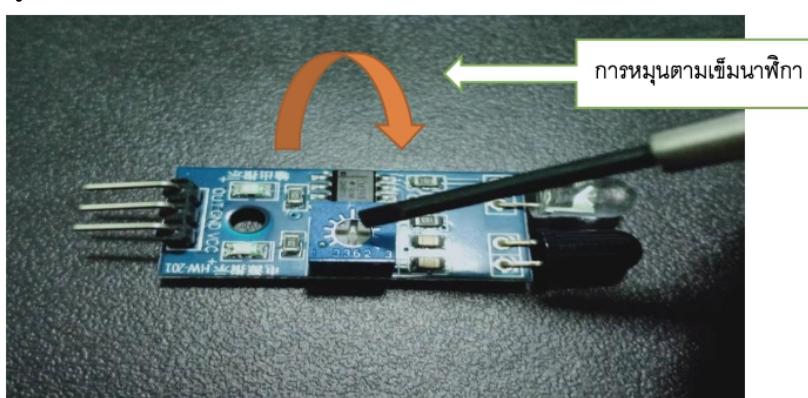
(IR Infrared Obstacle Avoidance Sensor Module)

ทางผู้จัดทำได้นำมาใช้ในการตรวจจับสิ่งกีดขวาง ซึ่งได้ทำการทดสอบการตรวจจับสิ่งกีดขวางของ Sensor ด้วยการทดลองอ่านค่าที่ได้จาก Sensor ผ่านทาง Serial Monitor และวัดระยะที่เกิดขึ้นจริง เปรียบเทียบกับค่าที่ Sensor นั้นสามารถตรวจจับสิ่งกีดขวางได้จากโปรแกรม Arduino ซึ่งสามารถใช้โปรแกรม DigitalReadSerial ที่อยู่ในตัว Example ของโปรแกรม Arduino ได้เลยดังรูปที่ 29



รูปที่ 29 โปรแกรม DigitalReadSerial ที่อยู่ในตัว Example ของโปรแกรม Arduino

นอกจากนี้ตัว Sensor ของ IR Infrared Obstacle Avoidance Sensor Module สามารถปรับระยะการตรวจจับสิ่งกีดขวางได้ โดยการหมุนที่ตัวปรับค่าการอ่านระยะของ Sensor ซึ่งการหมุนตามเข็มนาฬิกาจะทำให้ระยะการตรวจจับสิ่งกีดขวางนั้นมากขึ้น ส่วนการหมุนทวนเข็มนาฬิกาจะทำให้ระยะการตรวจจับสิ่งกีดขวางนั้นน้อยลงดังรูปที่ 30

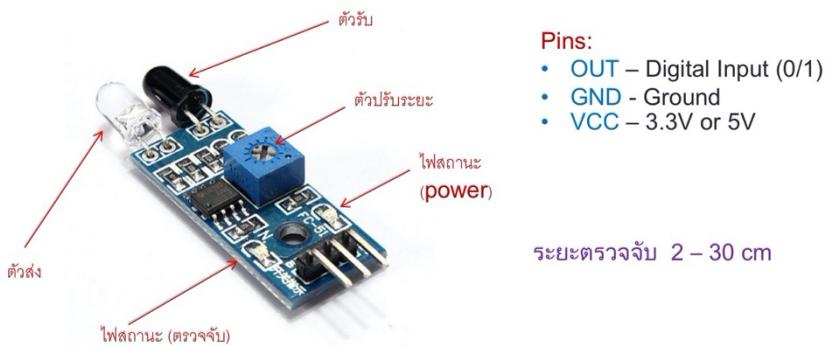


รูปที่ 30 ปรับระยะการตรวจจับสิ่งกีดขวางโดยการหมุนตามเข็มนาฬิกา

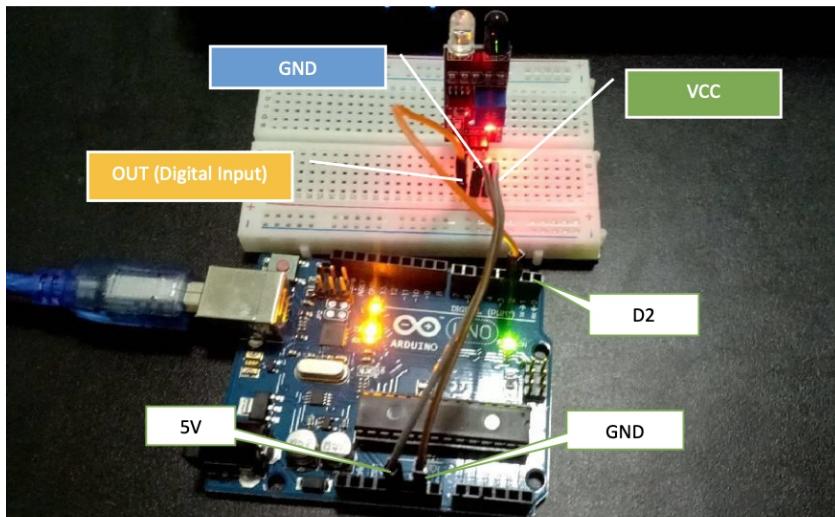
ในการตรวจสอบระยะนั้นทางผู้จัดทำต้องทำการต่อวงจรและใช้โค้ดคำสั่งดังที่ได้กล่าวไว้การต่อวงจรทดสอบนั้นก็ไม่ได้ยุ่งยากซึ่งในตัว IR Infrared Obstacle Avoidance Sensor Module นั้นจะมีขา (Pins) หลัก ๆ อยู่ 3 ตัว คือ 1. Out หรือ Digital Input ที่ให้ค่า 0 และ 1 2. GND (Ground) 3. VCC (5V) ซึ่งสามารถแสดงได้ดังรูปที่ 31 และการต่อวงจรเบื้องต้นดังรูปที่ 32

