



รายงานวิชา Pre-Project รหัสวิชา 01216747

จัดทำโดย

นาย กษิต อีนกอง      รหัสนักศึกษา 60010042

นาย กิรติ ชาวสามทอง      รหัสนักศึกษา 60010087

นาย สหรัฐ รัตนโมคา      รหัสนักศึกษา 60011044

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

ภาคเรียนที่ 2 ปีการศึกษา 2562

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## บทคัดย่อ

ในปัจจุบันมีการนำเทคโนโลยีมาใช้ในโรงงานอุตสาหกรรม หรือธุรกิจต่างๆ การประกอบการเหล่านี้ได้นำเทคโนโลยีทางปัญญาประดิษฐ์หรือAI เข้ามามีส่วนร่วมในการลดเวลาในการผลิต ความราบรื่นในการดำเนินงาน เพื่อสร้างผลผลิตที่ตอบสนองต่อความพึงพอใจของลูกค้า

การจัดทำโครงงานครั้งนี้ได้ทำให้นักศึกษาได้เรียนรู้ถึงการใช้ Arduino board และการใช้อุปกรณ์ไฟฟ้า เช่น ตัวต้านทาน ตัวเก็บประจุ และอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เช่น Diode LED สามารถต่อยอดเพื่อใช้ในการสร้างปัญญาประดิษฐ์ได้ในอนาคต และเป็นประโยชน์ต่ออุตสาหกรรมการผลิต หรือ การประกอบการต่างๆ

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาของ ผศ.ดร.อุดม จันทร์จรัสสุข อาจารย์ที่ปรึกษาโครงการ และอาจารย์ผู้สอน ที่ได้ให้คำแนะนำ แนวคิด ตลอดจนวิธีการแก้ไขข้อบกพร่องต่างๆ จนโครงการนี้เป็นอันเสร็จสมบูรณ์ ผู้ศึกษาจึงขอกราบพระคุณเป็นอย่างสูง

ขอบคุณเพื่อนสมาชิกกลุ่มที่ช่วยกันแก้ไขปัญหา ให้ข้อเสนอแนะดีๆ รับผิดชอบต่อหน้าที่ที่ได้รับมอบหมาย ตลอดจนโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

# บทที่ 1

## บทนำ

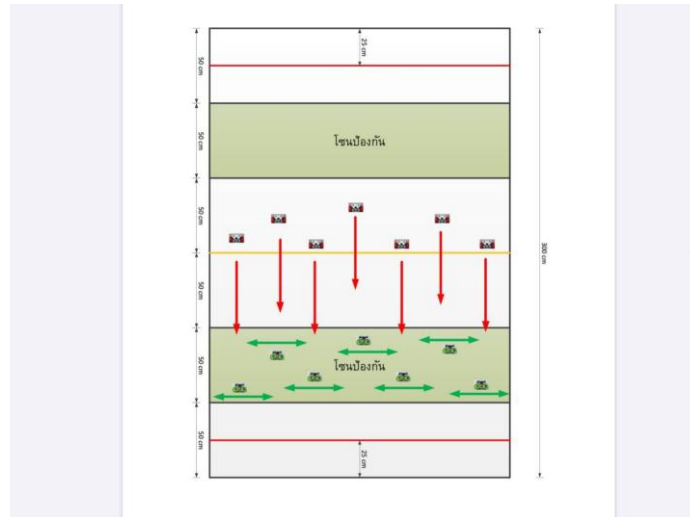
### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมีความก้าวหน้าทางเทคโนโลยี และมีการพัฒนาเทคโนโลยีที่หลากหลาย นวัตกรรมเหล่านี้ถูกพัฒนาขึ้นเพื่อตอบสนองความต้องการของมนุษย์ โดยเฉพาะการคิดค้นสิ่งต่างๆ เพื่อคอยช่วยเหลือผู้คน อย่าง นวัตกรรมหุ่นยนต์ ด้วยเทคโนโลยีที่มีความหลากหลาย หุ่นยนต์ได้กระจายไปอยู่ในแทบทุกที่ ไม่ว่าจะเป็นโรงพยาบาล มหาวิทยาลัย โรงเรียน หรือแม้แต่บ้านของเราเอง และแน่นอนว่าสถานที่ที่มีการใช้นวัตกรรมหุ่นยนต์มากที่สุดก็หนีไม่พ้นโรงงานอุตสาหกรรม

