



รายงานวิชา Pre-Project รหัสวิชา 01216747

จัดทำโดย

นางสาวปภาณัน แท่งทอง รหัสนักศึกษา 60010573

นางสาวเสาวลักษณ์ หมื่นไกร รหัสนักศึกษา 60011119

นางสาวอมรรัตน์ ปฐมพรสุริยะ รหัสนักศึกษา 60011158

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

ภาคเรียนที่ 2 ปีการศึกษา 2562

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำนำ

ในปัจจุบันภาคอุตสาหกรรมไทยมีการใช้หุ่นยนต์ เพื่อเพิ่มประสิทธิภาพในการทำงานในภาคอุตสาหกรรมการผลิต และธุรกิจบริการ นวัตกรรมหุ่นยนต์สามารถทำงานอย่างต่อเนื่อง มีความถูกต้องแม่นยำ และรวดเร็วกว่ามนุษย์ การจัดทำรายงานนี้มีวัตถุประสงค์เพื่อศึกษาผลการทดลองการทำงานของหุ่นยนต์

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา 01216747 Pre-Project ภาคเรียนที่ 2 ปีการศึกษา 2562 รายงานฉบับนี้จัดทำขึ้นเพื่อออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์

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

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงาน Robot Defense จะเป็นประโยชน์ต่อผู้ที่สนใจทุกท่านและหากมีข้อผิดพลาดประการใด ขออภัยไว้ ณ ที่นี้

สารบัญ

คำนำ	ก
สารบัญ	ข
ปัญหาหรือโจทย์.....	1
แนวทางการแก้ไขปัญหา	1
ขอบเขตของโครงการ	1
แนวคิดและเบื้องหลังที่จำเป็นในการทำโครงการ.....	2
Arduino board (UNO R3).....	2
IR Infrared Obstacle Avoidance Sensor Module	3
Ultrasonic Sensor Module (HC-SR04)	3
IR Reflective Sensor (TCRT5000)	4
H-bridge with L298N mini dual motor driver.....	5
Switch on-off.....	6
วิธีการดำเนินงาน	7
หลักการทำงานใน Arduino	9
การทำงานของกลยุทธ์รุก.....	15
การทำงานของกลยุทธ์รับ.....	18
แผนการดำเนินงาน	20
ความรับผิดชอบของสมาชิกในทีมงาน	21
งบประมาณ.....	21
สิ่งที่คาดว่าจะได้รับ	21
สรุป	22
ข้อเสนอแนะเพิ่มเติม.....	23
เอกสารอ้างอิง.....	24
ภาคผนวก	25

ปัญหาหรือโจทย์

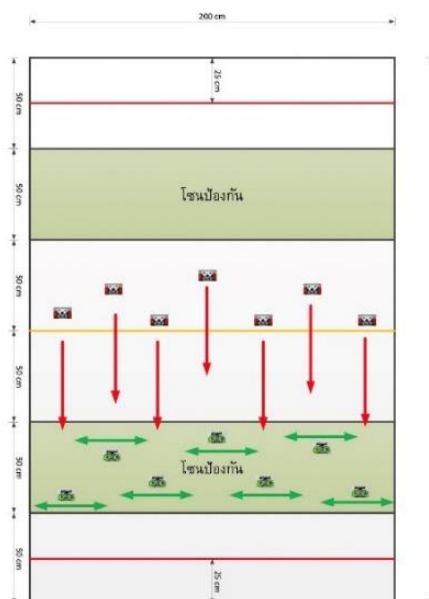
การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่น บอลลุนดำน หรือ เล่นเตย โดยแบ่งเป็นทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้าม จนผ่านเส้นแดง แล้ว กลับมาอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้ ก็จะเป็นฝ่ายชนะในการแข่งขันรอบนั้น หุ่นยนต์ที่ถูกจับได้จะ ถูกตัดออกจากการแข่งขันในรอบนั้น ส่วนทีมรับ จะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ ก็จะถูกตัดออกจากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ ทีมรับจะเป็นฝ่ายชนะ การแข่งขัน ของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

แนวทางการแก้ไขปัญหา

จากโจทย์ปัญหาที่ได้รับ ทางผู้จัดทำได้แบ่งวิเคราะห์และแบ่งการทำงานของโจทย์ปัญหาเป็น 2 ส่วน คือ ฝ่ายรุกและฝ่ายรับ โดยฝ่ายรุกจะเป็นฝ่ายโจมตีเพื่อวิ่งผ่านเขตป้องกันไปยังเส้นสีแดง วิ่งย้อนกลับมาผ่านเขตป้องกันและเข้าเขตเส้นสีเหลืองจึงจะเป็นผู้ชนะ สำหรับฝ่ายรับจะเป็นฝ่ายป้องกัน ป้องกันไม่ให้ฝ่ายตรงข้ามผ่านเขตป้องกันไปยังเส้นสีแดงหรือเส้นสีเหลืองได้ ฝ่ายรับต้องกำจัดฝ่ายตรงข้ามหรือฝ่ายรุกให้หมดจึงจะเป็นผู้ชนะ เมื่อวิเคราะห์แบ่งการทำงานโดยรวมได้แล้ว จึงทำการออกแบบการทำงานของแต่ละกลยุทธ์ โดยเริ่มจากการเขียนแผนผัง Flow Chart จากนั้นนำแผนผังที่ได้มาเขียน code Arduino ต่อไป

ขอบเขตของโครงการ

ทำการออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยมีขนาดของหุ่นยนต์เท่ากับ 10 x 10 เซนติเมตร, การใช้ DC motor เพียง 2 ตัว, การใช้ ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V และมีสนามการแข่งขัน ดังรูปที่ 1



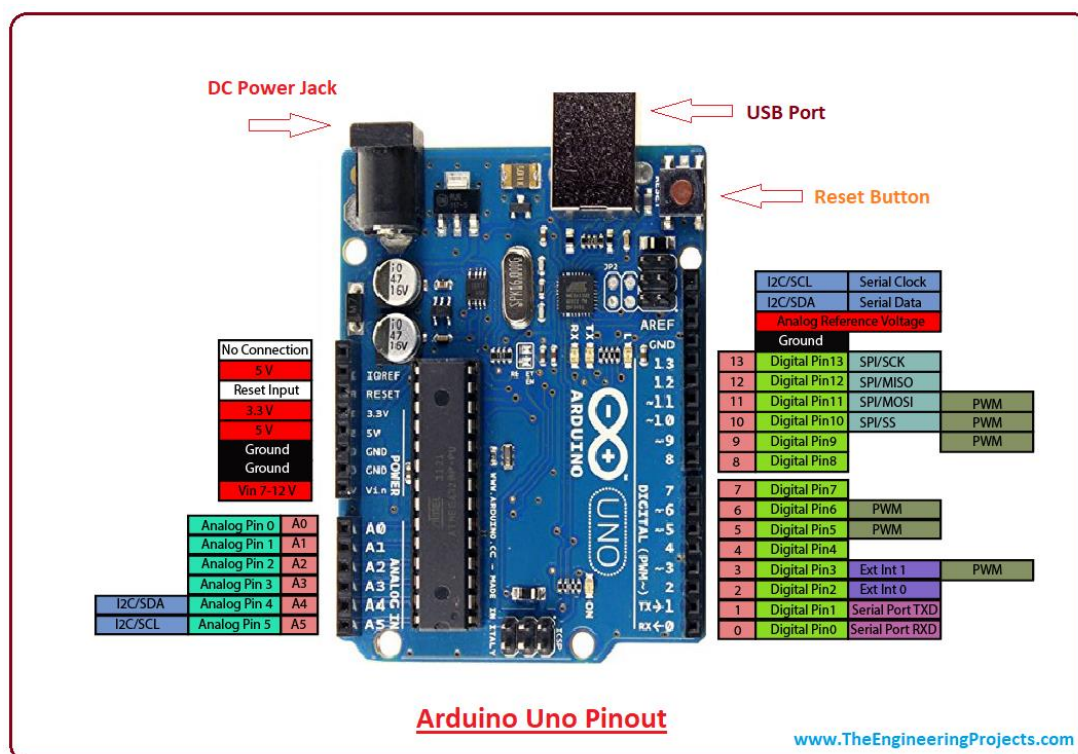
รูปที่ 1 สนามการแข่งขัน

แนวคิดและเบื้องหลังที่จำเป็นในการทำโครงงาน

Arduino board (UNO R3)

Features of the Arduino UNO:

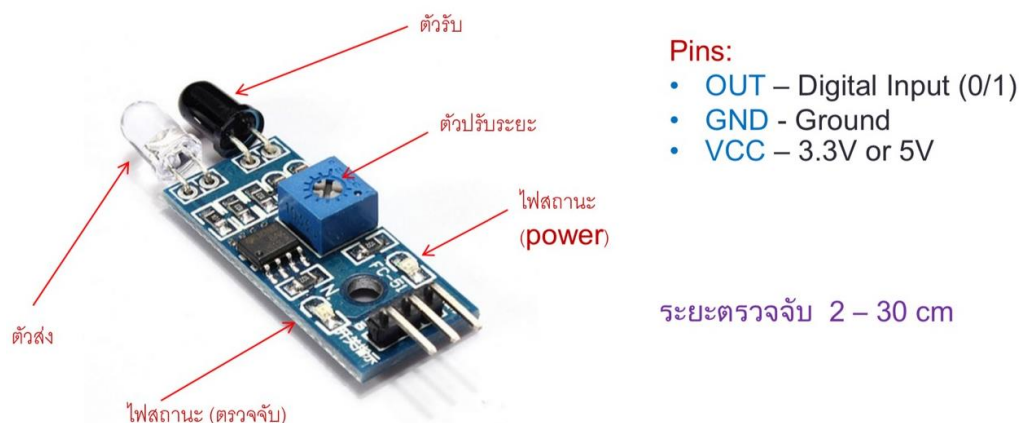
- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz



รูปที่ 2 Arduino Uno Pinout

IR Infrared Obstacle Avoidance Sensor Module

โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง โดยโมดูลนี้จะมีตัวรับและตัวส่ง 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 โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อนและทำให้ทำงานผิดพลาดได้ โมดูลพอร์ตส่งออกเซ็นเซอร์พอร์ต OUT สามารถเชื่อมต่อโดยตรงกับ IO ไมโครคอนโทรลเลอร์โดยตรงสามารถขับ 5V รีเลย์ การเชื่อมต่อ VCC-VCC, GND-GND, OUT-IO สามารถใช้สำหรับโมดูลไฟ 3-5V DC เมื่อเปิดเครื่องไฟแสดงสถานะสีแดงจะสว่างขึ้น

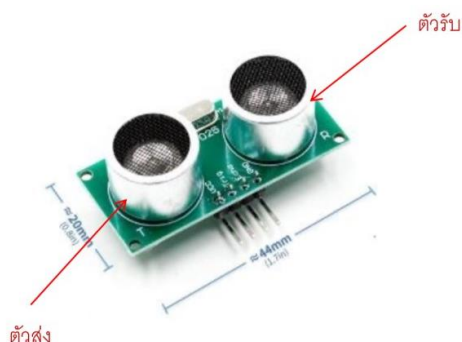


รูปที่ 3 IR Infrared Obstacle Avoidance Sensor Module

Ultrasonic Sensor Module (HC-SR04)

เป็นเซ็นเซอร์โมดูลสำหรับตรวจจับวัตถุและวัดระยะทางแบบไม่สัมผัส [1-2] โดยใช้คลื่นอัลตราโซนิก ซึ่งเป็นคลื่นเสียงความถี่สูงเกินกว่าการได้ยินของมนุษย์ วัดระยะได้ตั้งแต่ 2 – 400 เซนติเมตร หรือ 1 – 156 นิ้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำ เหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติ หรืองานด้านหุ่นยนต์ หลักการทำงาน จะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของคังคาว ตามรูปที่ 1 โดยจะประกอบไปด้วยตัว รับ-ส่ง อัลตราโซนิก ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ เมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาที่ใช้ในการเดินทางไป-กลับ (t) ก็จะสามารถคำนวณหาระยะทางของวัตถุ (S) ได้จาก $S = 346 \times 0.5t$ เพื่อให้การคำนวณหาระยะเป็นไปด้วยความง่าย โมดูลเซ็นเซอร์นี้จึงได้ประมวลผลให้เรียบร้อยแล้ว และส่งผลลัพธ์ของการคำนวณเป็นสัญญาณพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้

การต่อใช้งานโมดูล โมดูลนี้มีจุดต่อใช้งานทั้งหมด 4 จุด การใช้งานบอร์ด STM32F4DISCOVERY การทดลองในเบื้องต้นสามารถต่อวงจรอย่างง่ายได้โดยใช้โปรโตบอร์ดและสายไฟต่อวงจรตามรูปที่ 2 ทั้งนี้ต้องตรวจสอบคุณสมบัติของพอร์ตของไมโครคอนโทรลเลอร์จากดาต้าชีท [3] ว่าสามารถทนระดับแรงดันลอจิก High (5V) ได้ 1.ขา VCC สำหรับต่อแรงดันไฟเลี้ยงไม่เกิน 5V 2.ขา Trig เป็นขาอินพุตรับสัญญาณพัลส์ความกว้าง 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตราโซนิกความถี่ 40KHz ออกสู่อากาศจากตัวส่ง 3.ขา Echo เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจากโมดูลไปยังไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์และคำนวณเป็นระยะทาง 4.ขา GND สำหรับต่อจุดกราวด์ร่วมแรงดันและสัญญาณ