Sensor วัดระยะทางแบบ digital

- IR Infrared Obstacle Avoidance Sensor Module
สามารถใช้ในการตรวจจับสิ่งกีดกวางได้



รูปที่ 31 ส่วนประกอบต่าง ๆ ของ IR Infrared Obstacle Avoidance Sensor Module



รูปที่ 32 การต่อวงจรเบื้องต้น

หลังจากต่อวงจรเสร็จเรียบร้อยก็ทำการ Upload ตัวโปรแกรมเข้าไปใน Board Arduino แล้วทำการหมุนตัวปรับระยะพร้อมกับอ่านค่าบนหน้าจอ (Serial Monitor) บน Menu Bar ที่ซึ่งว่า Tools ในโปรแกรม Arduino ดังรูปที่ 33



DigitalReadSerial | Arduino 1.8.9 (Windows Store App) | File Edit Sketch Tools Help

DigitalReadSerial

Reads a digital input from the Arduino Uno. The sketch reads the state of the pushbutton connected to pin 2 and prints it to the Serial Monitor.

```
http://www.arduino.cc/en/Tutorial/DigitalReadSerial
```

```
// digital pins
int pushButton = 2;

// the setup routine runs once at startup, but has nothing to do
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
    // make the pushbutton's pin an input:
    pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    // read the input pin:
    int buttonState = digitalRead(pushButton);
    // print out the state of the button:
    Serial.println(buttonState);
    delay(1); // delay in between reads for stability
}
```

Sketch uses 1942 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 188 bytes (9%) of dynamic memory, leaving 1860 bytes for local variables. Maximum is 2048 bytes.

Tools

- Auto Format
- Archive Sketch
- File Encoding & Reload
- Manage Libraries...
- Serial Monitor **Ctrl+Shift+M**
- Serial Plotter **Ctrl+Shift+L**
- WIFIO101 / WIRINNA Firmware Updater

Board: "Arduino Uno" Port: "COM4 (Arduino Uno)"

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

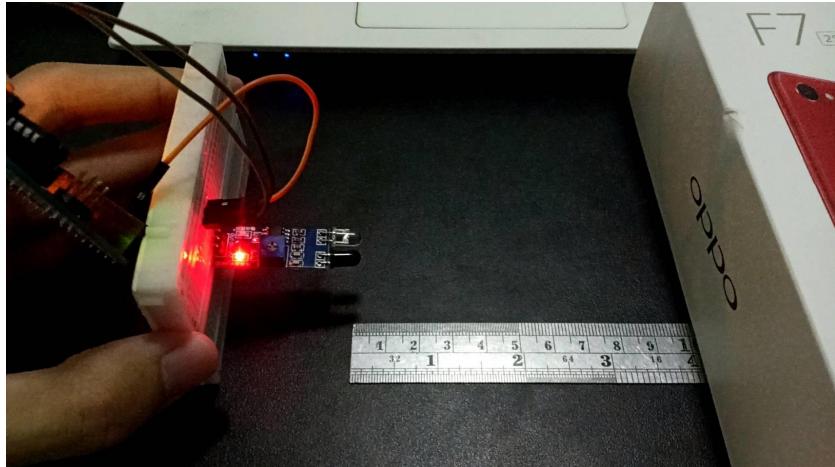
รูปที่ 32 อ่านค่าบนหน้าจอ (Serial Monitor)

สำหรับตัว IR Infrared Obstacle Avoidance Sensor นั้นหาก Sensor ตรวจไม่พบสิ่งกีดขวางจะแสดงค่าบน Serial Monitor แสดงค่าเป็น 1 หากตรวจพบสิ่งกีดขวาง Serial Monitor จะแสดงค่า เป็น 0 ดังรูปที่ 34 และรูปที่ 35

รูปที่ 32 ตรวจไม่พบสิ่งกีดขวางจะแสดงค่าบน Serial Monitor เป็น 1

รูปที่ 33 ตรวจพบสิ่งกีดขวาง Serial Monitor จะแสดงค่า เป็น 0

ซึ่งทางผู้จัดทำต้องทำการอ่านค่าบนจอของ Serial Monitor พร้อมกับทำการหมุนตัวปรับระยะของ IR Infrared Obstacle Avoidance Sensor ให้สามารถตรวจจับสิ่งกีดขวางในระยะตามที่เรานั้นต้องการได้ ซึ่งได้แสดงตัวอย่างการปรับระยะให้ Sensor สามารถตรวจจับสิ่งกีดขวางได้ภายในระยะ ที่มากกว่าหรือเท่ากับ 10 เซนติเมตรดังรูปที่ 34



