



รายงาน

วิชา Pre-Project  
เรื่อง Robot Defense

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

จัดทำโดย

60010309	นางสาวณัฐนรี	ดวงเพียรราช
60010959	นายศรารุณี	คงเพชร
60010978	นางสาวศรินารถ	ดินดี

ภาคเรียนที่ 2 ปีการศึกษา 2562

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอุตสาหกรรม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา 01216747 Pre-Project ภาคเรียนที่ 2 ปีการศึกษา 2562 รายงานฉบับนี้จัดทำขึ้นเพื่อออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์

โดยได้จัดทำโปรเจกต์ด้วยการใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของรถผ่านตัวโปรแกรม Arduino โดยต่อวงจรจำลองของหุ่นยนต์รถต้นแบบจากโปรแกรม TinkerCad.com เพื่อใช้งานจริง รวมถึงการออกแบบตัวรถจากโปรแกรม Solidworks สำหรับทำ 3D- Printer เพื่อใช้งานจริงอีกด้วย รายงานเล่มนี้จะสำเร็จไม่ได้หากขาดความอนุเคราะห์จาก ผศ.ดร.อุดม จันทร์จรัสสุข และเพื่อนชั้นปีที่ 3 ภาควิชาวิศวกรรมอุตสาหการ จนสำเร็จออกมาเป็นรายงานเล่มนี้

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงาน Robot Defense จะเป็นประโยชน์ต่อผู้ที่สนใจทุกท่านและหากมีข้อผิดพลาดประการใด ขออภัยไว้ ณ ที่นี้

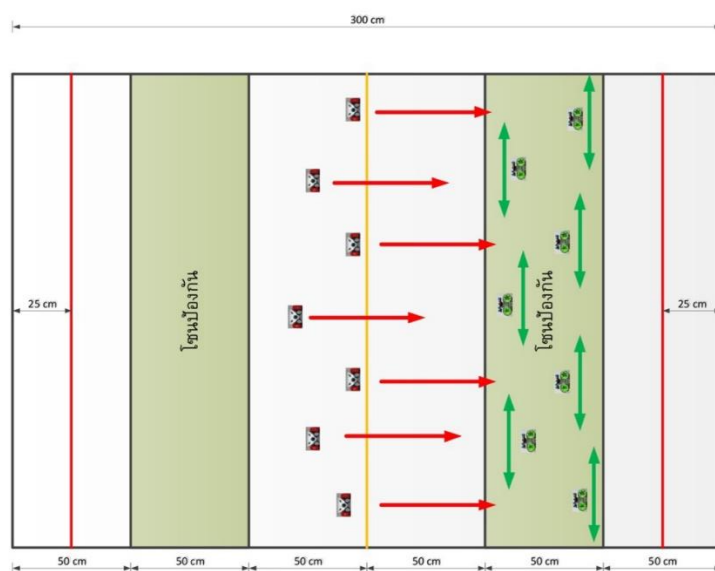
## สารบัญ

เรื่อง	หน้า
คำนำ	ก
สารบัญ	ข
ปัญหาหรือโจทย์ที่ต้องการแก้	1
แนวคิดและกลยุทธ์	2
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
วงจร	13
Car Body Design	15
โปรแกรม	18
เอกสารอ้างอิง	ค

## ปัญหาหรือโจทย์ที่ต้องการแก้

การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่น บอลลุนดำน หรือ เล่นเตย โดยแบ่งเป็นทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้าม จนผ่านเส้นแดง แล้วกลับมามากอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้ ก็จะเป็นฝ่ายชนะในการแข่งขันรอบนั้น หุ่นยนต์ที่ถูกจับได้จะถูกตัดออกจากการแข่งขันในรอบนั้น ส่วนทีมรับ จะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

### ลักษณะสนามแข่งขัน



### วัตถุประสงค์ของโครงการ

- 1) จำลองหุ่นยนต์รถดันแบบ
- 2) การเขียนตัวโปรแกรมควบคุมการทำงานหรือ Code Arduinoเพื่อใช้งานจริง
- 3) การต่อวงจรและเข้าใจการทำงานของอุปกรณ์ต่างๆเพื่อใช้งานจริง
- 4) การออกแบบตัวรถสำหรับ 3D Printer
- 5) การทำงานเป็นทีมและการแบ่งเวลา
- 6) เพื่อศึกษาการสร้างหุ่นยนต์ให้เป็นพื้นฐานในการต่อยอดการทำหุ่นยนต์เพื่อเพิ่มประสิทธิภาพในการผลิตในอนาคต

### ขอบเขตของโครงการ

- 1) หุ่นยนต์รถดันแบบขนาดไม่เกิน 10x10 เซนติเมตร

2) ใช้ตัวขับเคลื่อนหรือมอเตอร์ได้ 2 ตัว

3) ออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลุยุทธ์ออกเป็น 2 กลุยุทธ์ คือ กลุยุทธ์รุกและกลุยุทธ์รับ โดยควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์

### กลุยุทธ์ของ Robot car

#### กลุยุทธ์ฝ่ายรุก

การวางตัวรถจะวางเป็นแนวตั้งฉากกับเส้นชัย โดยมี Ultrasonic sensor ติดที่ด้านหน้าและหลังของตัวรถ มี IR sensor ติดที่ด้านซ้ายและขวาของตัวรถ และมี Sensor ตรวจจับสีติดข้างล่างบริเวณด้านหน้ารถ โดยการทำงาน คือ ถ้า Sensor รอบรถตรวจจับวัตถุไม่ได้ในระยะ 10 cm.รถจะเดินทางด้วยความเร็ว 180 bits ไปเรื่อยๆ จนถึงเส้นสีแดงของฝ่ายตรงข้ามหลังจากนั้นจะหยุดและเดินถอยหลังกลับมาเส้นสีแดงฝั่งตนเอง แต่ถ้า Sensor ด้านหน้า ซ้ายหรือขวาตรวจจับวัตถุได้ในระยะ 5 cm.หรือน้อยกว่า รถจะหยุดและถอยหลังด้วยความเร็ว 255 bits เป็นเวลา 5 วินาที ไปจนกว่า Sensor ด้านหน้า ซ้ายหรือขวาตรวจจับวัตถุไม่ได้ในระยะ 10 cm รถจะหยุดและหลังจากนั้นรถจะเดินทางด้วยความเร็ว 180 bits ไปเรื่อยๆ

#### กลุยุทธ์ฝ่ายรับ

การวางตัวรถจะวางเป็นแนวขนานกับเส้นชัย โดยมี Ultrasonic sensor ติดที่ด้านหน้าและหลังของตัวรถ มี IR sensor ติดที่ด้านซ้ายและขวาของตัวรถ และมี Sensor ตรวจจับสีติดข้างล่างบริเวณด้านหน้ารถ โดยการทำงาน คือ รถจะวิ่งเดินทางถอยหลัง 5 วินาที ด้วยความเร็ว 180 bits ถ้า Sensor ด้านหน้าหรือหลังตรวจจับวัตถุได้ในระยะ 10 cm. รถจะเดินทางไปหาวัตถุนั้นด้วยความเร็ว 255 bits เพื่อชนฝ่ายตรงข้ามให้ออกจากการแข่งขัน แต่ถ้า Sensor ด้านซ้ายหรือขวาตรวจพบรถจะหยุดจนกว่า Sensor ด้านซ้ายหรือขวา จะไม่ตรวจพบวัตถุและหลังจากนั้นรถจะวิ่งเดินทางถอยหลัง 5 วินาที ด้วยความเร็ว 180 bits ตามเดิม

## ทฤษฎีและหลักการที่เกี่ยวข้อง

### 1. ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ (MCU) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน ไมโครคอนโทรลเลอร์สำหรับโครงการนี้ คือ บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3



รูปที่ 1. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

### บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 [1]

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้รับความนิยม ซึ่งส่วนใหญ่โปรเจกต์และไลบรารีต่างๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็น ขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ Arduino ซึ่งบอร์ด Arduino Uno ได้ถูกพัฒนาขึ้นมา ตั้งแต่ R2, R3 และมีรุ่นชิปไอซีเป็นแบบ SMD