Pins:

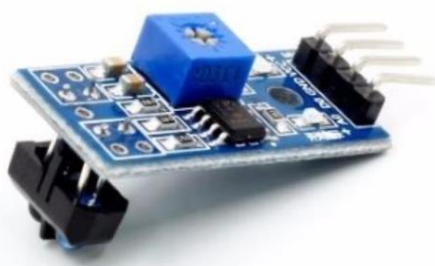
- VCC – 3.3V or 5V
- Trig – Digital output (0/1)
- Echo – Digital Input (0/1)
- GND - Ground

HC-SR04: ระยะตรวจจับ 2 – 400 cm
US-025: ระยะตรวจจับ 2 – 600 cm

รูปที่ 4 Ultrasonic Sensor Module (HC-SR04)

IR Reflective Sensor (TCRT5000)

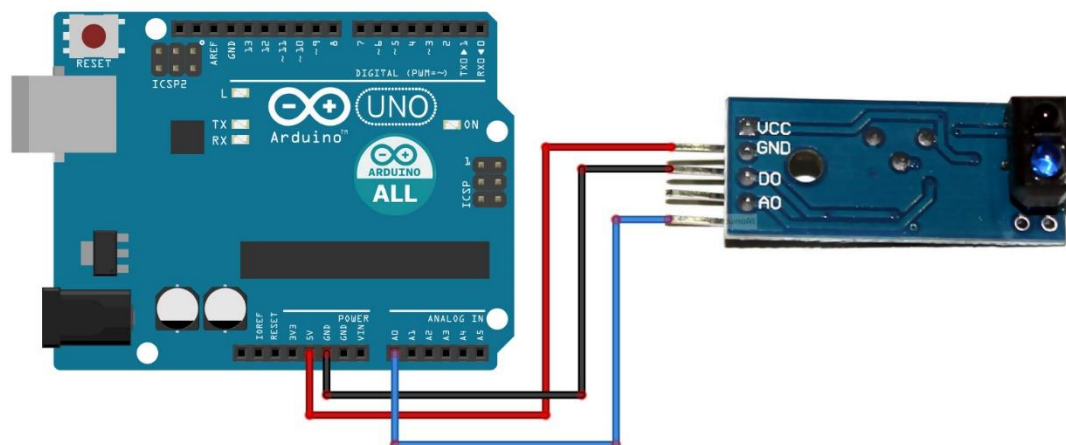
โมดูลอ่านค่าสะท้อนกลับของแสง ใช้ไฟ 3.3-5V เหมาะสำหรับใช้กับ Arduino ให้เอาต์พุตออกมา 2 แบบคือแบบดิจิตอลสามารถปรับค่าที่ต้องการได้ เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการก็จะส่งค่า 1 ออกมา ถ้ายังไม่ถึงระดับก็จะส่งค่า 0 ออกมา และอีกแบบคือเอาต์พุตแบบอนาล็อก อ่านค่าได้เป็นตัวเลข 0-1023 หรือสัญญาณไฟในช่วง 0-5V เซนเซอร์ TCRT5000 เป็นเซนเซอร์ที่ใช้ตรวจจับวัตถุโดยใช้แสงอินฟราเรด โดยจะมี led แบบอินฟราเรดยิงแสงอินฟราเรดออกไป และมีตัวรับแสงอินฟราเรดรับค่าแสงที่สะท้อนกลับมา เมื่อวัตถุอยู่ใกล้จะมีแสงสะท้อนกลับมามากกว่าวัตถุที่อยู่ไกล จึงสามารถนำมาใช้วัดวัตถุผ่าน หรือใช้ตรวจจับเส้นสีขาว/ดำได้ โดยเส้นขาวจะให้แสงสะท้อนกลับมากกว่าสีดำ



Pins:

- Vcc - 3.3 – 5V
- Gnd - Ground
- D0 - Digital output (0/1)
- A0 - Analog output

รูปที่ 5 IR Reflective Sensor (TCRT5000)

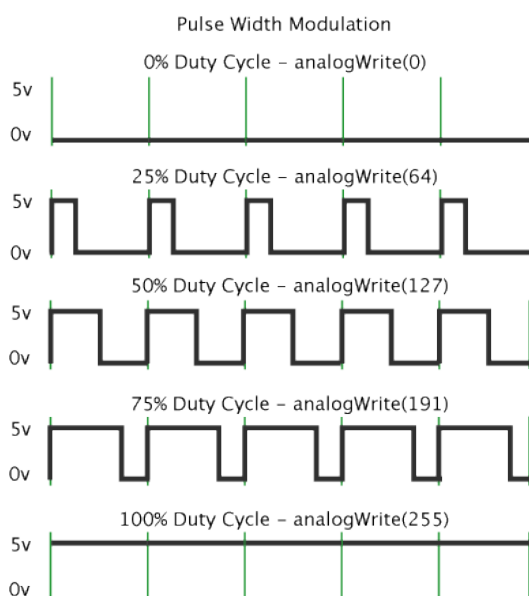


รูปที่ 6 การต่อใช้งานโมดูล

H-bridge with L298N mini dual motor driver

H-Bridge เป็นโมดูลที่ใช้ในการควบคุมความเร็วและทิศทางของมอเตอร์ และยังสามารถนำไปประยุกต์ใช้กับ Project อื่นได้อีกด้วย เช่น ตัวหรี่ไฟในบ้าน หรือที่หรี่ไฟในไฟฉายติดรถ แต่การใช้หลอดไส้ทำให้เราเห็นการกระพริบน้อยกว่าใน Led เป็นวงจรที่สามารถใช้ควบคุมกระแสได้ทั้งขั้วบวกและลบด้วยการควบคุม pulse width modulation (PWM) เป็นการควบคุมแบบ digital ที่มีการนำมาใช้กันมาก โดยส่วนมากเพื่อเป็นการประหยัดพลังงานและ สามารถควบคุม Out Put ได้ โดยมีการกระตุ้นอย่างต่อเนื่อง ซึ่งทำให้เกิดการสูญเสียพลังงานน้อยมาก กล่าวคือวงจรพวกนี้จะมีการปล่อยการสูญเสียพลังงานน้อยกว่าวงจรอื่น ๆ มาก โดยเฉพาะการควบคุมโวลต์หลอดกระแส out put

PWM คือเทคนิคการส่งสัญญาณแบบสวิต หรือ ส่งค่าดิจิทัล 0-1 โดยให้สัญญาณความถี่คงที่ การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำ ที่ต่างกัน ก็จะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิต ต่างกัน



รูปที่ 7 Duty Cycle

สำหรับโมดูล PWM ของ Arduino มีความละเอียด 8 bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณแบบดิจิตอล จะได้ 0 ถึง 255 ซึ่งเราสามารถเทียบสัดส่วนค่านวนจากเลขจริง เป็น เลขทางดิจิตอลได้



รูปที่ 8 H-bridge with L298N mini dual motor driver

Switch on-off

สวิตช์โยก (Toggle Switch) สวิตช์โยก คือ สวิตช์ที่ใช้ปิดเปิดวงจรด้วยการโยกขึ้น-ลง มีตั้งแต่ 2 ขา ถึง 6 ขา แบ่งตามลักษณะและโครงสร้างได้ดังนี้ สวิตช์ทางเดียว ขั้วเดียว (Single Pole Single Throw) สวิตช์แบบนี้เรียกละย่อว่า SPST ปกติมี 2 ขา ใช้ทำหน้าที่ปิดเปิดวงจร, สวิตช์ทางเดียว 2 ขั้ว (Double Pole Single Throw) สวิตช์แบบนี้เรียกละย่อว่า DPST มีขาใช้งาน 4 ขา ใช้ทำหน้าที่เปิด - ปิด วงจรได้ 2 ชุดพร้อมกัน, สวิตช์ 2 ทาง ขั้วเดียว (Single Pole Double Throw) สวิตช์แบบนี้เรียกละย่อว่า SPDT มีขาใช้งาน 2 ขา ใช้ทำหน้าที่เปิด - ปิดวงจร การใช้ขา 2 ขาหรือจะใช้เป็นสวิตช์เลือก 2 ทางก็ได้, สวิตช์ 2 ทาง 2 ขั้ว (Double Pole Double Throw) สวิตช์แบบนี้เรียกละย่อว่า DPDT มีขาใช้งาน 6 ขา ใช้ทำหน้าที่เปิด - ปิดวงจร 2 ชุดพร้อมกัน โดยใช้ขา 4 ขา และสามารถใช้อย่างแทนสวิตช์แบบทางเดียว 2 ขั้วได้ด้วย

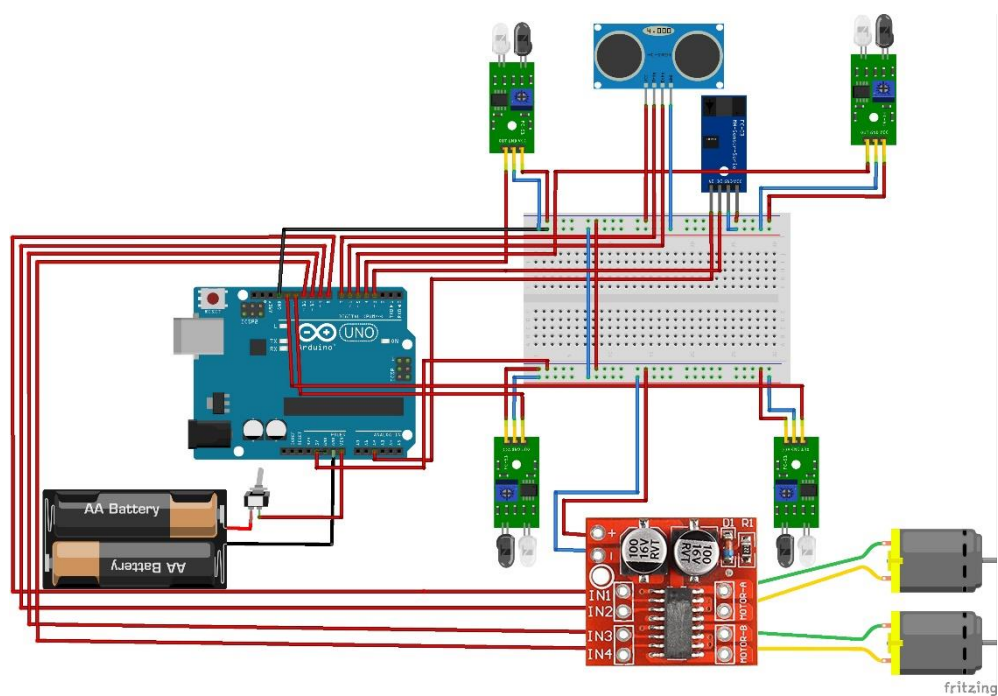


รูปที่ 9 Switch on-off

วิธีการดำเนินงาน



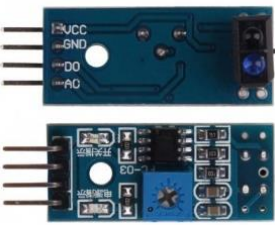
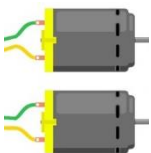
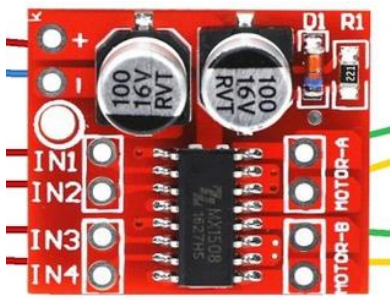


อุปกรณ์ของหุ่นยนต์มีดังนี้

1. IR Infrared Obstacle Avoidance Sensor Module จำนวน 4 ตัว
2. Ultrasonic Sensor Module (HC-SR04) จำนวน 1 ตัว
3. IR Reflective Sensor (TCRT5000) จำนวน 1 ตัว
4. DC motor จำนวน 2 ตัว
5. H-bridge with L298N mini dual motor driver จำนวน 1 ตัว
6. Battery จำนวน 1 ก้อน
7. Switch on-off จำนวน 1 ตัว