รูปที่ 34 ตรวจพบสิ่งกีดขวางที่ผู้จัดทำได้ทดสอบ

4.2 การทดสอบ TCRT5000 Infrared Reflective Sensor

สำหรับตัวของ TCRT5000 Infrared Reflective Sensor นั้นจะมีขา(Pins) หลัก ๆ อยู่ 4 ตัว คือ 1. D0 หรือ Digital Input ที่ให้ค่า 0 และ 1 2. A0 หรือ Analog Input ที่ให้ค่า 0-1023 3. GND (Ground) และ 4. VCC (5V) ซึ่งสามารถแสดงดังรูปที่ 35

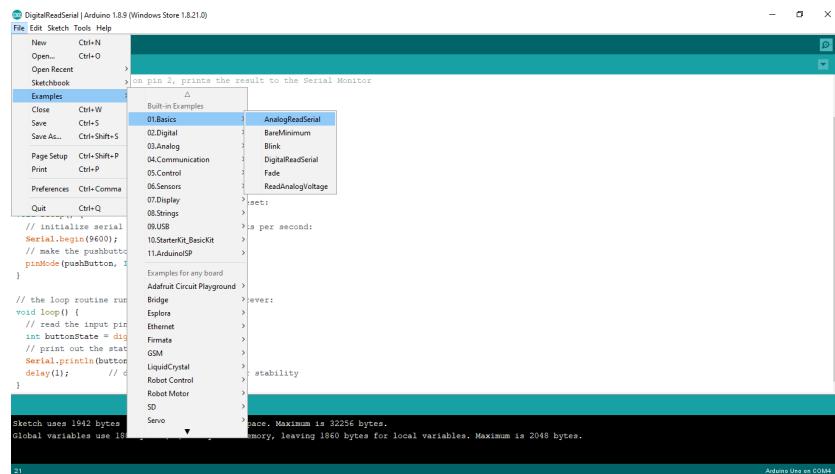
TCRT5000 Infrared Reflective sensor



รูปที่ 35 ส่วนประกอบต่าง ๆ ของ TCRT5000 Infrared Reflective Sensor

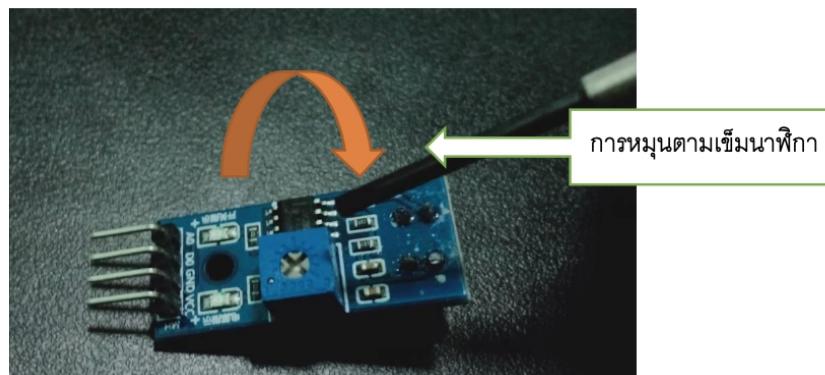
สำหรับตัว TCRT5000 Infrared Reflective Sensor ทางผู้จัดทำได้นำมาใช้ในการตรวจสอบของ ของสามารถแข่งขัน ซึ่งจะทำการทดสอบด้วยการทดลองนำสีต่าง ๆ มาให้ Sensor TCRT5000 นั้นตรวจจับ และอ่านค่าที่ได้จาก Sensor ผ่านทาง Serial Monitor โดยบันทึกค่าหรือช่วงของสีเหล่านั้นเอาไว้ สำหรับตัว โปรแกรมที่ใช้ในการตรวจค่าของ Sensor TCRT5000 ที่อ่านได้จากการนำสีต่าง ๆ มาให้ sensor นั้นตรวจจับ

ได้จากโปรแกรม Arduino ซึ่งสามารถใช้โปรแกรม AnalogRead ที่อยู่ในตัว Example ของโปรแกรม Arduino ได้เลยดังรูปที่ 36



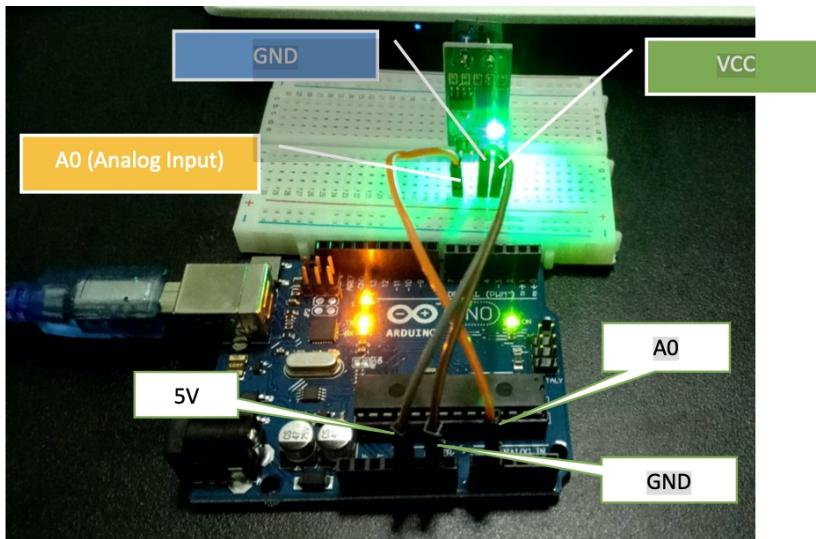
รูปที่ 36 โปรแกรม AnalogRead ที่อยู่ในตัว Example ของโปรแกรม Arduino