โรงงานอุตสาหกรรมในประเทศไทยมีการผลิตยานยนต์ หรือเครื่องมืออิเล็กทรอนิกส์ เป็นส่วนใหญ่ ดังนั้นการใช้ทรัพยากรบุคคลจึงมีจำนวนมาก ทว่าด้วยปัญหาด้านการขาดแคลนแรงงาน และงานบางประเภทมีการทำงานซ้ำซ้อนกัน หรืองานเสี่ยงและอันตราย ซึ่งได้นำนวัตกรรมหุ่นยนต์เข้ามามีส่วนร่วมในการทำงาน หุ่นยนต์นั้นสามารถทำงานได้อย่างต่อเนื่อง มีความแม่นยำ และรวดเร็วกว่ามนุษย์ อีกทั้งยังอดทนต่อสภาพแวดล้อม จึงถูกนำมาใช้แทนที่ และให้มนุษย์เป็นผู้ควบคุมแทน ซึ่งจุดประสงค์หลักในการนำนวัตกรรมหุ่นยนต์เข้ามามีส่วนร่วมในโรงงานอุตสาหกรรม เพื่อเพิ่มกำลังในการผลิต ยิ่งนวัตกรรมหุ่นยนต์ใหม่ๆมีมากเท่าไร ก็เป็นการเพิ่มความหลากหลายให้การทำงานและเปิดโอกาสให้ผู้ประกอบการ ผลิตชิ้นงานได้หลากหลายและรวดเร็วยิ่งขึ้น

### 1.2 ปัญหา

การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่น บอลลุนดำน หรือ เล่นเตย โดยแบ่งเป็นทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้าม จนผ่านเส้นแดง แล้วกลับมาอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้ ก็จะเป็นฝ่ายชนะในการแข่งขันรอบนั้น หุ่นยนต์ที่ถูกจับได้จะถูกตัดออกจากการแข่งขันในรอบนั้น ส่วนทีมรับ จะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น ดังรูปที่ 1-1



รูปที่ 1-1 สนามการแข่งขัน

### 1.3 วัตถุประสงค์

1. เพื่อศึกษาการโปรแกรมและการออกแบบหุ่นยนต์ ซึ่งเป็นพื้นฐานนำไปสู่การต่อยอดในอนาคต เพื่อใช้ในโรงงานอุตสาหกรรม หรือเป็นนวัตกรรมใหม่ๆ ที่อำนวยความสะดวกของมนุษย์
2. เพื่อศึกษาอุปกรณ์ เครื่องมืออิเล็กทรอนิกส์ ทำให้เข้าใจถึงหลักการการทำงานของอุปกรณ์
3. ฝึกการทำงานเป็นทีมอย่างเป็นระบบ
4. ฝึกการคิดสร้างสรรค์และการแก้ไขปัญหาเฉพาะหน้า

### 1.4 ขอบเขตโครงการ

ออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลยุทธ์ออกเป็น 2 กลยุทธ์คือฝ่ายรุกและฝ่ายรับ ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ ซึ่งกำหนดให้ตัวรถมี 4 ล้อ ขนาด 10\*10 ซม. (เป็นรูป สามเหลี่ยม) สูงประมาณ 10 ซม. ใช้ออเตอร์ 4 ตัว

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงงานนี้ได้จัดทำขึ้นเพื่อออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลุยุทธ์ออกเป็น 2 กลุยุทธ์ คือกลุยุทธ์รับและกลุยุทธ์รุก มีทฤษฎีที่เกี่ยวข้องต่อไปนี้

#### 2.1 ภาษา C/C++

โครงสร้างภาษา C/C++ บนไมโครคอนโทรลเลอร์ Arduino โปรแกรมจะมีฟังก์ชันหลัก (Structure) อย่างน้อย 2 ฟังก์ชัน มีดังต่อไปนี้

1. ฟังก์ชัน Setup () เป็นฟังก์ชันการกำหนดค่าต่าง ๆ ในส่วนนี้มีการกำหนดค่าเพียงครั้งเดียวเท่านั้น เช่น กำหนดขาในการใช้งานให้เป็นขาอินพุตหรือขาเอาต์พุต การกำหนดค่าของการเรียกใช้ไลบรารี[1]

Void setup ()

```
{  
  
    //เป็นส่วนของคำสั่ง สำหรับกำหนดการทำงานในโปรแกรม และทำเพียงครั้งเดียว  
  
}
```

2. ฟังก์ชัน Loop () เป็นส่วนในการเขียนโปรแกรมและสั่งให้โปรแกรมทำงาน ซึ่งมีการทำงานเป็นแบบวนลูปไปเรื่อย ๆ ตามการเขียนโปรแกรมของผู้พัฒนาโปรแกรมเพื่อรับค่าจากอินพุต นำค่าที่ได้มาประมวลผล แล้วทำการส่งข้อมูลออกเอาต์พุตเพื่อควบคุมการทำงานตามโปรแกรม