คุณสมบัติของบอร์ดดังนี้

ไอซีไมโครคอนโทรลเลอร์	ATmega328
แรงดันไฟฟ้าเลี้ยงไอซีไมโครคอนโทรลเลอร์	มีค่า 5 โวลต์
แรงดันไฟฟ้าป้อนที่บอร์ดไมโครคอนโทรลเลอร์	อยู่ในช่วง 7 - 12 โวลต์
พอร์ตดิจิตอลอินพุต/เอาต์พุต (Digital VO)	14 พอร์ต (มี PWM output จำนวน 6 พอร์ต)
พอร์ตอนาล็อกอินพุต (Analog Input)	จำนวน 6 พอร์ต
สามารถจ่ายกระแสไฟฟ้า	40 มิลลิแอมป์ (mA)
สามารถจ่ายกระแสไฟฟ้าในพอร์ต 3.3 V	50 มิลลิแอมป์ (mA)
พื้นที่หน่วยความจำโปรแกรม	32 กิโลไบต์ (KB)
พื้นที่หน่วยความจำชั่วคราวแบบ SRAM	2 กิโลไบต์ (KB)
พื้นที่หน่วยความจำถาวรแบบ EEPROM	1 กิโลไบต์ (KB)
ใช้ความถี่สัญญาณนาฬิกา	16 เมกะเฮิรตซ์ (MHz)

โดยมีรายละเอียดการใช้งานเบื้องต้น ดังนี้

## 1. ภาคจ่ายไฟฟ้า

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 สามารถใช้พลังงานไฟฟ้าจากพอร์ต USB ของคอมพิวเตอร์ หรือแหล่งพลังงานไฟฟ้าจากภายนอกโดยบอร์ดสามารถเลือกแหล่งจ่ายไฟฟ้าได้โดย อัตโนมัติ ในส่วนของแหล่งพลังงานไฟฟ้าจากภายนอก สามารถใช้ได้ทั้งแบบไฟฟ้ากระแสสลับและ ไฟฟ้ากระแสตรงจากอะแดปเตอร์ หรือจากแบตเตอรี่โดยมีขั้วไฟฟ้าของอะแดปเตอร์สามารถเชื่อมต่อ ด้วยการเสียบปลั๊กขนาด 2.1 มม. เข้ากับแจ็คพาวเวอร์ของบอร์ด ช่วงแรงดันของแหล่งจ่ายไฟฟ้าที่ แนะนำควรมีค่าอยู่ในช่วง 7 - 12 โวลต์ แต่ถ้าใช้แรงดันไฟฟ้ามากกว่า 12 โวลต์ ส่งผลให้อิซิควบคุม แรงดันไฟฟาร้อนมากเกินไปและเกิดความเสียหายต่อบอร์ดได้ ขาพาวเวอร์ซัพพลาย มีดังนี้

- Vin เป็นขารับแรงดันไฟฟ้าเลี้ยงบอร์ด Arduino จากแหล่งจ่ายไฟฟ้าภายนอก
- 5 V เป็นขาจ่ายแรงดันไฟฟ้า 5 โวลต์ ที่ได้จากแรงดันจาก Vin ผ่านวงจรเรกกูเลเตอร์ภายในบอร์ด หรือจากแรงดันไฟฟ้าที่พอร์ต USB
- 3.3V เป็นขาจ่ายแรงดันไฟฟ้า 3.3 โวลต์ ที่สร้างขึ้นโดยวงจรเรกกูเลเตอร์ภายในบอร์ดจ่ายกระแสสูงสุดคือ 50 มิลลิแอมป์
- GND เป็นขากราวนด์

## 2. หน่วยความจำ

ไมโครคอนโทรลเลอร์ ATmega328 มีหน่วยความจำแบบแฟลชสำหรับการจัดเก็บโปรแกรม ขนาด 32 กิโลไบต์ (มีหน่วยความจำใช้สำหรับการบูต ขนาด 0.5 กิโลไบต์) มีหน่วยความจำชั่วคราวแบบสแตติกแรม (SRAM) ขนาด 2 กิโลไบต์ และมีหน่วยความจำถาวรแบบอีอีพรอม (EEPROM) ขนาด 1 กิโลไบต์

## 3. พอร์ตอินพุต - เอาต์พุต

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตดิจิตอลทั้งหมด 14 ขา สามารถเป็นได้ทั้งพอร์ตอินพุตหรือเอาต์พุต โดยใช้ฟังก์ชัน pinMode (), digitalWrite () และ digitalRead () แต่ละขาทำงานที่แรงดัน 5 โวลต์ สามารถจ่ายหรือรับกระแสไฟฟ้าได้สูงสุด 40 มิลลิแอมป์ และมียังมีหน้าที่พิเศษ ดังนี้ ตัวต้านทานต่อแบบพูลอัพอยู่ภายในมีค่าความต้านทาน 20 - 50 กิโลโอห์ม นอกจากนี้แล้ว บางพอร์ต

- พอร์ต 0 เป็นขา RX ใช้เป็นพอร์ตรับสัญญาณสื่อสารแบบอนุกรม
- พอร์ต 1 เป็นขา TX ใช้เป็นพอร์ตส่งสัญญาณสื่อสารแบบอนุกรม
- พอร์ต 2 และ 3 เป็นพอร์ตรับสัญญาณอินเตอร์รัพท์ภายนอก (Interrupts) พอร์ตเหล่านี้สามารถกำหนดค่าให้รับสัญญาณขัดจังหวะได้ทั้งแบบลอจิกสูง ลอจิกต่ำ หรือแบบอื่น ๆ
- พอร์ต 3, 5, 6, 9, 10 และ 11 เป็นพอร์ตส่งสัญญาณเอาต์พุตแบบ PWM ขนาด 8 บิต
- พอร์ต 10, 11, 12 และ 13 เป็นพอร์ตสื่อสารแบบ SPI

- พอร์ต 13 เป็นพอร์ตควบคุมแอลอีดีที่ติดตั้งบนบอร์ด เมื่อขา 13 จ่ายเอาต์พุตลอจิก “1” ทำให้แอลอีดีติดสว่าง และเมื่อจ่ายลอจิก “0” ทำให้แอลอีดีดับ

#### 4. อนาล็อกอินพุต

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตอนาล็อกอินพุต 6 ขา คือ ขา AO - ขา A5 ซึ่งแต่ละช่องมีความละเอียดขนาด 10 บิต แบ่งระดับความแตกต่างได้ 1,024 ค่า โดยเริ่มต้นจากระดับแรงดัน 0 โวลต์ จนถึงระดับ 5 โวลต์ และสามารถเปลี่ยนระดับแรงดันอ้างอิงได้โดยใช้ แรงดันอ้างอิงจากภายนอกที่ขา AREF ร่วมกับฟังก์ชัน `analogReference()` นอกจากนี้ยังมี บางขาที่มีหน้าที่พิเศษ ดังนี้

- พอร์ต A4 (SDA) และพอร์ต A5 (SCL) เป็นพอร์ตสื่อสารแบบ IRC
- พอร์ต Aref แรงดันอ้างอิงสำหรับอินพุตอนาล็อก ใช้งานร่วมกับฟังก์ชัน `analog Reference ()`

#### 5. การสื่อสาร

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตสื่อสารเพื่อเชื่อมต่อกับคอมพิวเตอร์ หรือบอร์ด Arduino อื่นๆ หรือไมโครคอนโทรลเลอร์เบอร์อื่น ๆ หลายรูปแบบ ตามความสามารถของไมโครคอนโทรลเลอร์ ATmega328 ที่มีพอร์ตสื่อสารอนุกรมแบบ UART ที่พอร์ตดิจิทัล ขา 0 (R และพอร์ตดิจิทัล ขา 1 (TX) ช่องทางการสื่อสารแบบอนุกรมยังเชื่อมโยงผ่านพอร์ต USB และยัง ปรากฏเป็นพอร์ต COM เสมือนซอฟต์แวร์บนเครื่องคอมพิวเตอร์ ด้วยเฟิร์มแวร์ 8U2 คอมพิวเตอร์ สามารถเชื่อมต่อได้โดยใช้ไดรเวอร์ USB มาตรฐาน และไม่ต้องใช้ไดรเวอร์ภายนอก

## 2. IR Infrared Obstacle Avoidance Sensor Module [2]

โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุขวาง IR Infrared Obstacle Avoidance Sensor Module โดยโมดูลนี้ จะมีตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ(สีขาว) infrared จะส่งสัญญาณออกมา และเมื่อมีวัตถุมาบัง คลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้อนกลับไปเข้าตัวรับสัญญาณ (สีดำ) สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้