นอกจากนี้ตัว TCRT5000 Infrared Reflective Sensor สามารถปรับระยะเวลาการตรวจจับค่าของสีได้โดยการหมุนที่ตัวปรับค่าการอ่านระยะของ Sensor ซึ่งการหมุนตามเข็มนาฬิกาจะทำให้ระยะเวลาการตรวจจับค่าสีนั้นมากขึ้น ส่วนการหมุนทวนเข็มนาฬิกาจะทำให้ระยะเวลาการตรวจจับค่าสีนั้นน้อยลง ดังรูปที่ 37



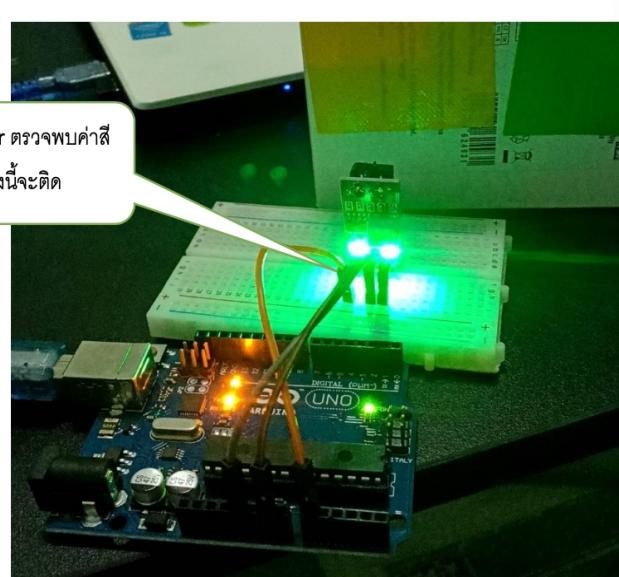
รูปที่ 37 ปรับระยะเวลาการตรวจจับค่าของสีโดยการหมุนตามเข็มนาฬิกา

ในการตรวจสอบค่าของสีนั้นเราต้องทำการต่อวงจรและใช้โค้ดคำสั่ง AnalogReadSeries จาก Example ในตัวโปรแกรมของ Arduino และต้องจดทดสอบหลังจากต่อวงจรเสร็จเรียบร้อยก่อนทำการ Upload ตัวโปรแกรมเข้าไปใน Board Arduino และทำการหมุนตัวปรับระยะเวลาและนำสีต่าง ๆ มาทดสอบและอ่านค่าบนหน้าจอ(Serial Monitor) และบันทึกช่วงของสีนั้นเอาไว้ รูปการณ์ต่อจะเป็นตัวอย่างดังรูปที่ 38



รูปที่ 38 การต่อวงจรเบื้องต้น

ตัวอย่างในการทดสอบค่าสีของ TCRT5000 Infrared Reflective Sensor คือการนำสีของที่มีสีต่าง ๆ นั่นมาให้ TCRT5000 Infrared Reflective Sensor ตรวจจับและอ่านค่าจาก Serial Monitor ของโปรแกรม Arduino ได้แสดงถึงการนำสีมาให้ Sensor นั้นตรวจจับ ดังรูปที่ 39 และค่าที่ได้จากการตรวจจับสีของ Sensor ใน Serial Monitor ดังรูปที่ 40



รูปที่ 39 แสดงถึงการนำสีมาให้ Sensor นั้นตรวจจับ

```

/*
  AnalogReadSerial | Arduino 1.8.9 (Windows Store 1.3.21.0)
  File Edit Sketch Tools Help
  AnalogReadSerial
  AnalogReadSerial
  Reads an analog input on pin 0, prints the
  graphical representation is available on
  Attach the center pin of a potentiometer
  This example code is in the public domain.
  http://www.arduino.cc/en/Tutorial/AnalogRead
  /*
  the setup routine runs once when you press
  void setup()
  // initialize serial communication at 9600 baud
  Serial.begin(9600);
  */
  // the loop routine runs over and over again forever:
  void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(A0);
    // print out the value you read:
    Serial.println(sensorValue);
    delay(1); // delay in between reads for stability
  }
  Done uploading
  Sketch uses 1860 bytes (5%) of program storage space. Maximum is 32256 bytes.
  Global variables use 188 bytes (6%) of dynamic memory, leaving 1860 bytes for local variables. Maximum is 2048 bytes.