Void loop ()

```
{  
  
    // เป็นโปรแกรมหลักของคำสั่ง ซึ่งในส่วนนี้โปรแกรมมีการทำงานตลอดเวลา  
  
}
```

ส่วนชุดคำสั่งในการควบคุม (Control Structure) เป็นคำสั่งให้ไมโครคอนโทรลเลอร์ทำงานตามเงื่อนไขหรือรูปแบบที่ผู้พัฒนาโปรแกรมต้องการ มีคำสั่งต่าง ๆ ดังต่อไปนี้

- คำสั่ง if เป็นคำสั่งในการตรวจสอบเงื่อนไขการทำงานของโปรแกรมถ้าเงื่อนไขเป็นจริง ให้ทำงานตามคำสั่งที่กำหนดนั้น มีรูปแบบคำสั่งดังนี้

if (เงื่อนไขที่ตรวจสอบ)

{

    คำสั่งที่ให้ทำงาน เมื่อเงื่อนไขเป็นจริง

}

- คำสั่ง if else เป็นคำสั่งกำหนดเงื่อนไขการทำงานของโปรแกรม โดยมี 2 เงื่อนไข ถ้าเงื่อนไขเป็นจริงทำงานตามคำสั่งที่กำหนดแบบหนึ่ง ถ้าเงื่อนไขเป็นเท็จทำงานตามคำสั่งที่กำหนดอีกแบบหนึ่ง มีรูปแบบคำสั่งดังนี้

if (เงื่อนไขที่ตรวจสอบ)

{

    คำสั่งที่ให้ทำงาน เมื่อเงื่อนไขเป็นจริง

}

else

{

    คำสั่งที่ให้ทำงาน เมื่อเงื่อนไขเป็นเท็จ

}

- คำสั่ง for เป็นคำสั่งให้โปรแกรมทำงานซ้ำตามจำนวนรอบที่ต้องการมีรูปแบบคำสั่งคือ

for (ค่าเริ่มต้น ; เงื่อนไขการทำซ้ำ ; การเพิ่มหรือลดค่าตัวแปรในแต่ละรอบ)

```
{
```

```
คำสั่งที่ทำงาน
```

```
}
```

- คำสั่ง Switch case เป็นคำสั่งเพื่อกำหนดการทำงานของโปรแกรมหลาย ๆ เงื่อนไข ถ้าตัวแปรที่กำหนดตรงกับเงื่อนไขนั้น ๆ ทำให้โปรแกรมทำงานตามที่กำหนดไว้แต่ละเงื่อนไข มีรูปแบบคำสั่ง ดังนี้ Switch (ตัวแปร ที่ต้องการตรวจสอบ)

```
{
```

```
case 1: คำสั่งที่ทำงาน เมื่อตรวจสอบว่า ตัวแปร == 1
```

```
break;
```

```
case 2: คำสั่งที่ทำงาน เมื่อตรวจสอบว่า ตัวแปร == 2
```

```
break;
```

```
default: คำสั่งที่ทำงาน เมื่อตรวจสอบว่า ตัวแปรไม่ตรงกับเงื่อนไขใด ๆ
```

- คำสั่ง while เป็นคำสั่งทำซ้ำแบบวนรอบ ถ้าเงื่อนไขเป็นจริงโปรแกรมทำงานตามคำสั่งที่เขียนไว้ในวงเล็บปีกกา แต่ถ้าเงื่อนไขเป็นเท็จโปรแกรมจบการทำงานในคำสั่ง while มีรูปแบบคำสั่ง ดังนี้

```
while (เงื่อนไขที่ตรวจสอบ)
```

```
{
```

```
คำสั่งที่ทำงาน เมื่อเงื่อนไขยังเป็นจริง
```

```
}
```

- คำสั่ง Do while เป็นคำสั่งทำซ้ำแบบวนรอบ โดยมีการทำงานตรงกันข้ามกับคำสั่ง

while คือทำงานตามคำสั่งที่เขียนไว้ในวงเล็บปีกกา แล้วจึงมาตรวจสอบเงื่อนไข แต่ถ้าเงื่อนไขเป็นเท็จโปรแกรมจบการทำงานในคำสั่ง do มีรูปแบบคำสั่งดังนี้



Do

{

คำสั่งที่ใหทำงาน

} while (เงื่อนไขที่ตรวจสอบ)