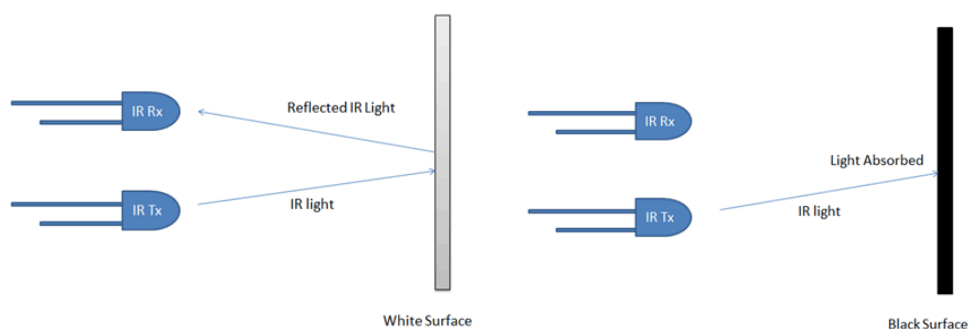


รูปที่ 2 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุขวาง



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

หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่สะท้อนกลับไปได้ จะแสดงค่า เป็น 1



รูปที่ 3 การส่ง และรับสัญญาณ infrared

ซึ่งจะทำให้วงจรภายในรับรู้ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาต์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate

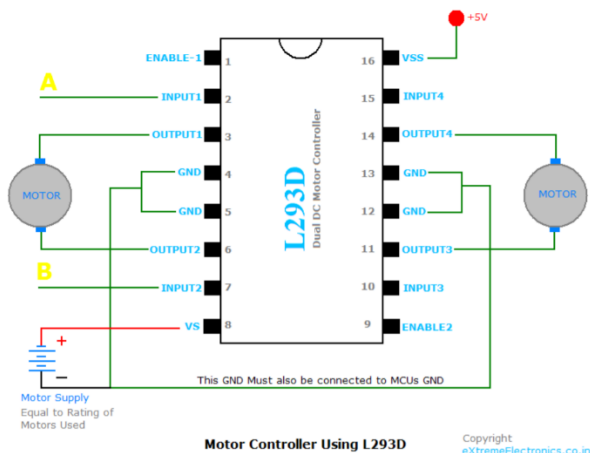
คุณสมบัติของ IR Infrared Obstacle Avoidance Sensor Module

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

### 3. DC Motor Speed Control L293D [3]

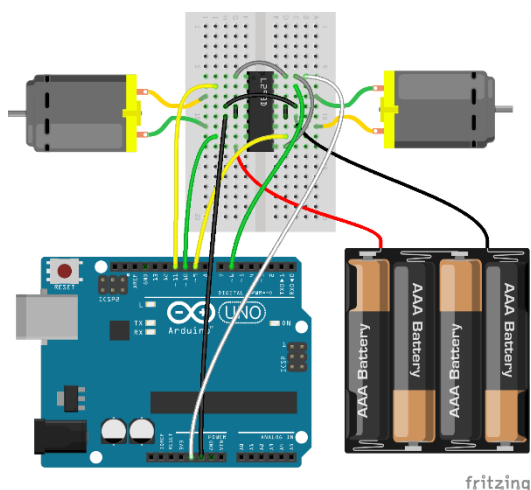
ประกอบด้วย 2 ส่วนใหญ่ ๆ คือ 1) H-bridge Driver และ 2) Pulse-width modulation (PWM) สิ่งที่เป็นพลังขับเคลื่อนหลักให้กับหุ่นยนต์นั้น ก็คงจะไม่พ้น มอเตอร์ ซึ่งต้องการการควบคุม จาก Motor Driver ที่จะมาควบคุม ทั้ง ทิศทาง และ ความเร็ว ของมอเตอร์ ซึ่งทางผู้จัดทำได้เลือกใช้ L293D Dual H-Bridge Motor Controller

L293D เป็นไอซีตัวถัง DIP-16 รองรับการขับมอเตอร์ 2 ตัว ทนกระแสได้สูงสุด 600mA รองรับมอเตอร์ที่ใช้แรงดันไฟฟ้า 4.5V ถึง 36V ใช้แรงดันควบคุมได้ 4.5V ถึง 36V วงจรขับมอเตอร์ด้วยไอซี L293D



รูปที่ 4 วงจรขับมอเตอร์ด้วยไอซี L293D

ไอซี L293D นิยมใช้งานกับมอเตอร์ขนาดเล็ก เช่น มอเตอร์ที่ใช้ในรถบังคับ เนื่องจากมีขนาดเล็ก ราคาถูก ไม่จำเป็นต้องมีอุปกรณ์ต่อร่วม ตัวอย่างการใช้ไอซี L293D ขับมอเตอร์ขนาดเล็ก ควบคุมด้วยบอร์ด Arduino Uno R3 แสดงดังรูปที่ 6 [4]



รูปที่ 5 วงจรต่อใช้งานบอร์ด Arduino ควบคุมมอเตอร์ด้วย L293D

จากรูปที่ 6 การเขียนโปรแกรมควบคุมมอเตอร์ให้หมุนตามเข็มนาฬิกา หรือหมุนทวนเข็มนาฬิกา ทำได้โดยสั่งให้ D11 และ D10 มีลอจิกที่แตกต่างกัน เช่น การสั่ง D11 เป็นลอจิก 1 และสั่ง D10 เป็นลอจิก 0 หรือการสั่ง D11 เป็นลอจิก 0 และสั่ง D10 เป็นลอจิก 1 หากกำหนดให้ทั้ง D11 และ D10 เป็นลอจิก 0 ทั้งคู่ หรือเป็น 1 ทั้งคู่ จะทำให้มอเตอร์หยุดหมุน หากต้องการควบคุมความเร็วของมอเตอร์ด้วย ทำได้โดยป้อนสัญญาณ PWM เข้าไปแทน

## คุณสมบัติ

INTERNAL CLAMP DIODES
600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
ENABLE FACILITY
1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
LOGICAL \"0\" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
OVERTEMPERATURE PROTECTION

#### 4. Ultrasonic Distance Sensor [5]

เซนเซอร์วัดระยะทางด้วย Ultrasonic ใช้หลักการ ส่งคลื่นเสียงความถี่ต่ำ Ultrasonic ไปเมื่อคลื่นเสียงกระทบกับวัตถุจะมีการสะท้อนกลับมา เซนเซอร์จับเวลาที่ส่งคลื่นเสียงออกไปจนถึงคลื่นเสียงสะท้อนกลับมา เมื่อนำมาคำนวณกับเวลาที่เสียงเดินทางในอากาศ ก็จะได้ระยะทางออกมาโมดูล Ultrasonic ตรวจจับวัตถุ คำนวณระยะทางโดยใช้คลื่น มีลักษณะเป็นกรวยและไม่ใช้เส้นตรง จึงเหมาะสำหรับใช้ตรวจจับสิ่งกีดขวางด้วย

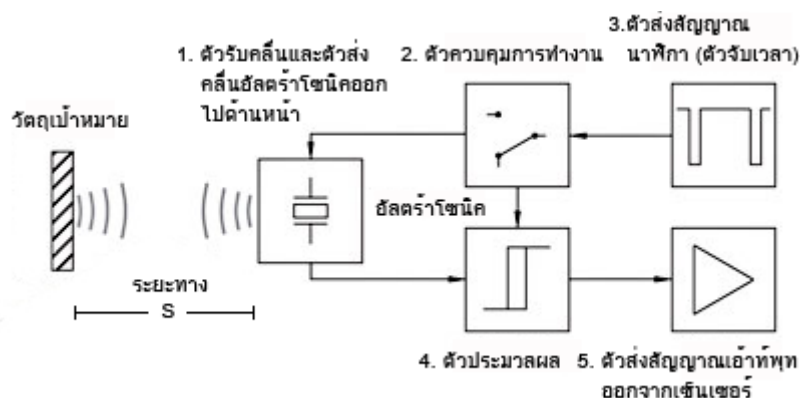


รูปที่ 6 Ultrasonic Distance Sensor

#### หลักการทำงานของอัลตราโซนิกเซนเซอร์

อัลตราโซนิกเซนเซอร์เป็นเซนเซอร์ที่ใช้คลื่นเสียงในการตรวจจับตำแหน่งของวัตถุ โดยส่วนประกอบของตัวเซนเซอร์จะประกอบด้วย