```

รูปที่ 40 ค่าที่ได้จากการตรวจจับสีของ Sensor ใน Serial Monitor

จากตัวอย่างในการทดสอบค่าสีที่อ่านได้จาก sensor ในรูปที่ 39 และ 40 นั้น หมายความว่าเมื่อ TCRT5000 Infrared Reflective Sensor อ่านค่าได้ในช่วงระหว่าง 40-60 โดยประมาณแสดงว่าสีนั้นคือ “สีเหลือง” ซึ่งควรทำการทดลองค่าของสีที่จำเป็นต้องใช้ในการเขียนโปรแกรมหุ่นยนต์ เช่น สีของขอบสนาม อยู่ในช่วงในของ TCRT5000 Infrared Reflective Sensor และนำช่วงเหล่านั้นไปใช้ในการเขียนโปรแกรมต่อไป

บรรณานุกรม

- [1] ROBOMAC. การใช้หุ่นยนต์ในงานอุตสาหกรรม [ออนไลน์]. เข้าถึงได้จาก :
<http://www.robomac.co.th/why-robotic-automation-robotics-manufacturing.html>
(วันที่ค้นข้อมูล : 28 มีนาคม 2563)
- [2] อุตสาหกรรมสาร. วารสารของกรมส่งเสริมอุตสาหกรรม พิมพ์ปีที่ 60 ฉบับเดือนพฤษภาคม - มิถุนายน 2561 (วารสารวิชาการ) [ออนไลน์]. เข้าถึงได้จาก :
<https://www.dip.go.th/uploadcontent/Busarin/event/May-June%202561%20ROBOTICS.pdf> (วันที่ค้นข้อมูล : 28 มีนาคม 2563)
- [3] ครุฑันพงษ์ ภู่รักษ์. โครงสร้างโปรแกรมของ Arduino [ออนไลน์]. เข้าถึงได้จาก :
http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_3.pdf
(วันที่ค้นข้อมูล : 28 มีนาคม 2563)
- [4] วิกิพีเดียว ไมโครคอนโทรลเลอร์ [ออนไลน์]. เข้าถึงได้จาก : <https://th.wikipedia.org/wiki/ไมโครคอนโทรลเลอร์> (วันที่ค้นข้อมูล : 28 มีนาคม 2563)
- [5] โครงสร้างของบอร์ด Arduino Uno R3 [ออนไลน์]. เข้าถึงได้จาก :
<http://dd4toew.blogspot.com/2017/05/arduino-uno-r3.html>
(วันที่ค้นข้อมูล : 29 มีนาคม 2563)
- [6] Arduino UNO R3 [ออนไลน์]. เข้าถึงได้จาก :
<https://commanddronestore.com/products/ca001.php> (วันที่ค้นข้อมูล : 29 มีนาคม 2563)
- [7] Ken RobotSiam. การใช้งาน IR Infrared Obstacle Avoidance Sensor Module [ออนไลน์]. เข้าถึงได้จาก : <https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html> (วันที่ค้นข้อมูล : 29 มีนาคม 2563)

- [8] เซ็นเซอร์วัดระยะทาง. Ultrasonic Module HC-SR04 [ออนไลน์]. เข้าถึงได้จาก :
<https://www.arduinoall.com/article/233/33-arduino-สอนใช้งาน-arduino-วัดระยะทางด้วยเซ็นเซอร์วัดระยะทาง-ultrasonic-module-hc-sr04> (วันที่ค้นข้อมูล : 29 มีนาคม 2563)
- [9] เซเรฟ. [Robot] การใช้ชุดขับมอเตอร์ L298N Dual H-Bridge Motor Controller [ออนไลน์]. เข้าถึงได้จาก : <http://naringroup.blogspot.com/2016/03/robot-l298n-dual-h-bridge-motor.html>
- [10] ROHM Semiconductor. Controlling DC Brush Motors with H-bridge Driver ICs (Research Proposal) [ออนไลน์]. เข้าถึงได้จาก :
https://www.rohm.com/documents/11303/41217/100260.H-BRDG_WP_Jan09.pdf (วันที่ค้นข้อมูล : 29 มีนาคม 2563)
- [11] MT3608 DC-DC Step Up Converter Booster Power Supply [ออนไลน์]. เข้าถึงได้จาก :
<https://th.aliexpress.com/item/32671917935.html> (วันที่ค้นข้อมูล : 30 มีนาคม 2563)
- [12] กฎของโอล์ม (Ohm's Law) [ออนไลน์]. เข้าถึงได้จาก : <http://www.electric-factories.com/ohms-law/> (วันที่ค้นข้อมูล : 30 มีนาคม 2563)
- [13] ภาษา C# [ออนไลน์]. เข้าถึงได้จาก :
<https://sites.google.com/site/programmingm42/phasa-c>
(วันที่ค้นข้อมูล : 1 เมษายน 2563)
- [14] ภาษา C# [ออนไลน์]. เข้าถึงได้จาก : <http://marcuscode.com/lang/csharp/methods>
(วันที่ค้นข้อมูล : 1 เมษายน 2563)
- [15] Mr. Suteerat. เริ่มต้นเขียนโปรแกรมด้วย C# [ออนไลน์]. เข้าถึงได้จาก :
<http://computer.bps.in.th/suteerat/start> (วันที่ค้นข้อมูล : 1 เมษายน 2563)