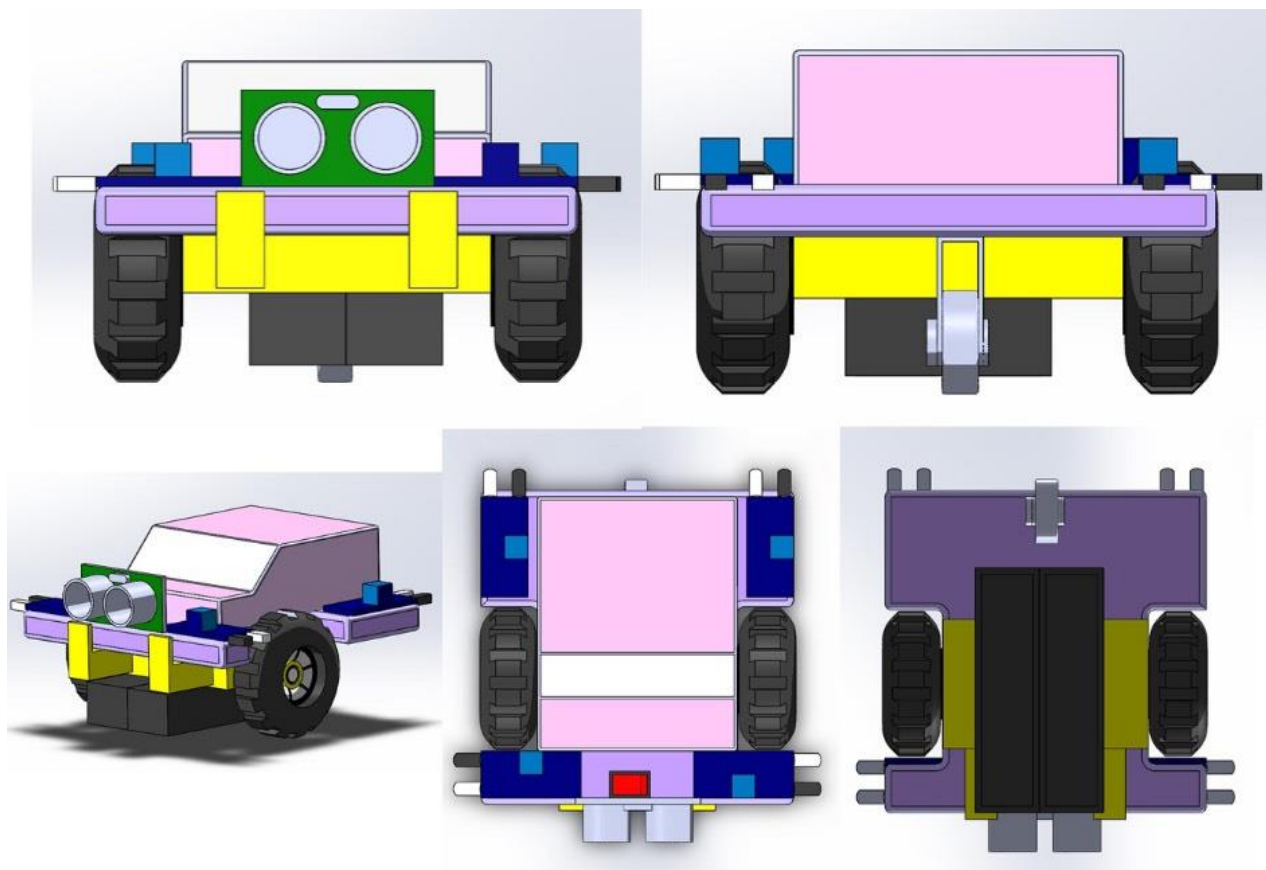
รูปที่ 10 วงจรโดยรวม

โดยมีรายละเอียดการต่อวงจรดังนี้

ชื่ออุปกรณ์	จำนวน	ตัวอย่างอุปกรณ์	การต่อ
IR Infrared Obstacle Avoidance Sensor Module	4		VCC : 5V บนบอร์ด Arduino GND : Ground OUT : D4 , D5 , D12 , D13
Ultrasonic Sensor Module (HC-SR04)	1		VCC : 5V บนบอร์ด Arduino Trig : D7 Echo : D6 GND : Ground
IR Reflective Sensor (TCRT5000)	1		VCC : 5V บนบอร์ด Arduino GND : Ground D0 : D3 A0 : A2
DC motor	2		ขั้วบวกขั้วลบ : MOTOR-A บน L298N ขั้วบวกขั้วลบ : MOTOR-B บน L298N
H-bridge with L298N mini dual motor driver	1		ขั้วบวก : 5V บนบอร์ด Arduino ขั้วลบ : Ground IN1 : D8 IN2 : D9 IN3 : D10 IN4 : D11 MOTOR-A : ขั้วบวก ลบ DC MOTOR A MOTOR-B : ขั้วบวก ลบ DC MOTOR B
ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V	1		ขั้วบวก : ขาบวก switch ขั้วลบ : Ground
Switch On-off	1		ขั้วบวก : ขั้วบวกถ่าน (battery) ขั้วลบ : Vin บนบอร์ด Arduino

* D = Digital บนบอร์ด Arduino

** A = Analog บนบอร์ด Arduino

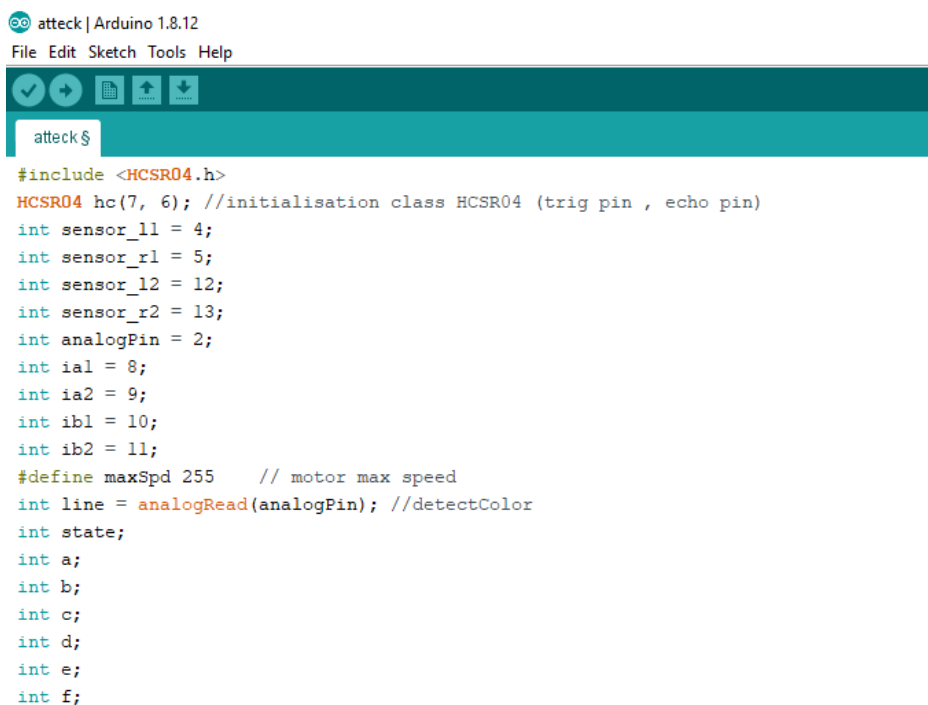


รูปที่ 11 การออกแบบหุ่นยนต์

หลักการทำงานใน Arduino

อธิบายหลักการทำงานของ code Arduino ในงานวิจัยนี้ มีการทำงานหลัก ๆ ทั้งหมด 4 ส่วน คือ การประกาศตัวแปรและกำหนดค่า ฟังก์ชัน setup ฟังก์ชัน loop และฟังก์ชันอื่น ๆ ที่เสริมเข้ามาเพื่อช่วยในการทำงาน

1. การประกาศตัวแปรและกำหนดค่า คือ การกำหนดค่าตัวแปรต่าง ๆ ที่ใช้ใน code และเป็นการกำหนดค่าที่จะเสียบลงบนบอร์ด Arduino ดังรูปที่ 12 นอกจากนี้ยังเป็นการประกาศค่า library ต่าง ๆ ที่เรา install มาใช้ใน code อีกด้วย โดยทางผู้จัดทำได้ install library HCSR04 สำหรับ Ultrasonic sensor มาใช้เพื่อความสะดวกในการเขียน code จากนั้นก็กำหนดค่า trig ให้เป็นขา digital 7 และ echo ให้เป็นขา digital 6 ส่วน IR infrared obstacle avoidance sensor ทั้ง 4 ตัว กำหนดให้เป็นขา digital 4 5 12 และ 13 ตามลำดับ สำหรับ TCRT5000 infrared reflective sensor ผู้จัดทำได้ใช้ขา analog ในการอ่านค่า เพราะจะให้ค่าที่มากกว่า 0 และ 1 แบบ digital โดยต่อเข้ากับขา analog 2 จากนั้นเป็นการกำหนดค่าสำหรับ motor ทั้ง 2 ตัวที่ผู้จัดทำได้ใช้ กำหนดให้เป็นขา digital 8 9 10 และ 11 ตามลำดับ สำหรับตัวแปร line นั้น กำหนดให้ค่าที่อ่านได้จากขา analog 2 มาเก็บไว้ในตัวแปร line นอกเหนือจากนี้คือการกำหนดตัวแปรต่าง ๆ ที่ไว้ใช้ใน code



```

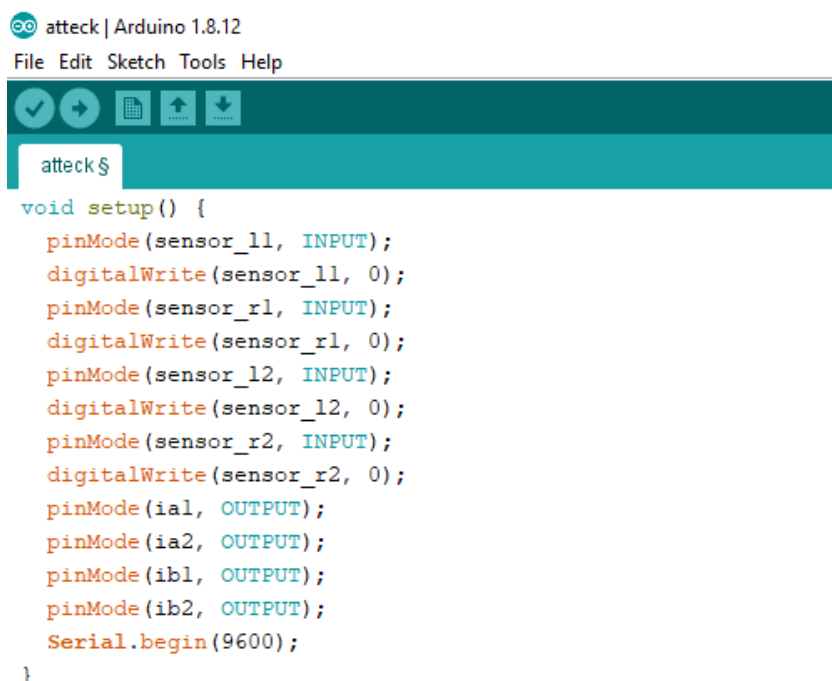
atteck | Arduino 1.8.12
File Edit Sketch Tools Help

atteck $
#include <HCSR04.h>
HCSR04 hc(7, 6); //initialisation class HCSR04 (trig pin , echo pin)
int sensor_l1 = 4;
int sensor_r1 = 5;
int sensor_l2 = 12;
int sensor_r2 = 13;
int analogPin = 2;
int ial = 8;
int ia2 = 9;
int ib1 = 10;
int ib2 = 11;
#define maxSpd 255 // motor max speed
int line = analogRead(analogPin); //detectColor
int state;
int a;
int b;
int c;
int d;
int e;
int f;

```

รูปที่ 12 การประกาศตัวแปรและกำหนดค่า

2. ฟังก์ชัน setup คือฟังก์ชันหลักสำหรับการทำงาน จะทำงานเพียง 1 ครั้งหลังจากได้รับโปรแกรม และจะทำการไปทำงานในฟังก์ชันถัดไป โดย pinMode เป็นการกำหนดค่าตัวแปรว่าเป็น INPUT หรือ OUTPUT ส่วน digitalWrite เป็นการกำหนดให้ค่าตัวแปรที่ประกาศไว้นั้นมีค่าสำหรับเท่าใด เช่น digitalWrite(sensor_l1,0) คือกำหนดให้ตัวแปร sensor_l1 มีค่าเริ่มต้นที่ 0



```

atteck | Arduino 1.8.12
File Edit Sketch Tools Help

atteck $
void setup() {
  pinMode(sensor_l1, INPUT);
  digitalWrite(sensor_l1, 0);
  pinMode(sensor_r1, INPUT);
  digitalWrite(sensor_r1, 0);
  pinMode(sensor_l2, INPUT);
  digitalWrite(sensor_l2, 0);
  pinMode(sensor_r2, INPUT);
  digitalWrite(sensor_r2, 0);
  pinMode(ial, OUTPUT);
  pinMode(ia2, OUTPUT);
  pinMode(ib1, OUTPUT);
  pinMode(ib2, OUTPUT);
  Serial.begin(9600);
}

```

รูปที่ 13 void setup()