- 1.ตัวส่งคลื่นอัลตราโซนิกและตัวรับคลื่นอัลตราโซนิก (อัลตราโซนิกทรานสดิวเซอร์)
- 2.ตัวควบคุมการทำงาน
- 3.ตัวส่งสัญญาณนาฬิกา
- 4.ตัวประมวลผล
- 5.ตัวส่งสัญญาณเอาท์พุท



รูปที่ 7 ไดอะแกรมภายในอัลตราโซนิกเซ็นเซอร์

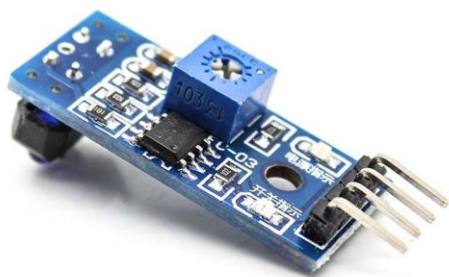
โดยตัวเซ็นเซอร์จะทำงานโดย ตัวส่งสัญญาณจะส่งสัญญาณนาฬิกาไปที่ตัวคอนโทรลเลอร์ เพื่อควบคุมการแปลงสัญญาณ แล้วส่งไปที่ตัวอัลตราโซนิกทรานสดิวเซอร์ซึ่งแบ่งเป็นสองส่วนคือ ตัวส่งและตัวรับ ตัวส่งจะสร้างคลื่นเสียงอัลตราโซนิก จากสัญญาณไฟฟ้าแล้วส่งคลื่นเสียงความถี่สูงหรืออัลตราโซนิกออกไปเป็นแนวตรง และเมื่อคลื่นเสียงอัลตราโซนิกไปกระทบกับวัตถุใดๆ ตามหลักการของคลื่นเสียง คือ มุมตกกระทบเท่ากับมุมสะท้อน คลื่นเสียงจะถูกสะท้อนกลับมายังตัวรับคลื่นเสียงอัลตราโซนิก เมื่อตัวรับได้รับคลื่นเสียงที่ถูกสะท้อนกลับมาแล้ว ตัวรับจะแปลงคลื่นเสียงอัลตราโซนิกนั้นเป็นสัญญาณไฟฟ้าแล้วส่งต่อ ให้ตัวประมวลผล ตัวประมวลผลจะทำการคำนวณค่าระยะห่างจากระยะทางที่คลื่นเสียงเดินทางไปและเดินทางกลับอย่างแม่นยำ และส่งค่าที่คำนวณได้ไปให้ตัวส่งสัญญาณเอาต์พุต เพื่อส่งสัญญาณเอาต์พุตไปให้อุปกรณ์อื่นต่อไป

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

## 5. TCRT5000 Infrared Reflective sensor [6]

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็กสะดวกในการนำไปใช้ติดตั้งกับงานจำพวก หุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น ซึ่งสามารถแสดงได้ดังรูปที่ 8 และคุณสมบัติดังตารางที่ 5 โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณเป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัวหลอดโฟโตไดโอดที่ทำหน้าที่รับแสงอินฟราเรด โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal แต่สำหรับบางโมดูลอาจจะรองรับ output แบบ Analog signal ด้วย ส่วนตัว R ปรับค่านั้นใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจพบวัตถุของตัวเซ็นเซอร์



รูปที่ 8 ส่วนประกอบของ TCRT5000 Infrared Reflective sensor

คุณสมบัติเซ็นเซอร์วัดระยะทาง TCRT5000 Infrared Reflective sensor

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

## 6. การเขียนโปรแกรม arduino เบื้องต้นจากโครงสร้างภาษา C/C++ [7]

โปรแกรมของ Arduino แบ่งได้ เป็นสองส่วนคือ

ภาษาซีของ Arduino จะจัดรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยๆ หลายๆ ส่วน โดยเรียกแต่ละส่วนว่า ฟังก์ชัน และ เมื่อนำฟังก์ชัน มารวมเข้าด้วยกัน ก็จะเรียกว่าโปรแกรม โดยโครงสร้าง การเขียนโปรแกรมของ Arduino นั้น ทุกๆ โปรแกรมจะต้องประกอบไปด้วยฟังก์ชันจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมีฟังก์ชัน จำนวน 2 ฟังก์ชัน คือ setup() และ loop() จะได้เห็นได้โครงสร้างพื้นฐานของภาษาซี ที่ใช้กับ Arduino นั้นจะประกอบไปด้วย 3 ส่วนใหญ่ๆด้วยกัน คือ

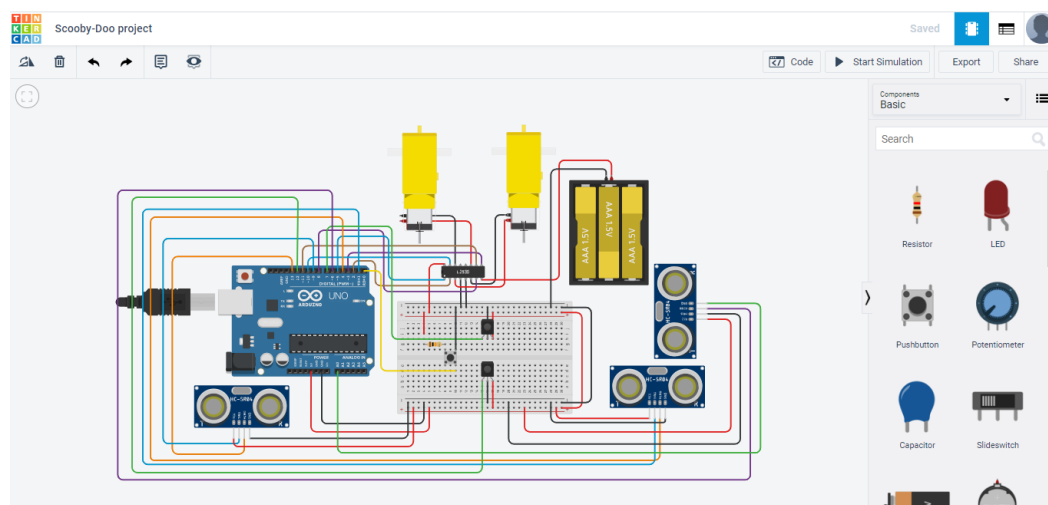
1. Header ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรม ซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่างๆรวมไปถึงส่วนของการประกาศตัวแปร และค่าคงที่ต่างๆที่จะใช้ในโปรแกรม [8]

2. setup() ในส่วนนี้เป็นฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆโปรแกรม ถึงแม้ว่าในบางโปรแกรมจะ ไม่ต้องการใช้งานก็ยังจำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใดๆไว้ในระหว่างวงเล็บปีกกา {} ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชัน โดยฟังก์ชันนี้จะใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรม ทำงานเพียงรอบเดียวตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่า การทำงานต่างๆ เช่น การกำหนดหน้าที่การใช้งานของ PinMode และการกำหนดค่า Baudrate สำหรับใช้งานพอร์ตสื่อสารอนุกรม เป็นต้น [8]

3.loop() เป็นส่วนฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆโปรแกรมเช่นเดียวกับฟังก์ชัน setup() โดยฟังก์ชัน loop() นี้จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวนรอบซ้ำๆกันไปไม่รู้จบ ซึ่งถ้าเปรียบเทียบกับรูปแบบของ ANSI-C ส่วนนี้ก็คือ ฟังก์ชัน main() นั่นเอง [8]

## 7. โปรแกรม Tinkercad [9]

เว็บ tinkercad.com 3.1k (tinkercad) เป็นเว็บ IDE ตัวหนึ่งที่เราสามารถออกแบบและจำลองการทำงานของวงจร ลาย PCB หรือแม้กระทั่งรูปทรงสามมิติด้วย ซึ่งเครื่องมือเครื่องใช้ที่นำมาให้ใช้งานค่อนข้างที่จะครบครัน ในโครงการผู้จัดทำขอกล่าวถึงเฉพาะส่วนของ Circuits