- คำสั่ง break เป็นคำสั่งใช้ร่วมกับคำสั่งการทำงานแบบวนรอบ ได้แก่ คำสั่ง do, for while หรือ Switch เพื่อให้โปรแกรมหยุดการทำงานจากการวนรอบโดยไม่มีเงื่อนไข

- คำสั่ง continue เป็นคำสั่งใช้สำหรับข้ามการทำงานของคำสั่งถัดไป คำสั่งนี้เขียนอยู่ใน คำสั่งการทำงานแบบวนรอบ ได้แก่ คำสั่ง do, for หรือ while

- คำสั่ง return เป็นคำสั่งจบการทำงานในโปรแกรมน้อย

โดยชุดคำสั่งที่ใช้เป็นชุดคำสั่งในการเขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามโปรแกรมที่ออกแบบไว้ โดยมีคำสั่งต่าง ๆ ดังนี้

#### 1. คำสั่งดิจิทัล อินพุต/เอาต์พุต

- คำสั่ง pinMode () เป็นการกำหนดพอร์ตเป็นอินพุตหรือเอาต์พุต

- คำสั่ง digitalWrite () เป็นการเขียนข้อมูลออกพอร์ตที่กำหนด

- คำสั่ง digitalRead () เป็นการอ่านข้อมูลเข้าพอร์ตที่กำหนด

#### 2. คำสั่งอนาล็อก อินพุต/เอาต์พุต

- คำสั่ง analogReference () เป็นการกำหนดค่าแรงดันอ้างอิงที่ใช้สำหรับอนาล็อกอินพุต

- คำสั่ง analogRead () เป็นการอ่านแรงดันไฟฟ้าแบบอนาล็อกและแปลงเป็นจำนวนเต็ม มีค่าระหว่าง 0 ถึง 1023

- คำสั่ง analogWrite () เป็นการให้ PWM เขียนค่าออกทางพอร์ตที่กำหนด

#### 3. คำสั่งเวลา

- คำสั่ง `millisecond ()` เป็นการหนดเวลามีหน่วยเป็นมิลลิวินาทีของ Arduino ตอนที่ไฟเลี้ยงเข้า Arduino

- คำสั่ง `microsecond ()` เป็นการหนดเวลามีหน่วยเป็นไมโครวินาทีของ Arduino ตอนที่ไฟเลี้ยงเข้า Arduino

- คำสั่ง `delay ()` เป็นการหนดเวลาตามค่าที่กำหนด มีหน่วยเป็นมิลลิวินาที

- คำสั่ง `delayMicroseconds ()` เป็นการหนดเวลาตามค่าที่กำหนด มีหน่วยเป็นไมวินาที

ซึ่งจะต้องประกอบด้วยตัวแปรชนิดต่าง กล่าวคือตัวแปรเป็นชื่อเรียกแทนพื้นที่เก็บข้อมูลในหน่วยความจำของไมโครคอนโทรลเลอร์ โดยชนิดของข้อมูลหรือรูปแบบของตัวแปรต่าง ๆ มีดังนี้

#### 1. ค่าคงที่

- คำสั่ง `HIGH/LOW` แทนสถานะลอจิก “1” กับลอจิก “0”
- คำสั่ง `INPUT/OUTPUT` ใช้สำหรับกำหนดค่าอินพุตกับเอาต์พุต

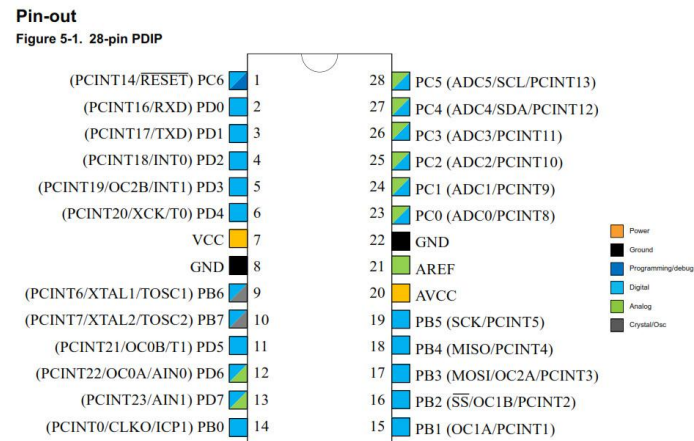
#### 2. ชนิดของข้อมูล

- `Void ()` ใช้เฉพาะในการประกาศฟังก์ชัน
- `char ()` มีค่าตั้งแต่ 127 ถึง 127 ใช้สำหรับเก็บข้อมูลที่เป็นตัวอักษร
- `int ()` มีค่าตั้งแต่ - 32,767 ถึง 32,767