3. ฟังก์ชัน loop คือฟังก์ชันการทำงานของฟังก์ชัน loop จะทำงานวนซ้ำไปเรื่อย ๆ เรียกว่า infinite loop การทำงานที่ไม่มีสิ้นสุด โดยผู้จัดทำต้องทำ กลยุทธ์รุก และ กลยุทธ์รับ ที่มีฟังก์ชัน loop แตกต่างกัน จึงจะอธิบายในหัวข้อถัดไป

4. ฟังก์ชันอื่น ๆ เป็นฟังก์ชันที่ผู้จัดทำเขียนขึ้นมาเพื่อช่วยในการเขียน code ในฟังก์ชัน loop ได้ง่ายและสั้นขึ้น โดยมีทั้งหมด 10 ฟังก์ชัน ดังนี้

4.1 void checkSensor เป็นฟังก์ชันสำหรับให้รถตรวจสอบเซนเซอร์ว่าตัวใดทำงานตามเงื่อนไขหรือไม่ ในการทำงานของรถตรวจสอบเซนเซอร์มีทั้งหมด 6 กรณี คือ

1. ถ้าเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายทำงานและเซนเซอร์ด้านหลังมุมขวาทำงาน ให้เก็บค่าเป็น state = a
2. ถ้าเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายทำงานและเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน ให้เก็บค่าเป็น state = b
3. ถ้าเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายไม่ทำงานและเซนเซอร์ด้านหลังมุมขวาทำงาน ให้เก็บค่าเป็น state = c
4. ถ้าเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายไม่ทำงานและเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน ให้เก็บค่าเป็น state = d
5. ถ้าเซนเซอร์ด้านหน้ามุมซ้ายไม่ทำงาน ให้เก็บค่าเป็น state = e
6. ถ้าเซนเซอร์ด้านหน้ามุมขวาไม่ทำงาน ให้เก็บค่าเป็น state = f

หมายเหตุ เมื่อเซนเซอร์ทำงานจะแสดงค่า = 0 ไม่ทำงาน = 1

attek | Arduino 1.8.12
File Edit Sketch Tools Help



attek
}

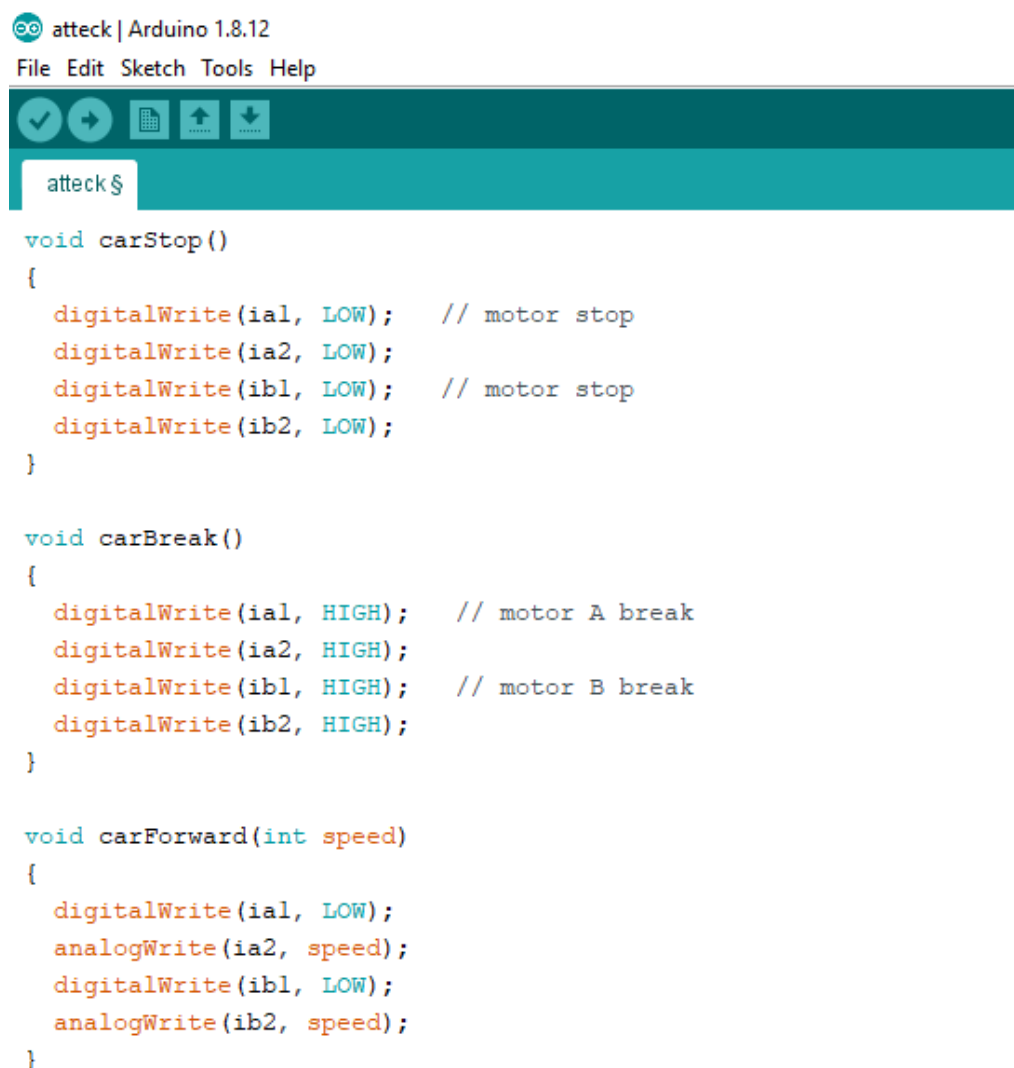
```
void checkSensor()
{
  if (digitalRead(sensor_l1) == 0 && digitalRead(sensor_r1) == 0 && digitalRead(sensor_l2) == 0 && digitalRead(sensor_r2) == 0 ) {
    state = a;
  }
  else if (digitalRead(sensor_l1) == 0 && digitalRead(sensor_r1) == 0 && digitalRead(sensor_l2) == 0 && digitalRead(sensor_r2) == 1) {
    state = b;
  }
  else if (digitalRead(sensor_l1) == 0 && digitalRead(sensor_r1) == 0 && digitalRead(sensor_l2) == 1 && digitalRead(sensor_r2) == 0) {
    state = c;
  }
  else if (digitalRead(sensor_l1) == 0 && digitalRead(sensor_r1) == 0 && digitalRead(sensor_l2) == 1 && digitalRead(sensor_r2) == 1) {
    state = d;
  }
  else if (digitalRead(sensor_l1) == 1) {
    state = e;
  }
  else if (digitalRead(sensor_r1) == 1) {
    state = f;
  }
}
```

รูปที่ 14 void checkSensor()

4.2 void carStop() เป็นฟังก์ชันสำหรับควบคุม motor ให้รถหยุด โดยกำหนดให้ตัวแปรที่กำหนดไว้ในรูป digital เป็นค่า LOW ทั้งหมด

4.3 void carBreak() เป็นฟังก์ชันสำหรับควบคุม motor ให้รถเบรก โดยกำหนดให้ตัวแปรที่กำหนดไว้ในรูป digital เป็นค่า HIGH ทั้งหมด

4.4 void carForward (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถเดินหน้า คือให้ล้อด้านซ้ายและขวามีความเร็วเท่ากัน โดยสามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog



```

attek | Arduino 1.8.12
File Edit Sketch Tools Help

attek$

void carStop()
{
    digitalWrite(ial, LOW); // motor stop
    digitalWrite(ia2, LOW);
    digitalWrite(ib1, LOW); // motor stop
    digitalWrite(ib2, LOW);
}

void carBreak()
{
    digitalWrite(ial, HIGH); // motor A break
    digitalWrite(ia2, HIGH);
    digitalWrite(ib1, HIGH); // motor B break
    digitalWrite(ib2, HIGH);
}

void carForward(int speed)
{
    digitalWrite(ial, LOW);
    analogWrite(ia2, speed);
    digitalWrite(ib1, LOW);
    analogWrite(ib2, speed);
}

```

รูปที่ 15 ฟังก์ชันอื่น ๆ (1)

4.5 void carReverse (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถถอยหลัง คือให้ล้อด้านซ้ายและขวามีความเร็วเท่ากัน โดยสามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog

4.6 void ReverseLeft (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถถอยหลังไปทางด้านซ้าย สามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog ในฟังก์ชันนี้ผู้จัดทำกำหนดให้ล้อด้านซ้ายมีความเร็วเป็น 20% ของความเร็วล้อด้านขวา

4.7 void ReverseRight (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถถอยหลังไปทางด้านขวา สามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog ในฟังก์ชันนี้ผู้จัดทำกำหนดให้ล้อด้านขวามีความเร็วเป็น 20% ของความเร็วล้อด้านซ้าย



```

attek | Arduino 1.8.12
File Edit Sketch Tools Help

attek $

void carReverse(int speed)
{
    digitalWrite(ia2, LOW);
    analogWrite(ial, speed);
    digitalWrite(ib2, LOW);
    analogWrite(ib1, speed);
}

void ReverseLeft(int speed)
{
    digitalWrite(ia2, LOW);
    analogWrite(ial, 0.2 * speed);
    digitalWrite(ib2, LOW);
    analogWrite(ib1, speed);
}

void ReverseRight(int speed)
{
    digitalWrite(ia2, LOW);
    analogWrite(ial, speed);
    digitalWrite(ib2, LOW);
    analogWrite(ib1, 0.2 * speed);
}


```

รูปที่ 16 ฟังก์ชันอื่น ๆ (2)

4.8 void turnLeft (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถเลี้ยวซ้ายคือกำหนดให้ล้อทางด้านขวาทำงานเพียงด้านเดียว โดยสามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog

4.9 void turnRight (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถเลี้ยวขวาคือกำหนดให้ล้อทางด้านซ้ายทำงานเพียงด้านเดียว โดยสามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog

4.10 void carReturn (int speed) เป็นฟังก์ชันสำหรับควบคุม motor ให้รถรีเทิร์นทางด้านซ้ายคือกำหนดให้ล้อทางด้านซ้ายทำงานเพียงด้านเดียวเหมือนการเลี้ยวขวา โดยสามารถกำหนดความเร็วรถได้ใน speed เป็นค่า analog



```

attek | Arduino 1.8.12
File Edit Sketch Tools Help

attek $

void turnLeft(int speed)
{
    digitalWrite(ial, LOW);
    digitalWrite(ia2, LOW);
    digitalWrite(ib1, LOW);
    analogWrite(ib2, speed);
}

void turnRight(int speed)
{
    digitalWrite(ial, LOW);
    analogWrite(ia2, speed);
    digitalWrite(ib1, LOW);
    digitalWrite(ib2, LOW);
}

void carReturn(int speed)
{
    digitalWrite(ial, LOW);
    analogWrite(ia2, speed);
    digitalWrite(ib1, LOW);
    digitalWrite(ib2, LOW);
}

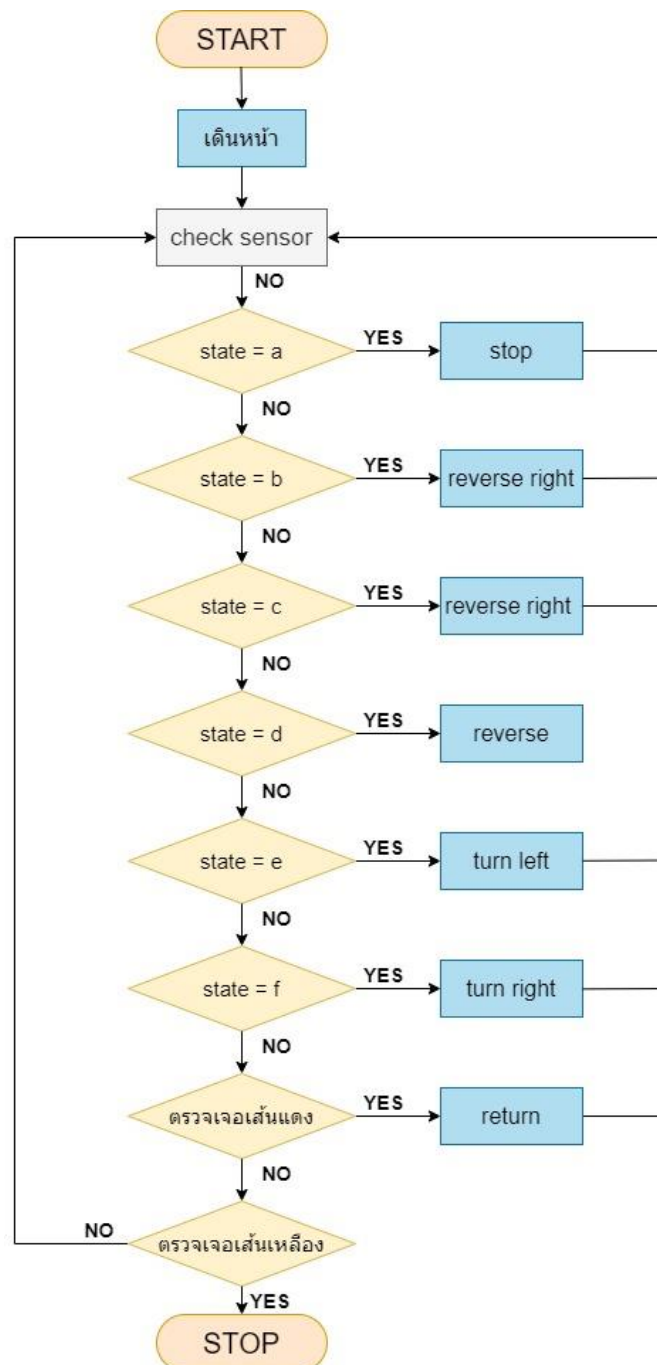
```