รูปที่ 9 tinkercad.com

### อุปกรณ์พื้นฐาน Basic หมวด Circuits ในตัวโปรแกรม

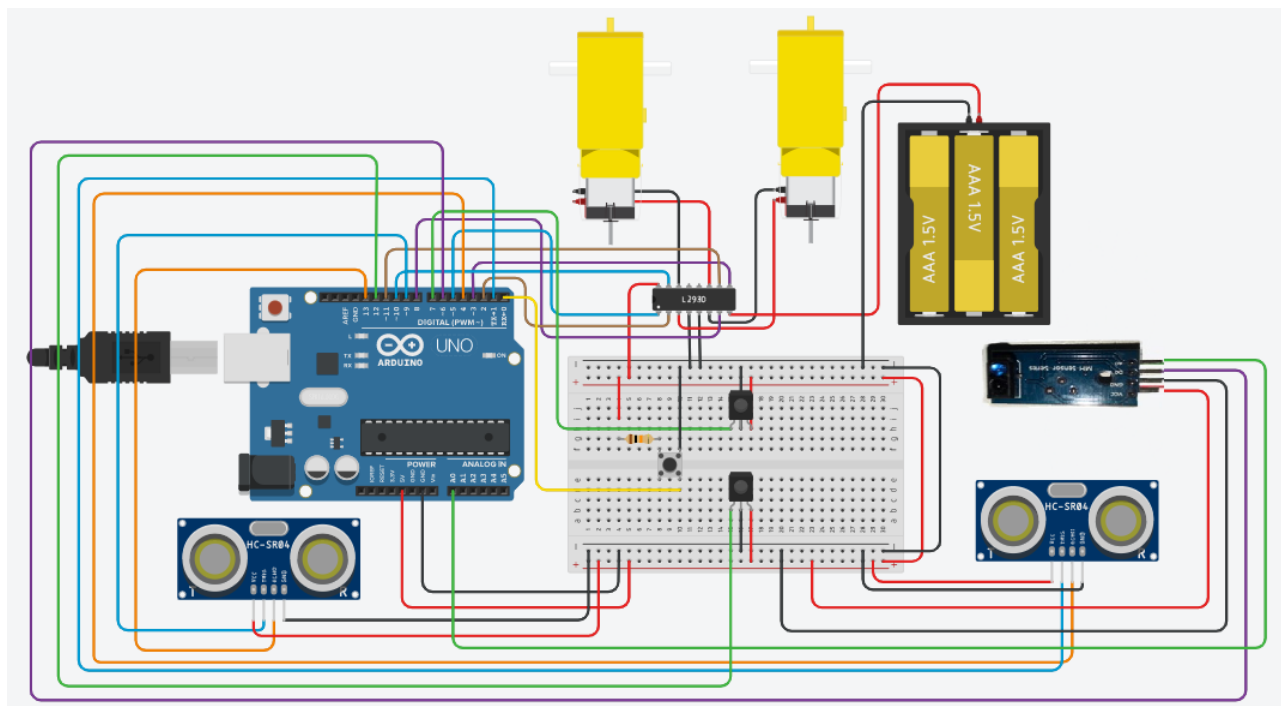
1. General ได้แก่ Resistor, Capacitor, Polarized Capacitor, Diode, Zener Diode และ Inductor
2. Input ได้แก่ Pushbutton, Potentiometer, Slideswitch, Photoresistor, Photodiode, Ambient Light, IR sensor, Ultrasonic Distance Sensor, PIR Sensor, Tilt Sensor, Tilt Sensor 4-pin, Temperature Sensor[TMP36], Gas Sensor, Keypad 4x4, DIP Switch DPST, DIP Switch SPST x 4 และ DIP Switch SPST x 6
3. Output ได้แก่ LED, LED RGB, Light bulb, NeoPixel, NeoPixel Jewel, NeoPixel Ring 12, NeoPixel Ring 16, NeoPixel Ring 24, Vibration Motor, DC Motor, DC Motor with encoder, DC Motor with Encoder, Micro Servo, Hobby Gearmotor, Piezo, IR remote, 7 segment Display และ LCD 16 x 2
4. Power ได้แก่ 9V Battery, 1.5 Battery และ Coin Cell 3V Battery
5. Breadboards ได้แก่ Breadboard, Breadboard Small และ Breadboard Mini

6. Microcontrollers ได้แก่ Arduino Uno R3 และ ATtiny
7. Instruments ได้แก่ Multimeter, Power Supply, Function Generator และ Oscilloscope
8. Integrated Circuits ได้แก่ Timer, Dual Timer, 741 Operational Amplifier, Quad comparator, Dual comparator และ Optocoupler
9. Power Control ได้แก่ NPN Transistor, PNP Transistor, Small Signal nMOS Transistor, Small Signal pMOS Transistor, nMOS Transistor(MOSFET) และ pMOS Transistor(MOSFET), TIP120, Relay SPDT, 5V Regulator[LM7805], 3.3V Regulator[LD1117V33], Pololu Simple Motor ConTroller และ H-bridge Motor Driver
10. Networking ได้แก่ Wifi Module(ESP8266)
11. Connectors ได้แก่ 8 Pin Header และ USB standard A
12. Logic ได้แก่ 74HC00, 74HC02, 74HC08, 74HC32, 74HC86 และ 74HC04, 74HC14, 74HC132, 74HC10, 74HC11, 74HC27, 74HC20, 74HC21, 74HC73, 74HC74, 74HC93, 74HC595, 74HC4017 และ CD4511

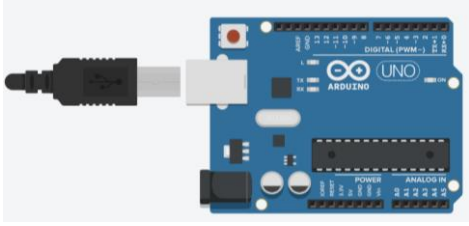
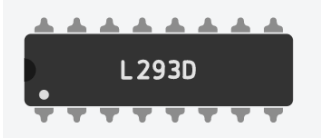

## วงจร

### แบบจำลองวงจร

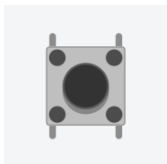
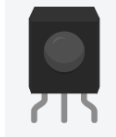
เขียนด้วยโปรแกรม tinkercad.com



### อุปกรณ์ที่ใช้

ชื่ออุปกรณ์	จำนวน	หน้าที่	ลักษณะ
Arduino Uno R3	1	ตัวบอร์ดสามารถต่อวงจรและป้อนโปรแกรมเพื่อให้วงจรทำงานได้	
H-bridge Motor Driver	1	ใช้ต่อกับตัวมอเตอร์ทั้งสองเพื่อควบคุมการทำงาน	
battery, AAA, no 1.5V Battery	3	ใช้จ่ายไฟให้กับตัวมอเตอร์	