## 2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ คอมพิวเตอร์ขนาดเล็กที่ถูกรวมไว้ในชิปเดียวประกอบด้วย หน่วยประมวลผลกลาง (CPU) วงจรอินพุต/เอาต์พุต หน่วยความจำแรมและแฟลช ตัวจับเวลา ตัวนับ เป็นต้น

สำหรับไมโครคอนโทรลเลอร์ Atmega 328p มีโครงสร้างภายในเป็นแบบ RISC (Reduced instruction set Computer) มีหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช สามารถเขียน-ลบโปรแกรมใหม่ได้หลายครั้ง โปรแกรมข้อมูลเป็นแบบ In-System programmable ดังรูปที่ 2-1



รูปที่ 2-1 ตำแหน่งและรูปร่างของไมโครคอนโทรลเลอร์ ATmega328p

โดยคุณสมบัติเบื้องต้นของไมโครคอนโทรลเลอร์ ATmega328 มีดังนี้

Microcontroller :	ATmega 328P (8 bit)
Operating Voltage :	5 Volts
Digital I/O Pin :	14 Pins
Analog Input Pin :	6 Pins
DC Current per I/O Pin :	20 mA
Flash Memory :	32 KB
SRAM :	2 KB
EEPROM :	1 KB
Clock Speed :	16 MHz
Size :	53.4 x 68.6 mm
Weight :	25 g

ตารางที่ 1 แสดงคุณสมบัติของ ไมโครคอนโทรลเลอร์ ATmega328

## 2.3 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง ( IR Infrared Obstacle Avoidance Sensor Module)

เซ็นเซอร์ใช้ตรวจจับวัตถุโดยใช้หลักการสะท้อนของแสงเมื่อไปชนวัตถุ (Reflective) สามารถปรับความไวในการตรวจจับได้ ใช้แสงอินฟราเรดในการตรวจจับ[2] แสดงดังรูปที่ 2-2



รูปที่ 2-2 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง

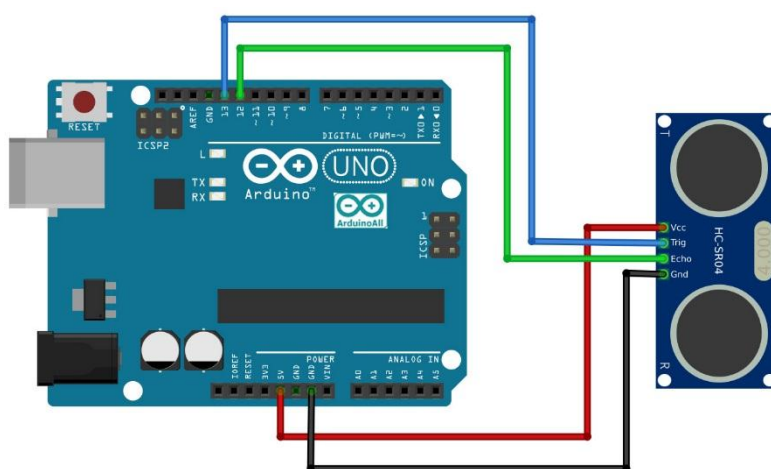
มีคุณสมบัติดังต่อไปนี้

- สามารถตรวจจับวัตถุได้ในระยะ 2 - 30 cm.
- ใช้แรงดันไฟฟ้าในการทำงาน 3V - 5.5V
- ใช้หลักการสะท้อนของแสงในการตรวจจับ โดยมีหลอด LED อินฟราเรดส่งแสง และมีโฟโตทรานซิสเตอร์ในการรับแสง
- สามารถแยกสีขาว - ดำ ได้
- ใช้ไอซีเปรียบเทียบแรงดันเบอร์ LM393

## 2.4 เซนเซอร์วัดระยะทาง (Ultrasonic Module)

เซนเซอร์วัดระยะทางด้วย Ultrasonic ใช้หลักการ ส่งคลื่นเสียงความถี่ต่ำ Ultrasonic ไปเมื่อคลื่นเสียงกระทบกับวัตถุจะมีการสะท้อนกลับมา เซนเซอร์จับเวลาที่ส่งคลื่นเสียงออกไปจนถึงคลื่นเสียงสะท้อนกลับมา เมื่อนำมาคำนวณกับเวลาที่เสียงเดินทางในอากาศ ก็จะได้ระยะทางออกมา

จุดต่อการใช้งานของ เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04



รูปที่ 2-3 จุดต่อใช้งานของเซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04