รูปที่ 17 ฟังก์ชันอื่น ๆ (3)

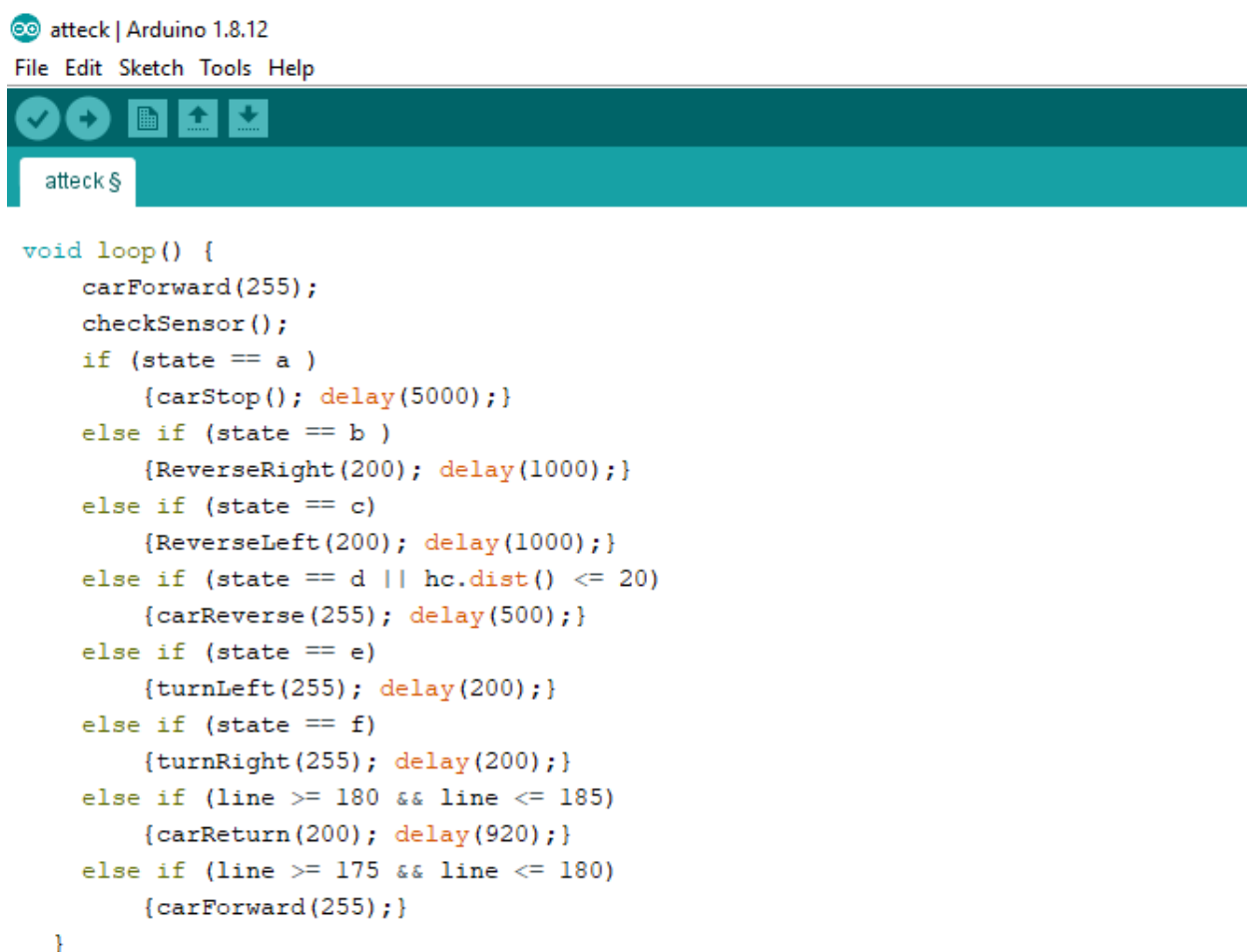
การทำงานของกลยุทธ์รูก

โจทย์ปัญหาของกลยุทธ์รูกคือให้รถเริ่มต้นหลังเส้นเหลือง จากนั้นวิ่งผ่านเขตป้องกันอย่างปลอดภัย โดยไม่ให้ฝ่ายตรงข้ามหรือฝ่ายรับจับได้ วิ่งไปถึงเส้นสีแดงและวิ่งกลับมาผ่านเข้าเขตป้องกันอย่างปลอดภัย จนถึงเส้นสีเหลืองได้นั้น ถือว่าเป็นผู้ชนะ

การออกแบบการทำงานของกลยุทธ์รูก ผู้จัดทำได้ออกแบบการทำงานไว้ ดังรูปที่ 18 เริ่มต้นโดยให้รถเดินทางด้วยความเร็วสูงสุด จากนั้นให้รถตรวจสอบเซนเซอร์ เช็คใน void check Sensor() ดังรูปที่ 14 โดยเงื่อนไขแรกคือค่าตัวแปร state = a หรือไม่ ถ้า ใช่ สั่งให้รถหยุดจากนั้นรถจะเดินทางเพื่อตรวจสอบเซนเซอร์ต่อไป ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ state = b หรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังไปด้านขวาเพื่อหลบฝ่ายตรงข้ามจากการตรวจสอบเซนเซอร์ได้ เพราะเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ state = c หรือไม่ ถ้าใช่สั่งรถถอยหลังไปด้านซ้ายเพื่อหลบฝ่ายตรงข้าม ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ state = d หรือไม่ โดยในการเขียน code นั้นได้เพิ่มเงื่อนไขการตรวจสอบการทำงานของ Ultrasonic sensor ด้วย ถ้าตรวจพบวัตถุด้านหน้าในระยะน้อยกว่าหรือเท่ากับ 20 เซนติเมตร ดังรูปที่ 19 ถ้า ใช่ สั่งรถถอยหลังเพื่อหลบฝ่ายตรงข้ามที่อยู่ด้านหน้าทั้งซ้ายและขวา ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ state = e หรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวซ้ายเพื่อเดินทางไปในทิศที่เซนเซอร์ไม่ตรวจพบวัตถุ ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ state = f หรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวขวาเพื่อเดินทางไปในทิศที่เซนเซอร์ไม่ตรวจพบวัตถุ ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ ตรวจพบเส้นสีแดงหรือไม่ ถ้าใช่สั่งให้รถรีเทิร์นกลับเพื่อกลับไปยังเส้นชัย ถ้า ไม่ ให้ตรวจสอบเงื่อนไข สุดท้ายคือ ตรวจสอบพบเส้นสีเหลือง ถ้า ใช่ สั่งให้รถเดินทางต่อไป แต่ในการเขียน Flow chart เมื่อพบเส้นสีเหลืองหลังจากการรีเทิร์นรถกลับมานั้นคือการชนะ จึงกำหนดให้เป็นการจบ loop การทำงานของกลยุทธ์รูกนั่นเอง



รูปที่ 18 Flow Chart กลยุทธ์รูก



The image shows the Arduino IDE interface. At the top, it says "attek | Arduino 1.8.12" and "File Edit Sketch Tools Help". Below the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The main text area contains the following C++ code:

```

void loop() {
  carForward(255);
  checkSensor();
  if (state == a )
    {carStop(); delay(5000);}
  else if (state == b )
    {ReverseRight(200); delay(1000);}
  else if (state == c)
    {ReverseLeft(200); delay(1000);}
  else if (state == d || hc.dist() <= 20)
    {carReverse(255); delay(500);}
  else if (state == e)
    {turnLeft(255); delay(200);}
  else if (state == f)
    {turnRight(255); delay(200);}
  else if (line >= 180 && line <= 185)
    {carReturn(200); delay(920);}
  else if (line >= 175 && line <= 180)
    {carForward(255);}
}

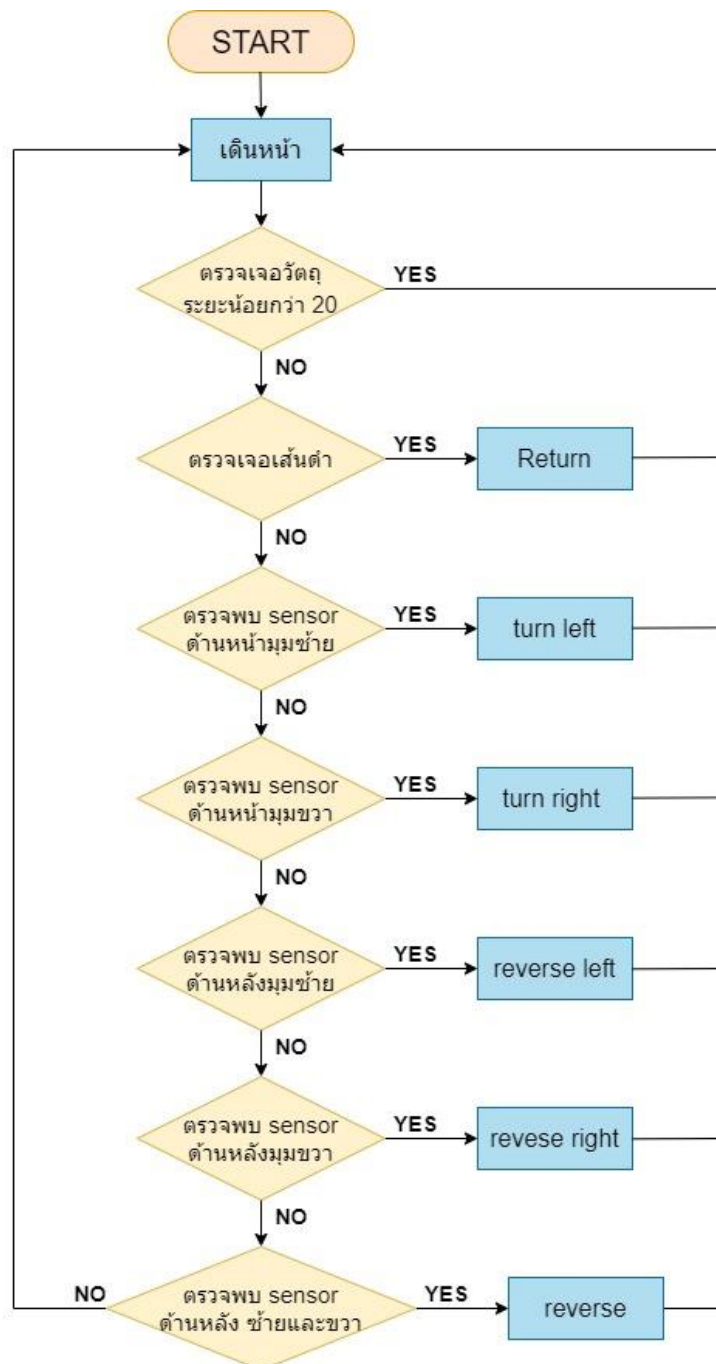
```

รูปที่ 19 code การทำงานของกลยุทธ์รถ

การทำงานของกลุ่ทรรับ

โจทย์ปัญหาของกลุ่ทรรับคือให้รถอยู่ภายในกรอบสีดำในเขตป้องกันเท่านั้น ห้ามเข้านอกเขต และมีฝายตรงข้ามหรือฝายรุ้เข้ามา ต้องจับฝายรุ้ไม่ให้ผ่านเขตป้องกันไปยังเส้นสีแดงได้ ถ้าป้องกันฝายรุ้ทั้ง 7 ทีมได้นั้น ถือว่าเป็นผู้ชนะ