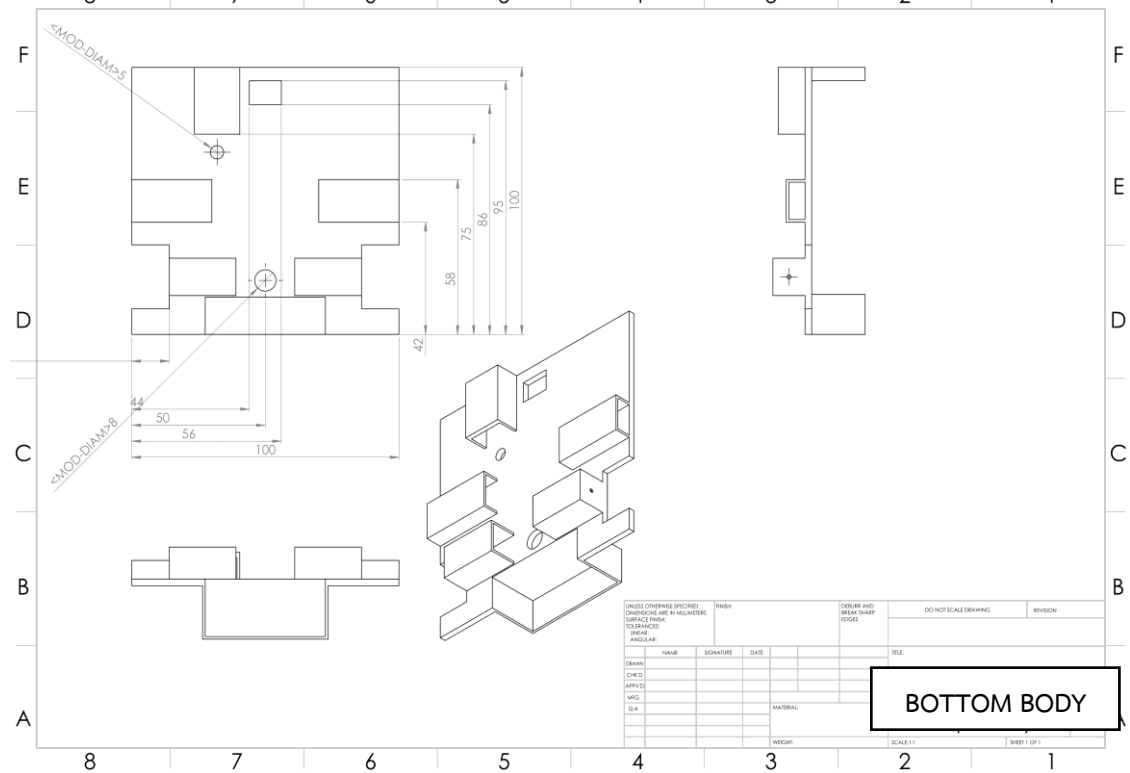
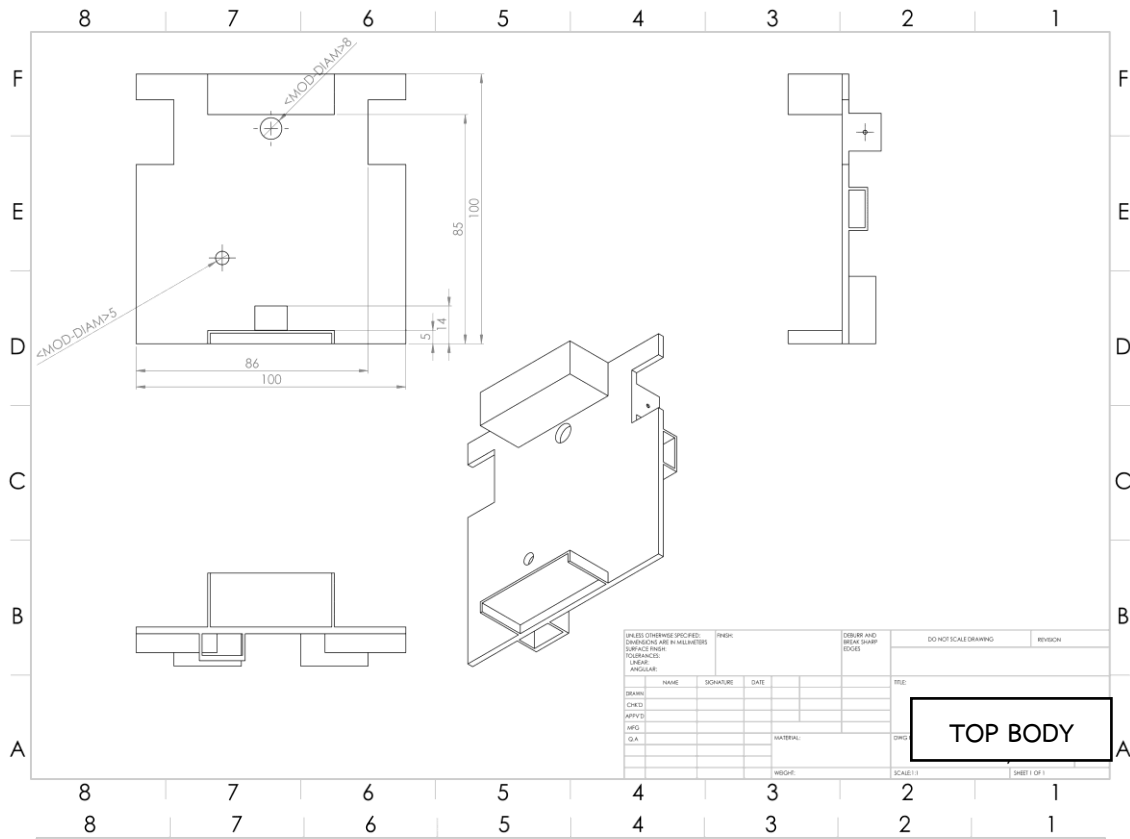


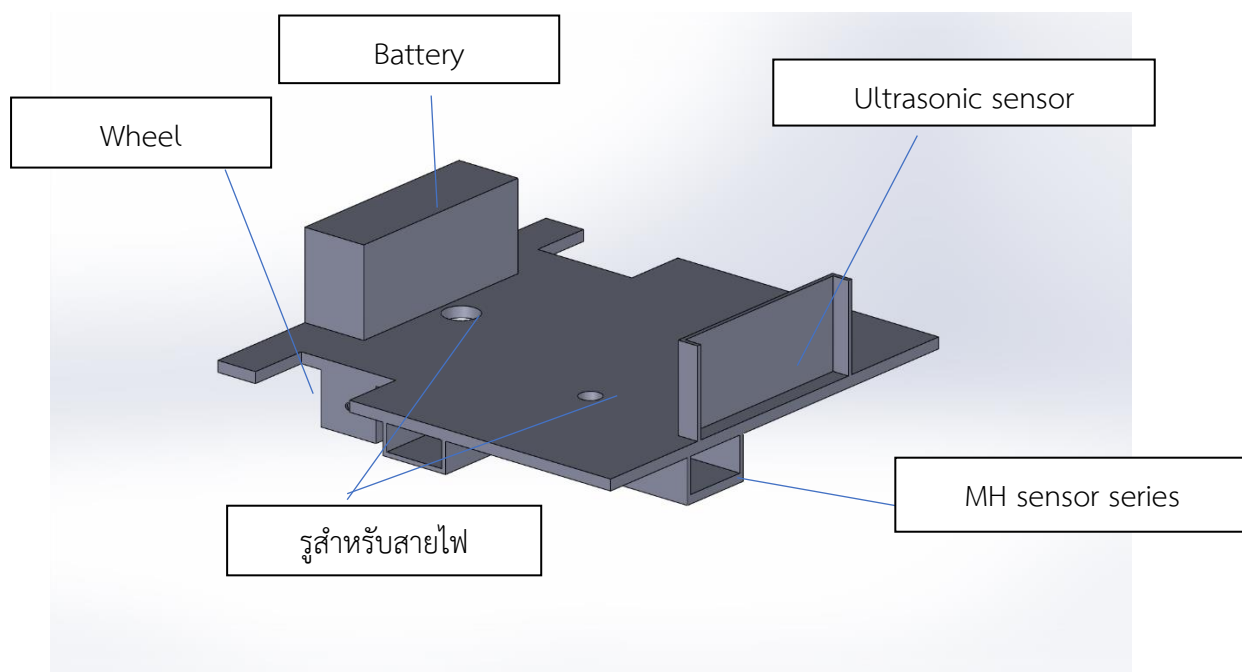
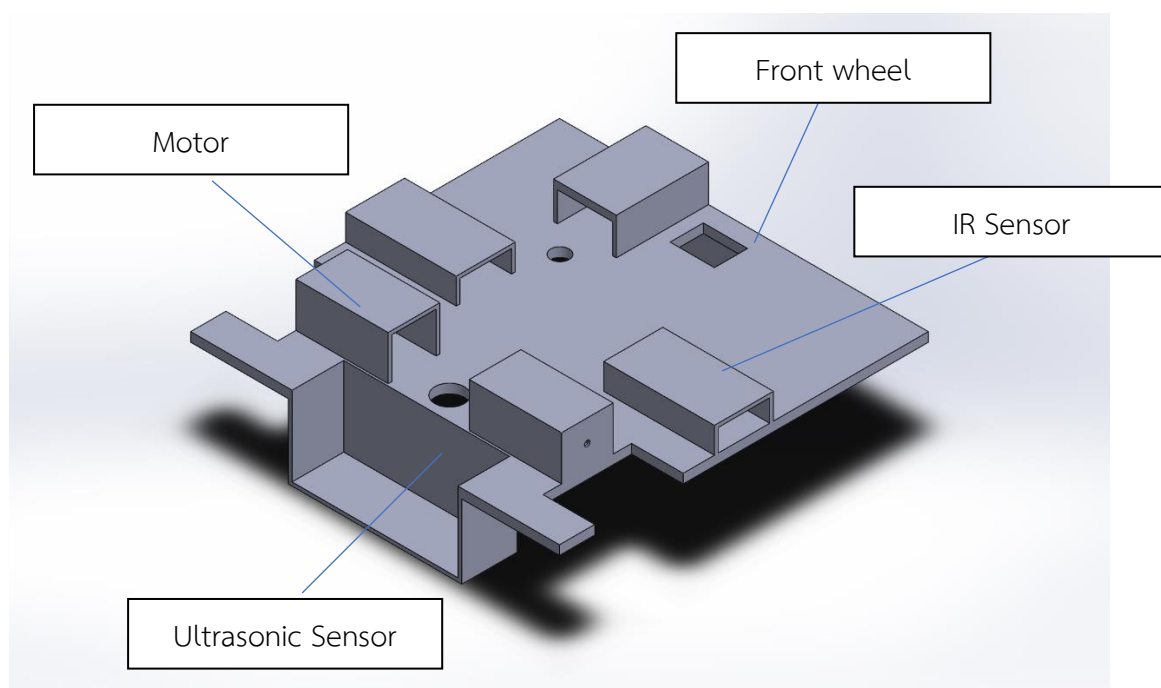
ชื่ออุปกรณ์	จำนวน	หน้าที่	ลักษณะ
Hobby Gear motor <b>Features:</b> - Suggested Voltage: 4.5VDC - No Load Speed: 256RPM - No Load Current: 190mA - Max. Load Current: 250mA - Torque: 800 gf-cm	2	<u>Motor ด้านซ้าย</u> Input enA = 5; Input in1 = 8; Input in2 = 2 <u>Motor ด้านขวา</u> Input enB = 3; Input in3 = 10; Input in4 = 11;	
IR sensor	2	<u>ด้านซ้าย</u> Input irSensor1=7 <u>ด้านขวา</u> Input irSensor2=12	
MH sensor series	1	Input ircolor1 = 6 Input analogInPin = A0	
Ultrasonic Distance Sensor	2	<u>ด้านหน้า</u> Input trigPin1 = 9 Input echoPin1= 13 <u>ด้านหลัง</u> Input trigPin2 =1 Input echoPin2 = 4	
Pushbutton	1	button = 0	
10 kΩ Resistor	1	ใช้ต่อกับสวิตช์เพื่อ ป้องกันกระแสที่มากเกินไป (ไม่ให้สวิตช์ขาด)	