ภาคผนวก ก

คำสั่งในส่วนของไมโครคอนโทรลเลอร์

1) คำสั่ง Arduino

```
#include <HCSR04.h>

HCSR04 hc(A0,A1); //initialisation class HCSR04 (trig pin , echo pin)

int ma1 = 2; // motor A
int ma2 = 3; // motor A
int mb1 = 4; // motor B
int mb2 = 5; // motor B

int sensor_A = 8; // sensor ตรวจจับสิ่งกีดขวางทาง ขวา ปกติมี output =1 ไม่เจอวัดดู อย่าลืมปรับ
//ระยะหด扯
int sensor_B = 9; // sensor ตรวจจับสิ่งกีดขวางทาง ซ้าย

int sensor_value_A;
int sensor_value_B;
int led=10;
int button=11;
int val=0;
int old_val=0;
int state=0;
int detectGoal=A3; //sensor ตรวจสอบเส้นชัย (สมมุติเส้นสนามน้ำเงิน note หากสีอื่นต้องมีการทดสอบ)
int detectRight=A4; //sensor ตรวจสอบเส้นขวา
int detectLeft=A5; // sensor ตรวจสอบเส้นซ้าย
int fieldvalue=0;
int sensorVcolor; //ด้านหน้า
int sensorRcolor; // ซ้าย
int sensorLcolor; // ขวา

void setup() {
    Serial.begin(9600);
    pinMode(ma1, OUTPUT);
    pinMode(ma2, OUTPUT);
    pinMode(mb1, OUTPUT);
    pinMode(mb2, OUTPUT);
```

```
pinMode(sensor_A,INPUT);
pinMode(sensor_B,INPUT);
pinMode(led,OUTPUT);
pinMode(button,INPUT);

pinMode(detectGoal,INPUT);
pinMode(detectRight,INPUT);
pinMode(detectLeft,INPUT);

}

}
```

```
void loop() {
//Serial.println( hc.dist() );
Serial.println(fieldvalue);

testsw();
testField(); // function ตรวจสอบ ขอบของสนามแข่งและเส้นชัย

//กลยุทธ์รุก
if(state==0){
testsensor(); // function ตรวจสอบการทำงานของ sensor ซ้าย และ ขวา
if(hc.dist()>=21&&sensor_value_A ==1&&sensor_value_B==1&&fieldvalue==0 ){
testforward(); }// function เดินหน้า
else if(hc.dist()<20&&sensor_value_A ==0&&sensor_value_B==1){ testMoveleft(); } //
function เคลื่อนที่ไปทางซ้าย
else if(hc.dist()<20&&sensor_value_A ==1&&sensor_value_B==0){ testMoveright(); } //
function เคลื่อนที่ไปทางขวา
else if (hc.dist()<20&&sensor_value_A ==0&&sensor_value_B==0){ testBreak(); } //
function หยุดรถ
```

```

        else if(fieldvalue==1){ testStop(); delay(1000); testReturn();} // function กลับรถ ถอย
        หลังประมาณ 3 วิ แล้วหมุนตัวกลับรถ ต้อง test เวลาที่ใช้
        else if(fieldvalue==2){ testStop(); delay(1000); testMoveleft();}
        else if(fieldvalue==3){ testStop(); delay(1000); testMoveright();}
        else testStop(); // function หยุดรถ
    }

    // ** ล้อแบบ mecanum
    //** function ที่ต้องเพิ่มเติม คือ การไปถึงเส้นแล้วกลับ รถ แล้วเคลื่อนรถกลับไปยังจุด start (ใช้
Sensor ตรวจสอบ สี หรือ Sensor วัดระยะ)
    // ** function ตรวจสอบขอบ สนาม อาจใช้ sensor 3 ตัว

    //กลยุทธ์รับ
else if (state==1){
    testsensor();
    if(hc.dist()<=30&&sensor_value_A ==1&&sensor_value_B==1 ){ testStop(); } // function หยุด
รถ
    else if(hc.dist()>20&&sensor_value_A ==0&&sensor_value_B==1){testMoveright(); } //
function เคลื่อนที่ไปทางขวา
    else if(hc.dist()>20&&sensor_value_A ==1&&sensor_value_B==0){ testMoveleft();} //
function เคลื่อนที่ไปทางซ้าย
    else if(hc.dist()>=40&&sensor_value_A ==1&&sensor_value_B==1&&fieldvalue==0)
{delay(3000);searchtest();} // function ค้นหา
    else if(fieldvalue==2){ testStop(); delay(1000); testMoveleft();}
    else if(fieldvalue==3){ testStop(); delay(1000); testMoveright();}
    else testStop(); // function หยุดรถ */
}

}

void testField(){ // function ตรวจสอบขอบสนาม
sensorVcolor=analogRead(detectGoal);
sensorRcolor=analogRead(detectRight);

```

```

sensorLcolor=analogRead(detectLeft);

if(sensorVcolor>=50&&sensorVcolor<=200){fieldvalue=1;}
else if (sensorRcolor>=50&&sensorRcolor<=200){fieldvalue=2;}
else if (sensorLcolor>=50&&sensorLcolor<=200){fieldvalue=3;}
else { fieldvalue=0; }

}

void testsensor(){
sensor_value_A = digitalRead(sensor_A);
sensor_value_B =digitalRead(sensor_B);
}

void testsw(){
val=digitalRead(button);
if( (val==1) && (old_val==0)) {
state=!state;
old_val=val;
delay(500);
if(state==0){
digitalWrite(10,LOW);
}
else {
digitalWrite(10,HIGH);
}
}
}

void searchtest(){ //function ทำการค้นหาฝ่ายรุก
if(fieldvalue==0){ testMoveright();}

else if(fieldvalue==2){ testStop(); delay(1000); do testMoveleft(); while(fieldvalue==3); }

else if(fieldvalue==3){ testStop(); delay(1000); testMoveright();}

else testStop();
}