การออกแบบการทำงานของกลุ่ทรรับ ผู้จัดทำได้ออกแบบการทำงานไว้ ดังรูปที่ 20 โดยเริ่มต้นให้รถเดินทาง และถ้าตรวจพบวัตถุด้านหน้าในระยะน้อยกว่าหรือเท่ากับ 20 เซนติเมตร ให้เดินทางไปหารถฝายตรงข้ามในความเร็วสูงสุด ถ้าตรวจสอบว่าไม่พบ ให้ตรวจสอบเงื่อนไขถัดไป คือ TCRT5000 Reflective sensor ตรวจสอบเจอเส้นสีดำหรือไม่ ถ้าใช่สั่งให้รถรีเทิร์น กลับเพื่อไม่ให้รถออกจากเขตป้องกันตามโจทย์ที่ตั้งไว้ ถ้าไม่ให้ตรวจสอบที่เงื่อนไขถัดไป คือ IR infrared sensor ที่ติดอยู่ด้านหน้ามุมซ้ายทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวซ้ายเพื่อวิ่งเข้าหาฝายตรงข้ามทันที ถ้า ไม่ ให้ตรวจสอบต่อว่า IR infrared sensor ที่ติดอยู่ด้านหน้ามุมขวาทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวขวาเพื่อเข้าหาฝายตรงข้าม ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ IR infrared sensor ที่ติดอยู่ด้านหลังมุมซ้ายทำงานหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังทางด้านซ้ายเพื่อเข้าหาฝายตรงข้ามอย่างทันที ถ้า ไม่ ให้ไปตรวจสอบทางด้านหลังมุมขวาวางานหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังไปทางด้านขวา ถ้า ไม่ ให้ตรวจสอบในเงื่อนไขสุดท้ายที่ว่า IR infrared sensor ที่ติดอยู่ด้านหลังมุมซ้ายและขวานั้นทำงานพร้อมกันหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังด้วยความเร็วสูงสุดเพื่อเข้าหาฝายตรงข้ามทันที จากการตรวจสอบถ้าไม่ตรงกับเงื่อนไขที่กล่าวมาเลย รถจะเดินทางเรื่อย ๆ จนกว่าจะเจอการทำงานของเซนเซอร์ว่าตรงกับเงื่อนไขใดหรือไม่ กล่าวคือ กลุ่ทรรับนั้นจะพุ่งเข้าหาฝายตรงข้ามเมื่อตรวจเจอเท่านั้นและจะชนะได้เมื่อกำจัดฝายตรงข้ามได้หมด การทำงานของกลุ่ทรจะทำงานไปเรื่อย ๆ หรือเรียกว่า infinite loop การทำงานที่ไม่มีสิ้นสุด



รูปที่ 20 Flow Chart กลยุทธ์รับ

defender | Arduino 1.8.12
File Edit Sketch Tools Help



```
void loop() {
  bool left1 = digitalRead(sensor_l1);
  bool right1 = digitalRead(sensor_r1);
  bool left2 = digitalRead(sensor_l2);
  bool right2 = digitalRead(sensor_r2);

  carForward(255);
  if (line > 500) //ถ้าเจอเส้นดำ //ต้องลองทดสอบ
    {carReverse(255); delay(200); carReturn(200); delay(920);}
  else if (hc.dist() <= 20)
    {carForward(255); delay(1000);}
  else if (left1 == 0 )
    {turnLeft(200); delay(500);}
  else if (right1 == 0 )
    {turnRight(200); delay(500);}
  else if (left2 == 0)
    {ReverseLeft(200); delay(500); carReverse(255); delay(500);}
  else if (right2 == 0)
    {ReverseRight(200); delay(500); carReverse(255); delay(500);}
  else if (left2 == 0 && right2 == 0)
    {carReverse(255); delay(500);}
}
```

รูปที่ 21 code การทำงานของกลยุทธรับ

แผนการดำเนินงาน

รายการ	W 1-2	W 3-4	W 5-6	W 7-8	W 9-10	W 11-12	W 13-14	W 15
จัดกลุ่มและแบ่งหน้าที่	↔							
วางแผนกลยุทธ์และทำproject proposal		↔						
ออกแบบรูปร่างรถ			↔	↔				
ทำตัวรถ				↔	↔	↔		
เขียนโค้ด					↔	↔	↔	
ปรับปรุงตัวรถและการทำงาน					↔	↔	↔	
จัดทำรูปเล่ม							↔	↔

ความรับผิดชอบของสมาชิกในทีมงาน

1. นางสาวปภาณัน แท่งทอง	60010573	ต่อวงจร, รูปเล่ม
2. นางสาวเสาวลักษณ์ หมีนไกร	60011119	เขียนโค้ด, รูปเล่ม
3. นางสาวอมรรัตน์ ปฐมพรสุริยะ	60011158	ออกแบบตัวรถ, รูปเล่ม

งบประมาณ

การออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์มีอุปกรณ์และมีงบประมาณดังนี้

1. Ultrasonic Sensor Module (HC-SR04) 1 ตัวเท่ากับ 38 บาท
2. IR Reflective Sensor (TCRT5000) 1 ตัวเท่ากับ 35 บาท
3. DC motor 2 ตัว ตัวละ 70 เท่ากับ 140บาท
4. H-bridge with L298N mini dual motor driver 1 ตัว เท่ากับ 30บาท
5. Battery 1 ก้อน เท่ากับ 74บาท
6. Switch On-Off 1 ตัว เท่ากับ 7บาท
7. ARDUINO UNO R3 1ตัว เท่ากับ 120บาท

โดยงบประมาณทั้งหมดเท่ากับ 444 บาท

สิ่งที่คาดว่าจะได้รับ

หุ่นยนต์ต้นแบบ สามารถนำไปพัฒนาต่อยอดใช้ในกระบวนการผลิตสินค้าและบริการได้ เช่น ใช้ในการขนส่งภายในโรงงาน ตรวจสอบระบบตำแหน่งต่างๆ

เรียนรู้ เข้าใจ การเขียนโค้ดโดยใช้โปรแกรม Arduino IDE เป็นโปรแกรมที่ใช้ง่าย สามารถเรียนรู้ได้ด้วยตัวเอง และนำไปใช้ในอนาคตได้

การใช้โปรแกรม SOLIDWORKS ทำให้สามารถมองภาพรวมหุ่นยนต์ต้นแบบได้เป็นอย่างดี ใช้ความคิดสร้างสรรค์ปรับแต่งให้มีความเหมาะสม และขนาดที่ถูกต้องตามที่กำหนด

การพูดคุยกันผ่านทาง GitHub ทุกคนสามารถเข้าถึงข้อมูลของกันและกันได้ สามารถเห็นได้ว่าเพื่อนในกลุ่มปรับแก้ตรงไหน เห็นด้วยหรือไม่เห็นด้วยอย่างไร ความคืบหน้าของงาน ถือได้ว่าเป็นอีกช่องทางหนึ่งมีประโยชน์ต่อยุคปัจจุบัน

สรุป

จากโจทย์ปัญหาที่ได้รับ ทางผู้จัดทำได้แบ่งวิเคราะห์และแบ่งการทำงานของโจทย์ปัญหาเป็น 2 ส่วน คือ ฝ่ายรุกและฝ่ายรับ โดยฝ่ายรุกจะเป็นฝ่ายโจมตีเพื่อวิ่งผ่านเขตป้องกันไปยังเส้นสีแดง วิ่งย้อนกลับมาผ่านเขตป้องกันและเข้าเขตเส้นสีเหลืองจึงจะเป็นผู้ชนะ สำหรับฝ่ายรับจะเป็นฝ่ายป้องกัน ป้องกันไม่ให้ฝ่ายตรงข้ามผ่านเขตป้องกันไปยังเส้นสีแดงหรือเส้นสีเหลืองได้ ฝ่ายรับต้องกำจัดฝ่ายตรงข้ามหรือฝ่ายรุกให้หมดจึงจะเป็นผู้ชนะ เมื่อวิเคราะห์แบ่งการทำงานโดยรวมได้แล้ว จึงทำการออกแบบการทำงานของแต่ละกลยุทธ์ดังนี้

กลยุทธ์รุก

การออกแบบการทำงานของกลยุทธ์รุก ผู้จัดทำได้ออกแบบการทำงานไว้ เริ่มต้นโดยให้รถเดินทางด้วยความเร็วสูงสุด จากนั้นให้รถตรวจสอบเซนเซอร์ โดยเงื่อนไขแรกคือเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายทำงานและเซนเซอร์ด้านหลังมุมขวาทำงานหรือไม่ ถ้า ใช่ สั่งให้รถหยุดจากนั้นรถจะเดินทางเพื่อตรวจสอบเซนเซอร์ต่อไป ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไปคือเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายทำงานและเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน หรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังไปด้านขวาเพื่อหลบฝ่ายตรงข้ามจากการตรวจสอบเซนเซอร์ได้ เพราะเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไปคือเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายไม่ทำงานและเซนเซอร์ด้านหลังมุมขวาทำงานหรือไม่ ถ้าใช่สั่งรถถอยหลังไปด้านซ้ายเพื่อหลบฝ่ายตรงข้าม ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไปคือเซนเซอร์ด้านหน้ามุมซ้ายทำงานและเซนเซอร์ด้านหน้ามุมขวาทำงานและเซนเซอร์ด้านหลังมุมซ้ายไม่ทำงานและเซนเซอร์ด้านหลังมุมขวาไม่ทำงาน หรือไม่ โดยในการเขียน code นั้นได้เพิ่มเงื่อนไขการตรวจสอบการทำงานของ Ultrasonic sensor ด้วย ถ้าตรวจพบวัตถุด้านหน้าในระยะน้อยกว่าหรือเท่ากับ 20 เซนติเมตร ถ้า ใช่ สั่งรถถอยหลังเพื่อหลบฝ่ายตรงข้ามที่อยู่ด้านหน้าทั้งซ้ายและขวา ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไปคือถ้าเซนเซอร์ด้านหน้ามุมซ้ายไม่ทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวซ้ายเพื่อเดินทางไปในทิศที่เซนเซอร์ไม่ตรวจพบวัตถุ ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไปเซนเซอร์ด้านหน้ามุมขวาไม่ทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวขวาเพื่อเดินทางไปในทิศที่เซนเซอร์ไม่ตรวจพบวัตถุ ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ ตรวจพบเส้นสีแดงหรือไม่ ถ้าใช่สั่งให้รถรีเทิร์นกลับเพื่อกลับไปยังเส้นชัย ถ้า ไม่ ให้ตรวจสอบเงื่อนไขสุดท้ายคือ ตรวจพบเส้นสีเหลือง ถ้า ใช่ สั่งให้รถเดินทางต่อไป เมื่อพบเส้นสีเหลืองหลังจากการรีเทิร์นรถกลับมานั้นคือการชนะ

กลยุทธ์รับ

การออกแบบการทำงานของกลยุทธ์รับ ผู้จัดทำได้ออกแบบการทำงาน โดยเริ่มต้นให้รถเดินทางและถ้าตรวจพบวัตถุด้านหน้าในระยะน้อยกว่าหรือเท่ากับ 20 เซนติเมตร ให้เดินทางไปหารถฝ่ายตรงข้ามในความเร็วสูงสุด ถ้าตรวจสอบว่าไม่พบ ให้ตรวจสอบเงื่อนไขถัดไป คือ TCRT5000 Reflective sensor ตรวจสอบเจอเส้นสีดำหรือไม่ ถ้าใช่สั่งให้รถรีเทิร์น กลับเพื่อไม่ให้รถออกจากเขตป้องกันตามโจทย์ที่ตั้งไว้ ถ้า ไม่ ให้ตรวจสอบที่เงื่อนไขถัดไป คือ IR infrared sensor ที่ติดอยู่ด้านหน้ามุมซ้ายทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวซ้ายเพื่อวิ่งเข้าหาฝ่ายตรงข้ามทันที ถ้า ไม่ ให้ตรวจสอบต่อว่า IR infrared sensor ที่ติดอยู่ด้านหน้ามุม

ขวทำงานหรือไม่ ถ้า ใช่ สั่งให้รถเลี้ยวขวาเพื่อเข้าหาฝ่ายตรงข้าม ถ้า ไม่ ให้ตรวจสอบเงื่อนไขถัดไป คือ IR infrared sensor ที่ติดอยู่ด้านหลังมุมซ้ายทำงานหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังทางด้านซ้ายเพื่อเข้าหาฝ่ายตรงข้ามอย่างทันที ถ้า ไม่ ให้ไปตรวจสอบทางด้านหลังมุมขวาว่าทำงานหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังไปทางด้านขวา ถ้า ไม่ ให้ตรวจสอบในเงื่อนไขสุดท้ายที่ว่า IR infrared sensor ที่ติดอยู่ด้านหลังมุมซ้ายและขวานั้นทำงานพร้อมกันหรือไม่ ถ้า ใช่ สั่งให้รถถอยหลังด้วยความเร็วสูงสุดเพื่อเข้าหาฝ่ายตรงข้ามทันที จากการตรวจสอบถ้าไม่ตรงกับเงื่อนไขที่กล่าวมาเลย รถจะเดินหน้าเรื่อย ๆ จนกว่าจะเจอการทำงานของเซนเซอร์ว่าตรงกับเงื่อนไขใดหรือไม่ กล่าวคือ กลยุทธ์รับนั้นจะพุ่งเข้าหาฝ่ายตรงข้ามเมื่อตรวจเจอเท่านั้นและจะชนะได้เมื่อกำจัดฝ่ายตรงข้ามได้หมด การทำงานของกลยุทธ์จะทำงานไปเรื่อย ๆ หรือเรียกว่า infinite loop การทำงานที่ไม่มีสิ้นสุด โดยมีขอบเขตในการออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยมีขนาดของหุ่นยนต์เท่ากับ 10 x 10 เซนติเมตร, การใช้ DC motor เพียง 2 ตัว, การใช้ ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V และมีสนามการแข่งขันดังรูปที่ 1