## Car Body Design

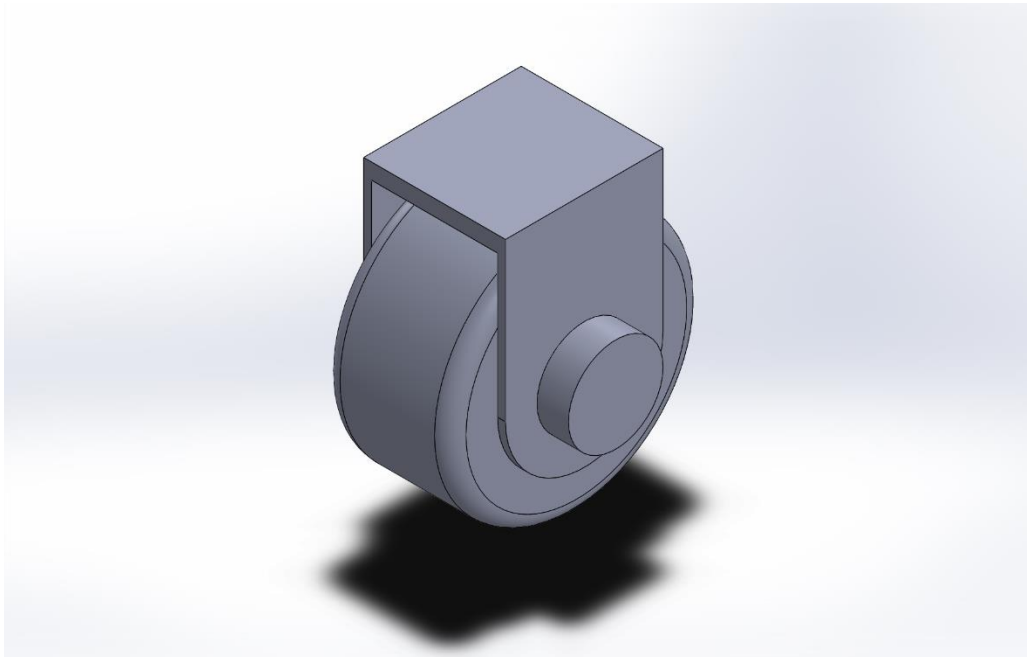
### Dimension Drawing

แบบจำลองตัวรถด้วยโปรแกรม SolidWorks โดยใช้ขนาดจริงตามโจทย์ (10x10 cm.)

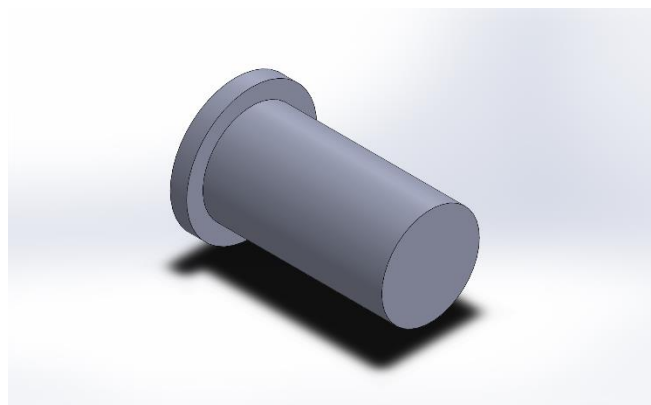
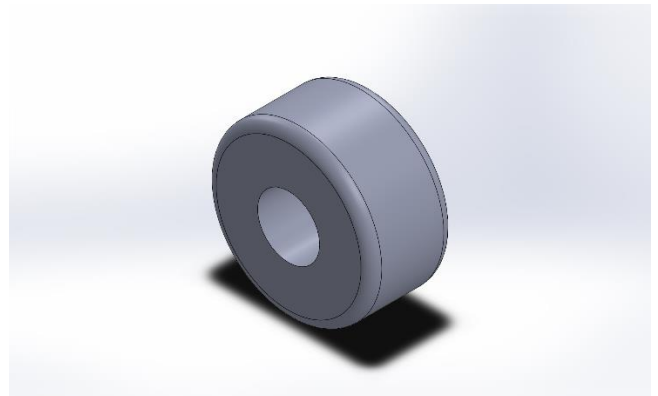
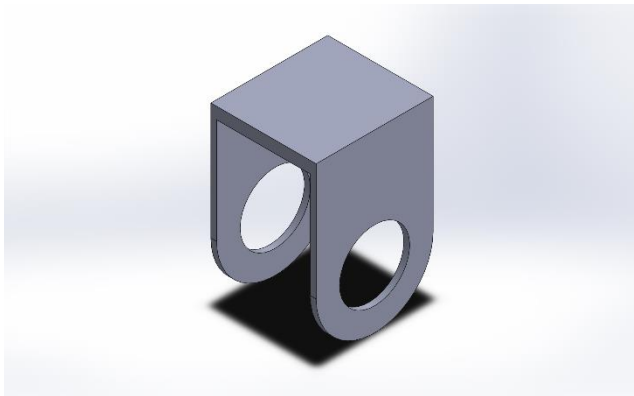