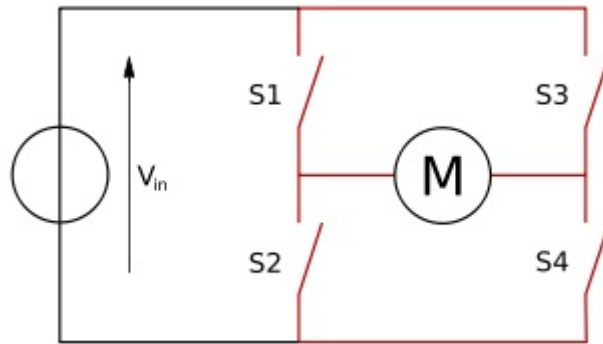
## 2.5 DC Motor Speed Control

ประกอบ H-bridge Driver และ Pulse-width modulation (PWM)

1. H-bridge Driver เป็นวงจรที่ทำหน้าที่ควบคุมทิศทางและความเร็วของมอเตอร์

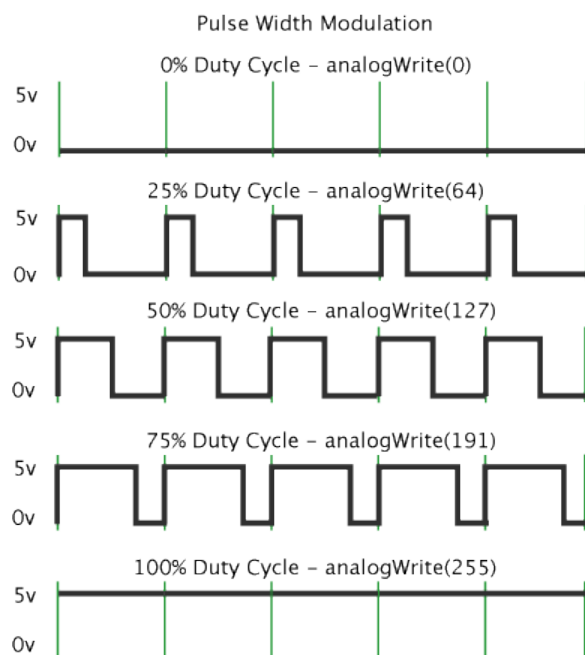
- หมุนตามเข็มนาฬิกา (Clockwise : CW) ก็ให้ S1 และ S4 ปิดวงจร และให้ S2 และ S3 เปิดวงจร
- หมุนทวนเข็มนาฬิกา (Counter Clockwise : CCW) ก็ให้ S2และ S3 ปิดวงจร และให้ S1 และ S4 เปิดวงจร

จะเห็นว่าสวิตช์จะทำงานเป็นคู่ S1 คู่กับ S4 และ S2 คู่กับ S3 คู่แรกทำงาน คู่สองต้องเปิดวงจร และในทางตรงข้ามก็คือคู่สองทำงาน คู่แรกต้องเปิดวงจร [3][4] แสดงผังรูป 2-4



รูปที่ 2-4 วงจร H-bridge Driver

2. Pulse-width modulation (PWM) เป็นการควบคุมพลังงานที่ส่งออกไปยังอุปกรณ์ที่ต้องการควบคุม เช่น ความแรงมอเตอร์ ความสว่างของไฟ LED โดยปกติหมายถึงการลดแรงดันที่ส่งออกไปยังมอเตอร์ แต่การลดแรงดันนั้นเป็นแนวทางที่ต้องใช้วงจรที่ซับซ้อนมีความยุ่งยากค่อนข้างมาก ซึ่งไม่ได้ลดแรงดัน หากแต่ใช้หลักการเปิด/ปิดมอเตอร์ด้วยความเร็วสูง จนผลค่าเฉลี่ยของแรงดันที่ได้ออกมาเทียบเท่ากับการเปลี่ยนแรงดันโดยตรง เทคนิคนี้ทำให้ไม่ต้องใช้วงจรซับซ้อน แต่การเขียนโปรแกรมจะยุ่งยากขึ้นบ้าง



รูปที่ 2-5 สัญญาณ PWM

L298N Dual H-bridge Motor Controller เป็นมอเตอร์ที่เลือกใช้ในโครงงานนี้มี สเปกดังนี้

Dual H bridge Drive Chip	L298N
แรงดันสัญญาณลอจิก	5V Drive voltage: 5V-35V
กระแสของสัญญาณลอจิก	0-36mA
กระแสขับเคลื่อนมอเตอร์	สูงสุดที่ 2A (เมื่อใช้มอเตอร์เดียว)
กำลังไฟฟ้าสูงสุด	25W
ขนาด	43 x 43 x 26 มิลลิเมตร
น้ำหนัก	26 กรัม