จากการสร้างแบบจำลองรถหุ่นยนต์เพื่อนำมาแข่งขัน มีการวางแผนการดำเนินงานร่วมกัน กำหนดระยะเวลาในการดำเนินงาน การใช้ความคิดสร้างสรรค์ในการออกแบบหุ่นยนต์ การเขียนโปรแกรมสั่งหุ่นยนต์ รวมไปถึงการวางแผนกลยุทธ์เพื่อการแข่งขัน จากการศึกษาและลงมือทำทำให้ได้เรียนรู้และเข้าใจเกี่ยวกับการทำงานของหุ่นยนต์ โดยใช้โปรแกรม Arduino สั่งการ นำไปสู่การออกแบบสร้างหุ่นยนต์โดยใช้โปรแกรม Solidwork สามารถใช้งานได้ตามโจทย์ และแผนที่วางไว้ ทั้งกลยุทธ์รุกและรับ ทั้งนี้เพื่อประสิทธิภาพของหุ่นยนต์ ควรลงมือปฏิบัติสร้างหุ่นยนต์ขึ้นมาจริง และดูปัญหาและข้อผิดพลาด นำมาแก้ไข รวมทั้งศึกษาเพิ่มเติมเพื่อให้หุ่นยนต์มีการทำงานที่แม่นยำมากยิ่งขึ้น

ข้อเสนอแนะเพิ่มเติม

1. ใช้มอเตอร์มากกว่า 2 ตัว ในการขับเคลื่อน
2. เพิ่มขนาดของหุ่นยนต์ให้มีขนาดมากกว่า 10 x 10 cm.
3. ใส่เซนเซอร์ในการตรวจจับที่ดีกว่าเดิม และเพิ่มจำนวนเซนเซอร์ให้มีการตรวจจับทิศทางมากกว่านี้ (จากขนาดรถจึงทำให้ใส่ได้จำกัด)
4. เพิ่ม Buzzer ในวงจรเพื่อเพิ่ม option ให้กับตัวหุ่นยนต์ (จากขนาดรถจึงทำให้พื้นที่มีจำกัด)
5. เปลี่ยนจากการอัปโหลด code ทุกครั้งเมื่อเปลี่ยนกลยุทธ์เป็นการใช้ค่าตัวแปรในการเปลี่ยนค่าแทน
6. ใช้การควบคุมหุ่นยนต์ผ่านอุปกรณ์บังคับได้ เพื่อเพิ่มประสิทธิภาพในการโจมตีหรือป้องกัน

เอกสารอ้างอิง

[1] “IR Infrared Obstacle Avoidance Sensor Module”

<https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html> (30 มีนาคม 2563)

[2] “Ultrasonic Sensor Module (HC-SR04)”

<http://aimagin.com/blog/%E0%B8%AD%E0%B8%B1%E0%B8%A5%E0%B8%95%E0%B8%A3%E0%B8%B2%E0%B9%82%E0%B8%8B%E0%B8%99%E0%B8%B4%E0%B8%81-hc-sr04/?lang=th> (30 มีนาคม 2563)

[3] “IR Reflective Sensor (TCRT5000)”

<https://www.arduinoall.com/product/698/tcrt5000-infrared-reflectance-obstacle-avoidance-line-tracking-sensor-%E0%B9%80%E0%B8%8B%E0%B8%99%E0%B9%80%E0%B8%8B%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B9%80%E0%B8%8A%E0%B9%87%E0%B8%84%E0%B8%AA%E0%B8%B4%E0%B9%88%E0%B8%87%E0%B8%81%E0%B8%B5%E0%B8%94%E0%B8%82%E0%B8%A7%E0%B8%B2%E0%B8%87-%E0%B9%80%E0%B8%AA%E0%B9%89%E0%B8%99%E0%B8%82%E0%B8%B2>(30 มีนาคม 2563)

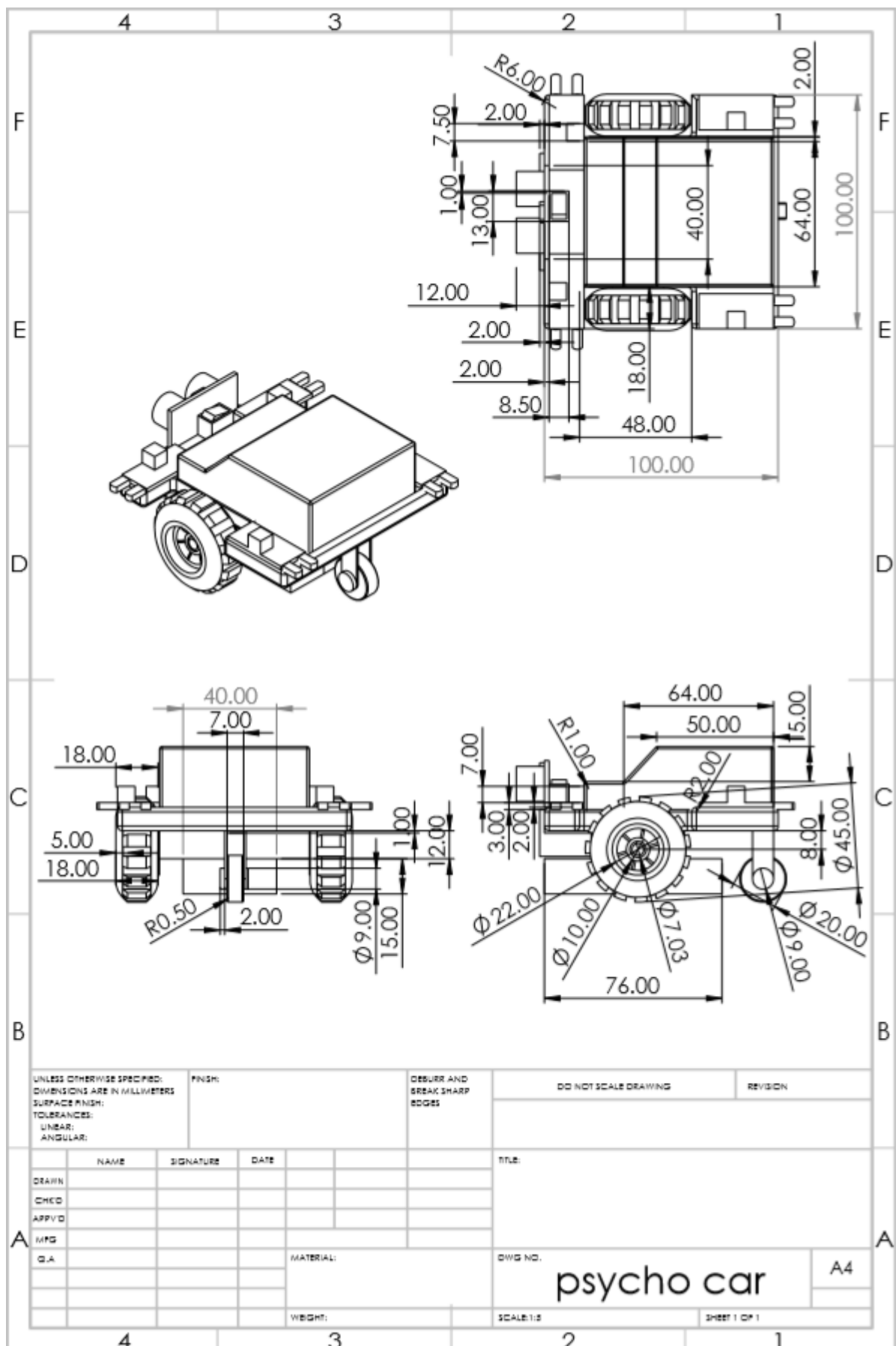
[4] “H-bridge with L298N mini dual motor driver”

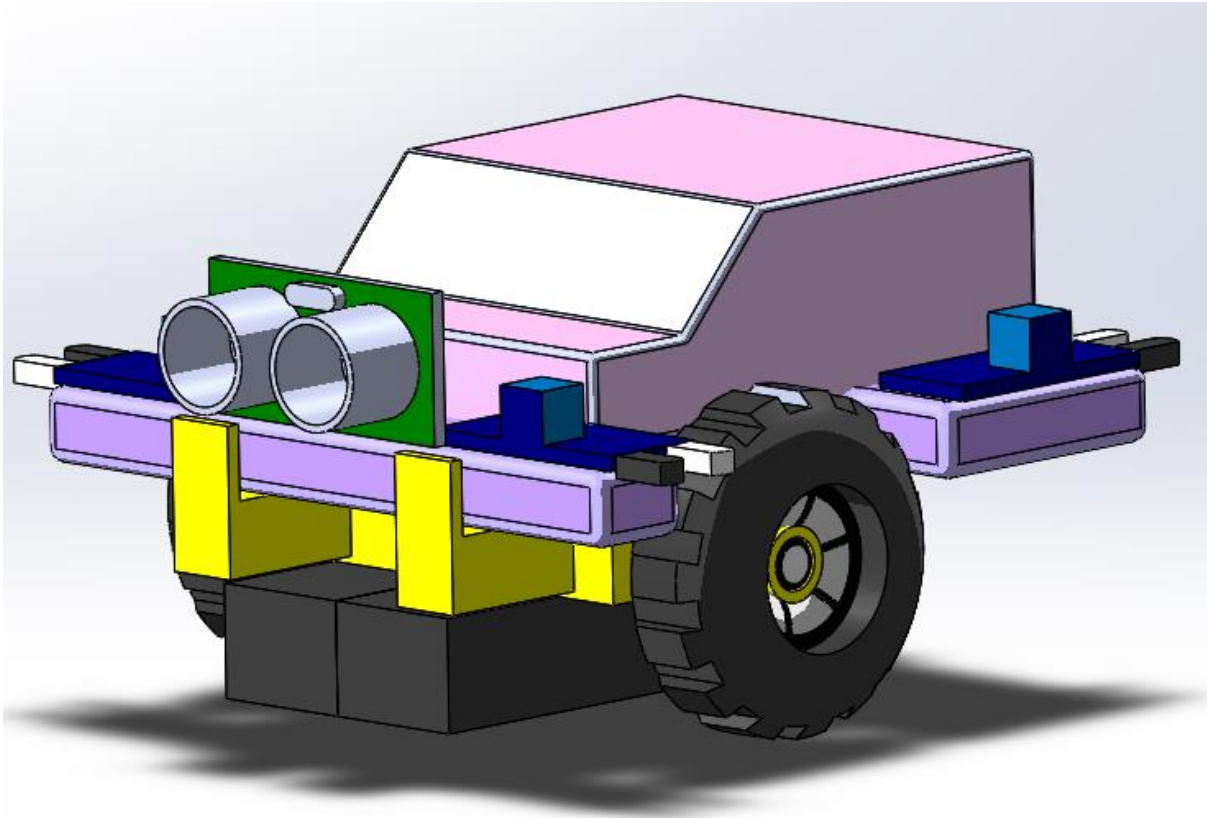
<http://www.robotsiam.com/article/7/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89-arduino-uno-r3-%E0%B8%81%E0%B8%B1%E0%B8%9A-l298n-%E0%B8%84%E0%B8%A7%E0%B8%9A%E0%B8%84%E0%B8%B8%E0%B8%A1%E0%B8%A1%E0%B8%AD%E0%B9%80%E0%B8%95%E0%B8%AD%E0%B8%A3%E0%B9%8C> (30 มีนาคม 2563)

[5] “Switch on-off”

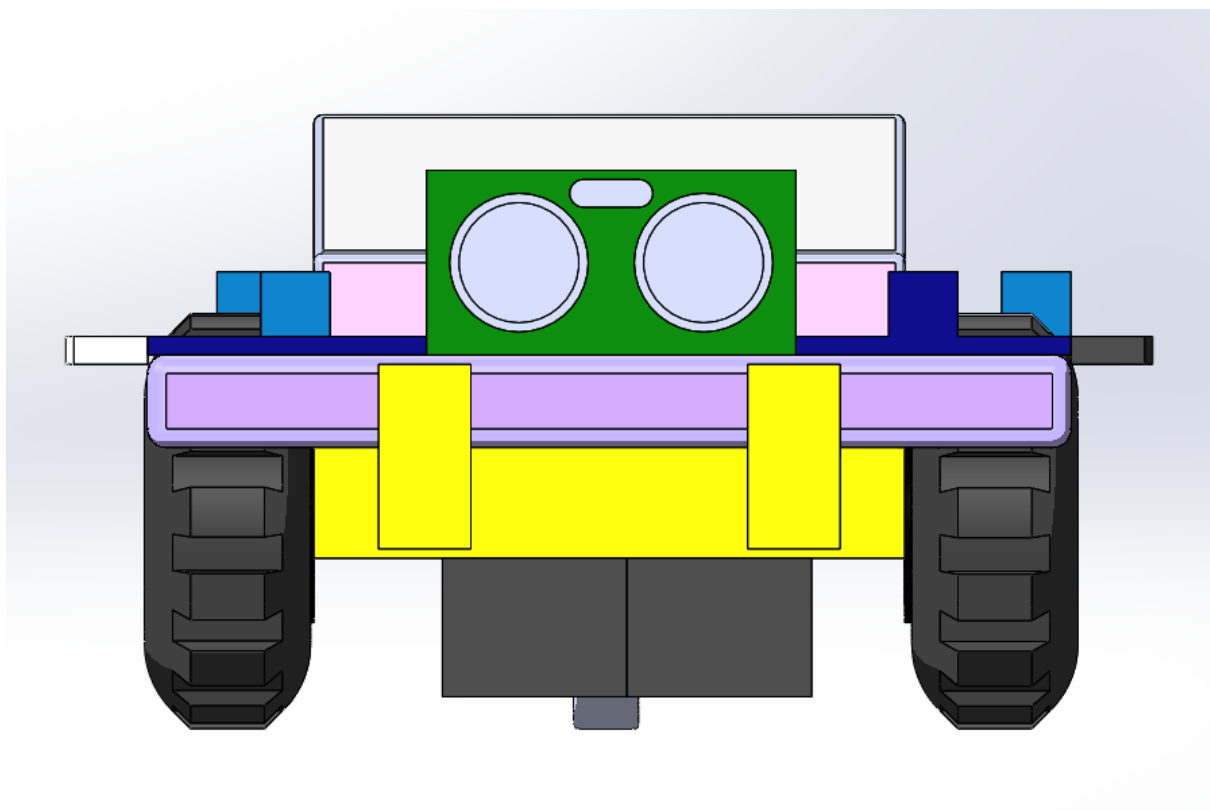
<http://joomlaleo.blogspot.com/2010/10/switch.html> (30 มีนาคม 2563)