Top BodyBottom Body

## Front Wheel Design

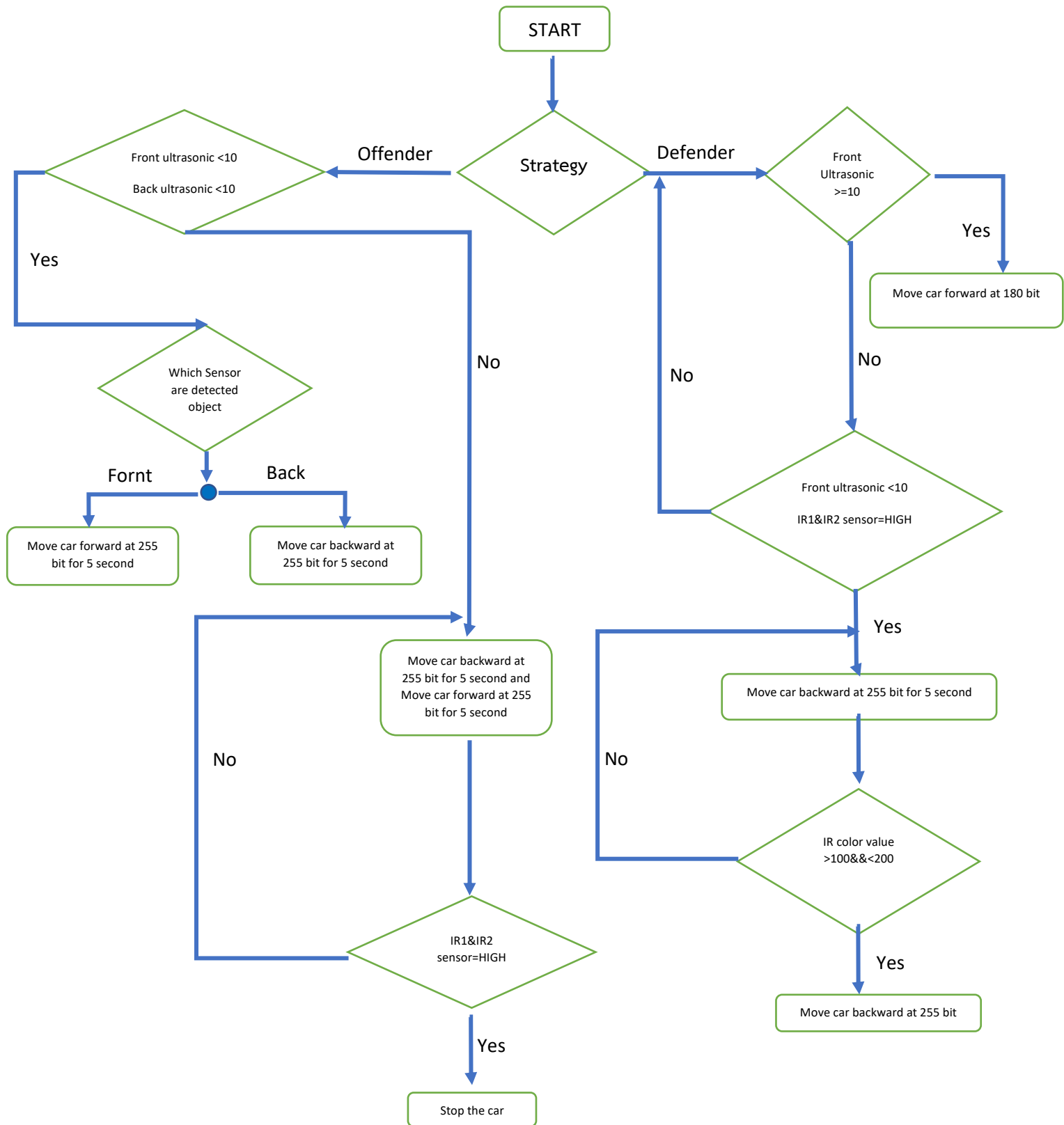


### Part of Front Wheel



## โปรแกรม

## Flow Chart



## Code

```

#include
<analogWrite.h>

//sensor ir
const int irSensor1=7;
const int irSensor2=12;
// ir color
const int ircolor1 = 6;
const int button = 0; //ir color
// infared detected color
int analogInPin = A0;
int ledValue = 0;
// Motor A
int enA = 5;
int in1 = 8;
int in2 = 2;
// Motor B
int enB = 3;
int in3 = 10;
int in4 = 11;
// ultra sensor defines pins numbers
//front
const int trigPin1 = 9;
const int echoPin1= 13;
//back
const int trigPin2 =1 ;
const int echoPin2 = 4;
//value holder
int duration1, distance1;
int duration2, distance2;
bool reading;//push button state
int ctr=0;
int sensorValue = 0;

void setup() {
// pinMode(Sensor,INPUT);
Serial.begin(9600);
// Set all the motor control pins to outputs
//motor
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

```

```

//sensor
pinMode(irSensor1, INPUT); //left side
pinMode(irSensor2, INPUT); //right side
pinMode(ircolor1, INPUT); //digital
pinMode(trigPin1, OUTPUT); //front
pinMode(trigPin2, OUTPUT); //back
pinMode(echoPin1, INPUT); //front
pinMode(echoPin2, INPUT); //back
//pushbutton
pinMode(button, INPUT);
}
//wheel
void backwardwheel()
{
    // This function will run the motors in both directions at a fixed
    speed
    // Turn on motor A
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // Set speed to 255
    analogWrite(enA, 255);
    // Turn on motor B
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    // Set speed to 255
    analogWrite(enB, 255);
}
void slowlyforwardwheel()
{
    // This function will run the motors in both directions at a fixed
    speed
    // Turn on motor A
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // Set speed to 180
    analogWrite(enA, 180);
    // Turn on motor B
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // Set speed to 180
    analogWrite(enB, 180);
}
void forwardwheel()
{
    // This function will run the motors in both directions at a fixed
    speed

```

```

// Turn on motor A
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
// Set speed to 255
analogWrite(enA, 255);
// Turn on motor B
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
// Set speed to 255
analogWrite(enB, 255);
}
void stopwheel()
{
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

void defender(){
    if (distance1 <= 10&& distance1 >= 0 )
    {
        forwardwheel();
    }
    else if(distance2 <= 10&& distance2 >= 0 )
    {
        backwardwheel();
    }
    //ir sensor face enemy
    else if
(digitalRead(irSensor1)==HIGH|digitalRead(irSensor2)==HIGH)
    {
        stopwheel();
    }
    else {
        for (int i = 0; i <= 5; i++)
        {
            forwardwheel();
            delay (1000);
        }
        for (int j = 0; j <= 5; j++)
        {
            backwardwheel();
            delay (1000);
        }
    }
}

```



```

    }

    void offender(){
        if (digitalRead(irSensor1)==HIGH||digitalRead(irSensor2)==HIGH)
        {
            for (int j = 0; j <= 5; j++)
            {
                backwardwheel();
                delay (1000);
            }
        }
        //front ultrasonic not detect
        else if(distance1 >= 10 )
        {
            slowlyforwardwheel();
        }
        //ir sensor face enemy
    }

    void loop() {
        // Duration will be the input pulse width and distance will be the
        distance to the obstacle in centimeters
        // Output pulse with 1ms width on trigPin
        digitalWrite(trigPin1, HIGH);
        delay(1);
        digitalWrite(trigPin1, LOW);
        // Measure the pulse input in echo pin
        duration1 = pulseIn(echoPin1, HIGH);
        // Distance is half the duration divided by 29.1 (from datasheet)
        distance1= (duration1/2) / 29.1;
        digitalWrite(trigPin2, HIGH);
        delay(1);
        digitalWrite(trigPin2, LOW);
        // Measure the pulse input in echo pin
        duration2 = pulseIn(echoPin2, HIGH);
        // Distance is half the duration divided by 29.1 (from datasheet)
        distance2 = (duration2/2) / 29.1;
        //win condition
        sensorValue = analogRead(analogInPin);
        // map it to the range of the analog out:
        // change the analog out value:
        //analogWrite(analogOutPin, outputValue);
    }

```

```
    if (sensorValue>100&&sensorValue<200)
    {
        backwardwheel();
    }

    //strategy select
    reading=digitalRead(button);//reads the push button
    if(reading==HIGH){
        delay(100);
        ctr++;//if button is pressed, counter goes up by one
    }
    if(ctr%2!=0)
    {
        defender();
    }
    else
        offender();
}
```

## เอกสารอ้างอิง

- [1] “บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3” [Online].  
<https://sites.google.com/site/mikhorkhxnthorllexr1/arduino-uno-r3> (1 เมษายน 2563)
- [2] “การใช้งาน IR Infrared Obstacle Avoidance Sensor Module” [Online].  
<https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html>  
 (1 เมษายน 2563)
- [3] “การควบคุมมอเตอร์สองทิศทางด้วยไอซี” [Online].  
<https://www.ioxhop.com/article/99/การควบคุมมอเตอร์ดีซี> (1 เมษายน 2563)
- [4] “L293D Dual H-Bridge Motor Driver for DC” [Online].  
<http://www.robotwinner.com/product/147/l293d-dual-h-bridge-motor>  
 (1 เมษายน 2563)
- [5] “อัลตราโซนิก เซนเซอร์” [Online].  
<https://www.supremelines.co.th/ultrasonic-sensors.html> (1 เมษายน 2563)
- [6] “Arduino tracking sensor TCRT5000” [Online].  
<https://www.myarduino.net/article/95/-arduino-tcrt500> (1 เมษายน 2563)
- [7] “โครงสร้างภาษา C Arduino เบื้องต้น” [Online].  
<https://www.ioxhop.com/article/5/arduino-c-arduino> (2 เมษายน 2563)
- [8] “การเขียนโปรแกรมเบื้องต้นกับ Arduino C++” [Online].  
<https://www.myarduino.net/article/5-arduino-c--arduino> (2 เมษายน 2563)
- [9] “การใช้งานโปรแกรมTinkerCad” [Online].  
<https://eleceasy.com/t/simulation-arduino-tinkercad/341> (3 เมษายน 2563)