ตารางที่ 2 คุณสมบัติของ Dual H bridge Drive Chip

## 2.6 TCRT5000 Infrared Reflective sensor

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็ก สะดวกในการนำไปใช้ติดตั้งกับงานจำพวก หุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณ เป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัว หลอดโฟโตไดโอดที่ทำหน้าที่รับแสงอินฟราเรด [5]



รูปที่ 2-6 TCRT5000 Infrared Reflective sensor

## 2.7 DC/DC step-up MT3608

โมดูลแปลงแรงดันต่ำเป็นแรงดันสูงรองรับแรงดันอินพุตได้ตั้งแต่ 2V ถึง 24V แรงดันเอาต์พุตสามารถปรับได้ตั้งแต่ 5V ถึง 28V จ่ายกระแสได้สูงสุดถึง 2A



รูปที่ 2-7 DC/DC step-up MT3608



## บทที่ 3

### การออกแบบและการจัดทำโครงการ

#### 3.1 การออกแบบกลยุทธ์

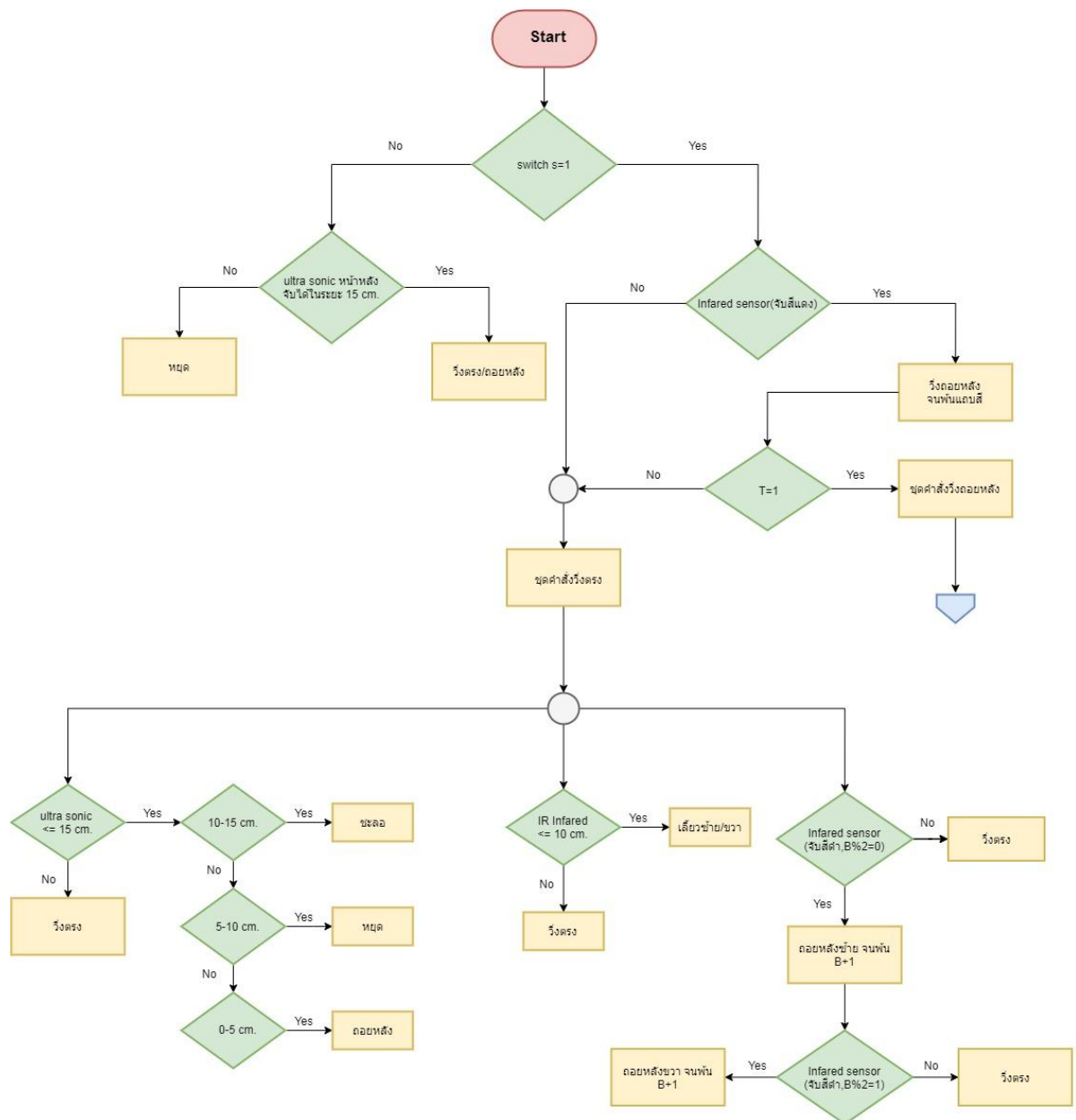
ในการออกแบบบรรจุภัณฑ์ทั้งหมด 4 ล้อ โดยมี 2 ล้อหน้า 2 ล้อหลัง สามารถขับเคลื่อนได้ 4 ล้อ เพื่อความคล่องตัวและสมมาตรในการเคลื่อนที่ และมีเซนเซอร์รอบตัว 4 จุดเพื่อใช้ในการตรวจจับสิ่งกีดขวาง เซนเซอร์ได้ทั้งรถ 1 จุด เพื่อใช้ในการตรวจจับเทปสี โดยมีกลยุทธ์การเคลื่อนที่ ดังนี้