ภาคผนวก

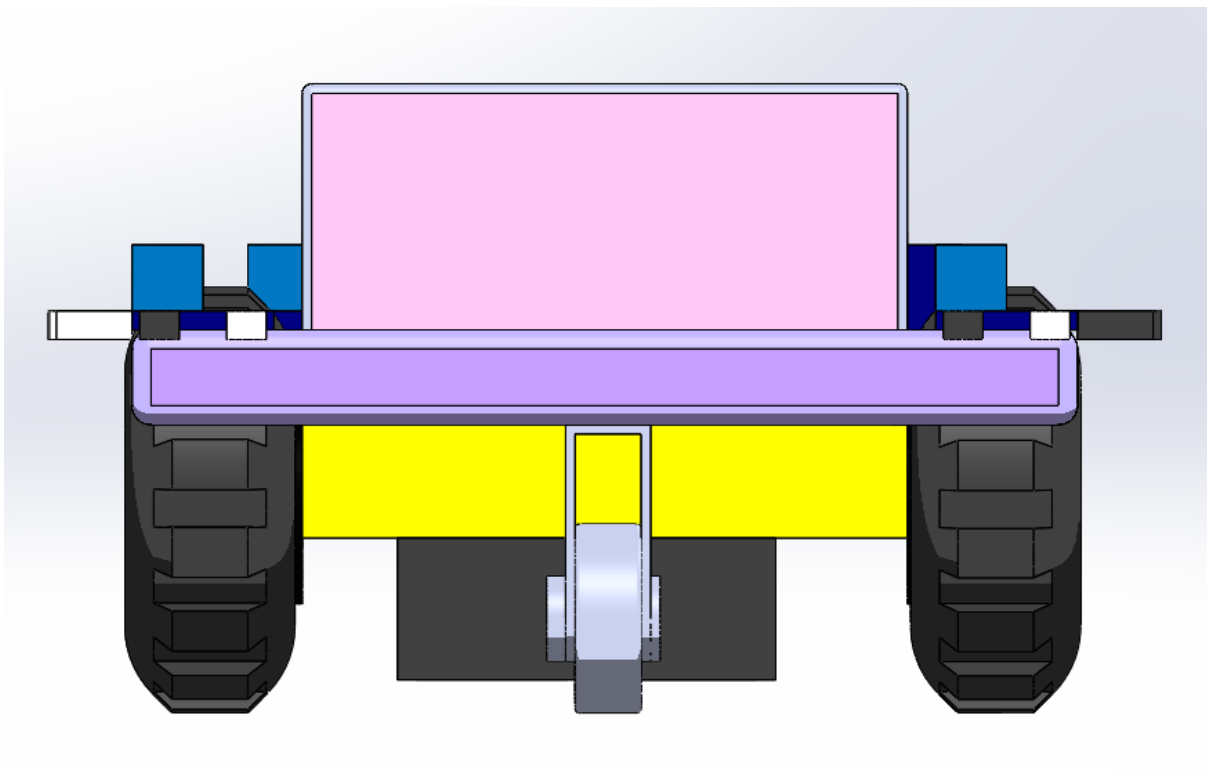




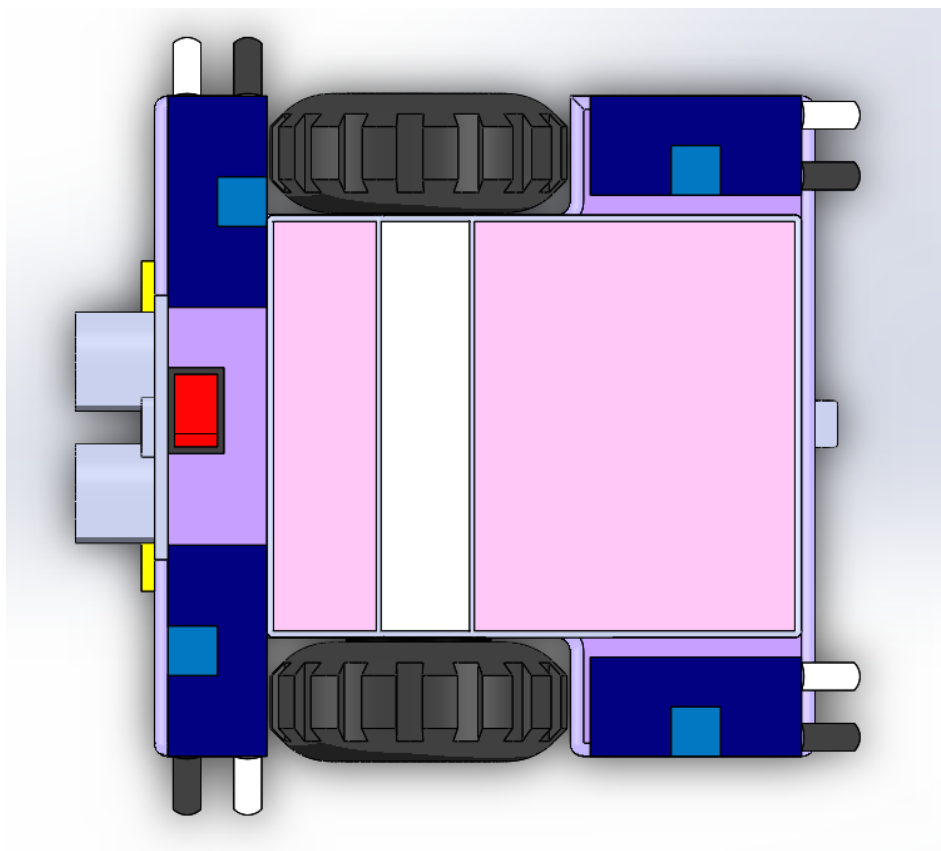
Isometric



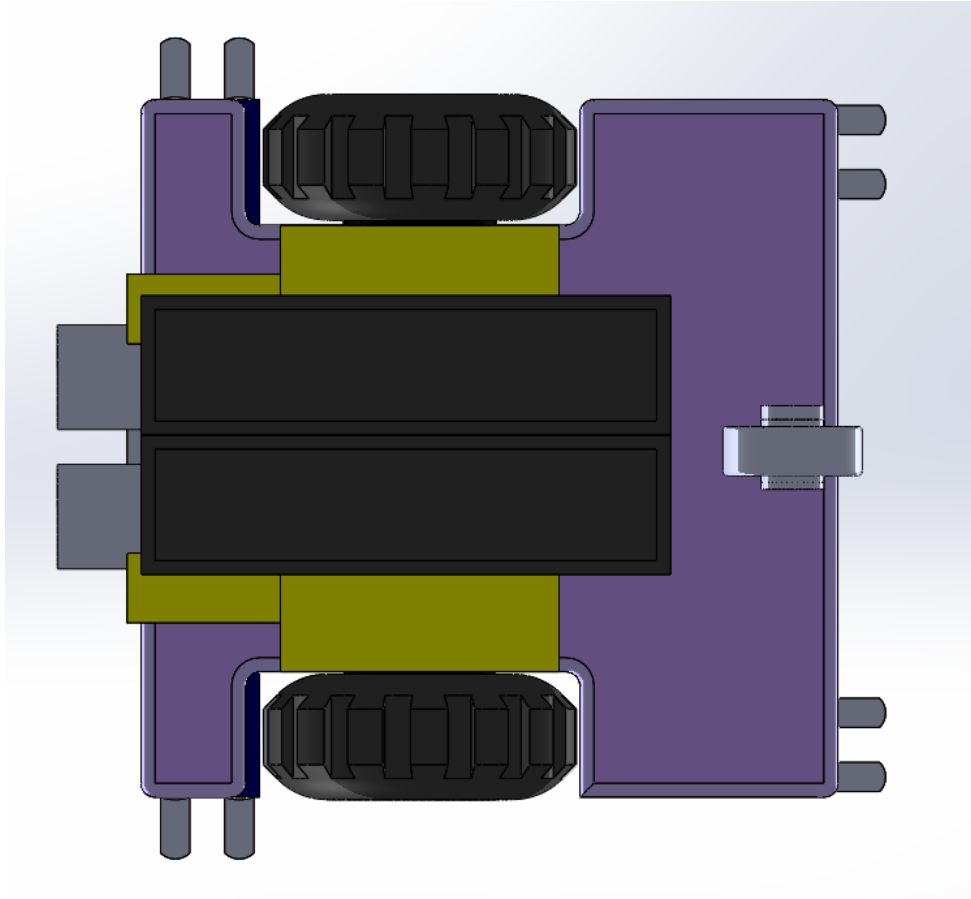
Front view



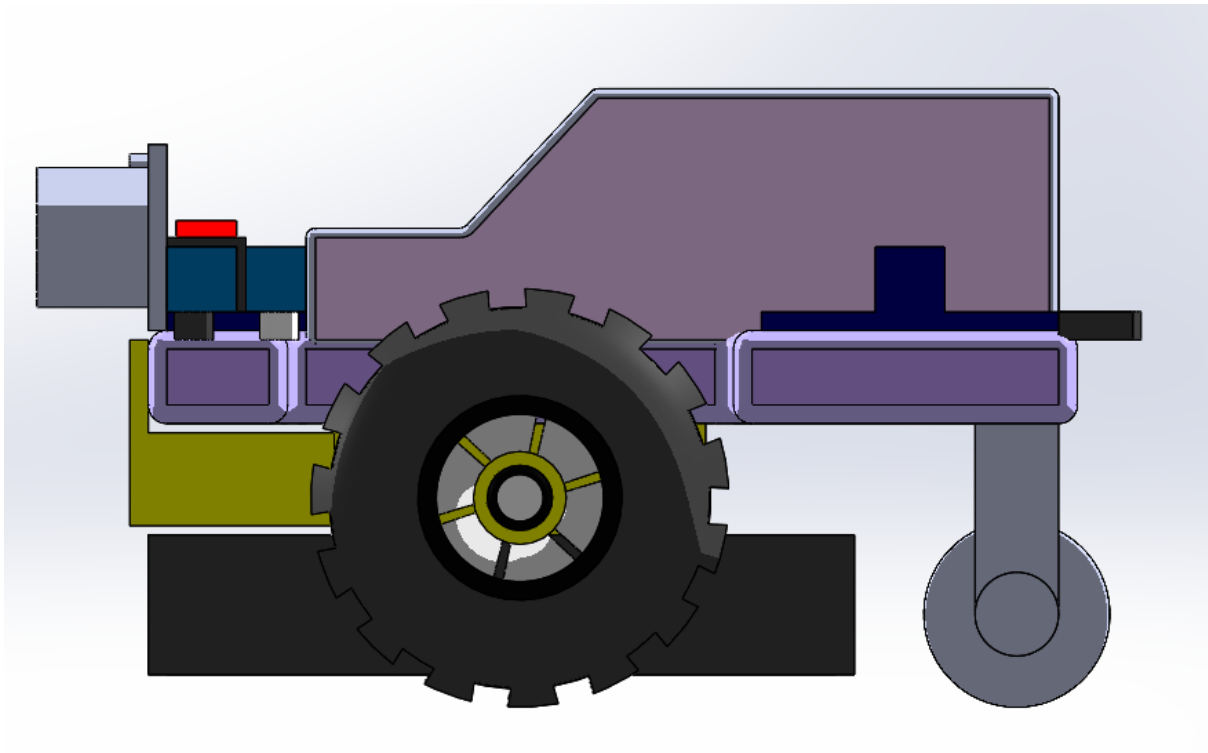
Back view



Top view



Bottom view



Side view