```

```
}
```

```
void testforward(){
    digitalWrite(ma1, HIGH);
    analogWrite(mb2, LOW);
    digitalWrite(mb1, HIGH);
    analogWrite(mb2, LOW);
}

void testMoveleft(){
    digitalWrite(ma1, LOW);
    analogWrite(mb2, HIGH);
    digitalWrite(mb1, HIGH); // ต้องมีการ test ค่าความเร็ว
    analogWrite(mb2, LOW); // ต้องมีการ test ค่าความเร็ว
}
```

```
void testMoveright(){
    digitalWrite(mb1, LOW);
    analogWrite(mb2, HIGH);
    digitalWrite(ma1, HIGH); // ต้องมีการ test ค่าความเร็ว
    analogWrite(mb2, LOW); // ต้องมีการ test ค่าความเร็ว
}
```

```
void testBreak(){
    digitalWrite(ma1, HIGH);
    analogWrite(mb2, HIGH);
    digitalWrite(mb1, HIGH);
    analogWrite(mb2, HIGH);
}
```

```
void testStop(){
    digitalWrite(ma1,LOW);
    analogWrite(mb2, LOW);
```

```
digitalWrite(mb1, LOW);
analogWrite(mb2, LOW);
}

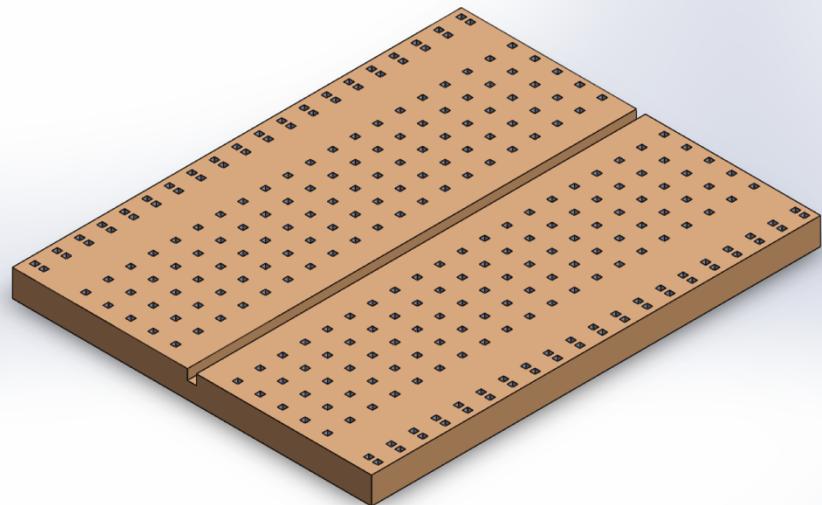
void testReturn(){ // funtion กลับรถ ถอยหลังประมาณ 3 วิ และหมุนตัวกลับรถ ต้อง test เวลาที่ใช้
digitalWrite(ma1,LOW);
analogWrite(ma2, HIGH);
digitalWrite(mb1, LOW);
analogWrite(mb2, HIGH);
delay(1000);
digitalWrite(ma1, HIGH); //ต้องมีการ test ค่าความเร็ว
analogWrite(ma2, LOW);
digitalWrite(mb1, LOW);
analogWrite(mb2, HIGH); //ต้องมีการ test ค่าความเร็ว

// fieldvalue=0;
delay(1000);

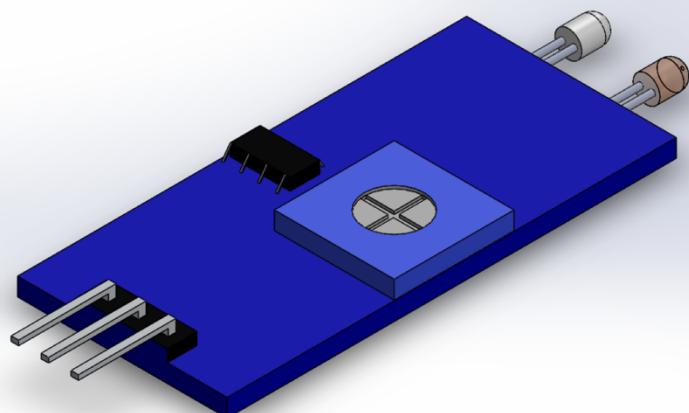
}
```

ภาคผนวก ข

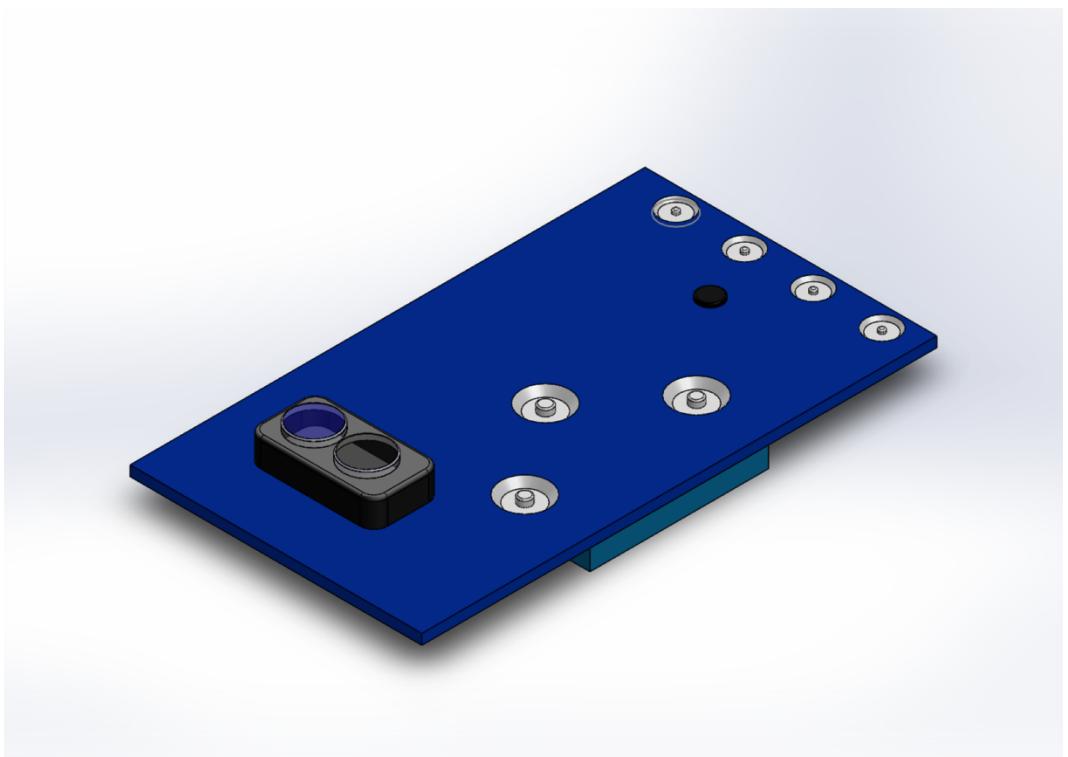
เบสบอร์ดและเซนเซอร์ที่ใช้ติดตั้งบนหุ่นยนต์



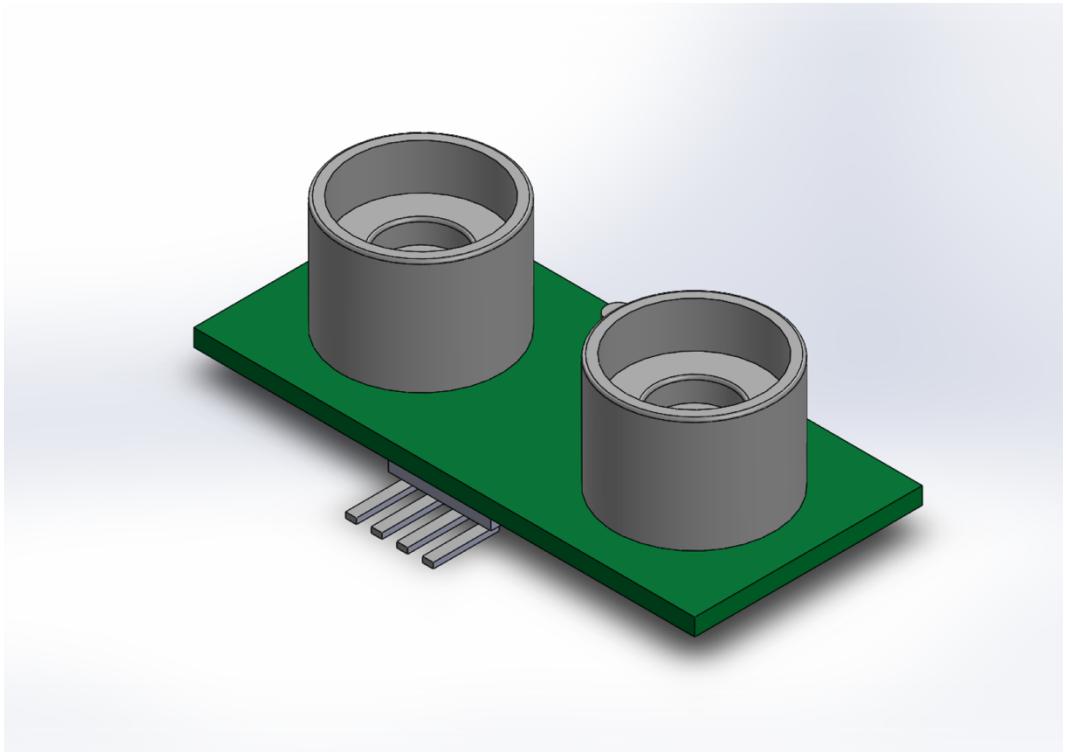
บะสบอร์ด 400 ช่องเสี่ยบ



IR Infrared Obstacle Avoidance Sensor Module



TCRT5000 Infrared Reflective sensor



Ultrasonic Module