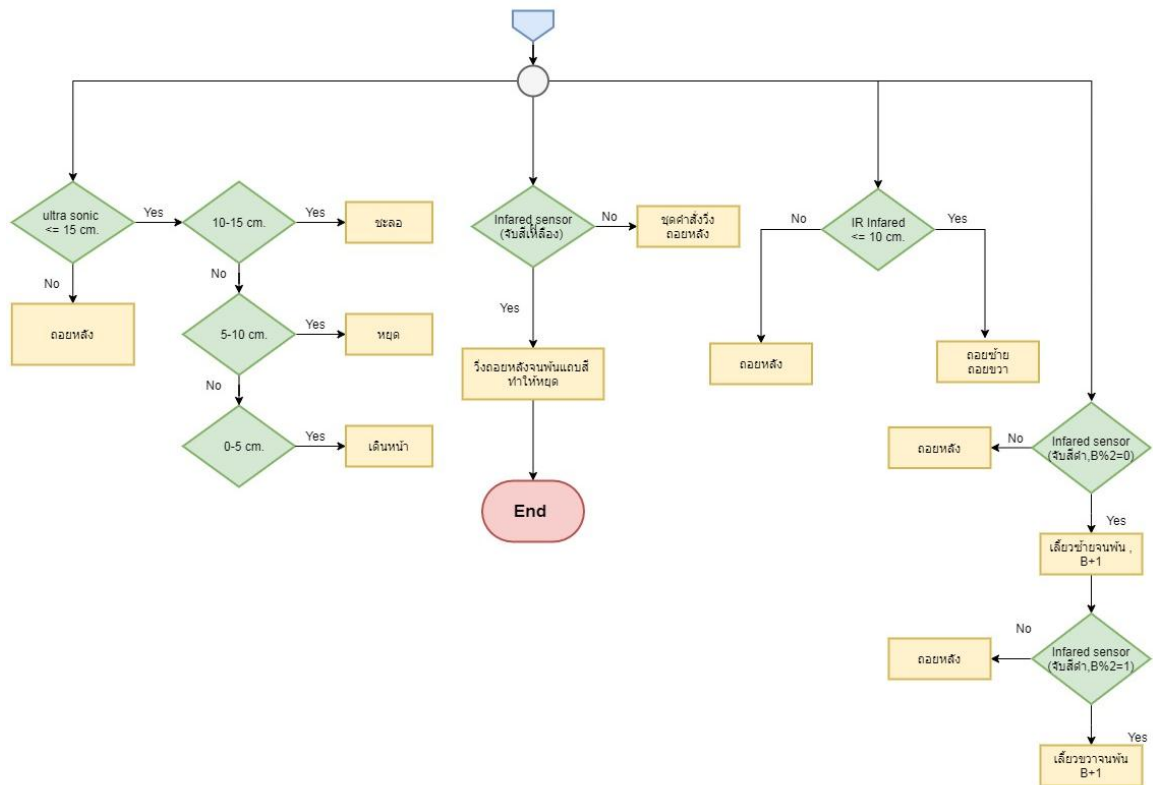
##### 3.1.1 กลยุทธ์รถ

เมื่อเป็นฝั่งรถ จะหันหน้าและเคลื่อนที่ในแนวตรง และมีการเปลี่ยนชุดคำสั่งเมื่อสัมผัสเส้นสีแดง และจะหยุดเมื่อสัมผัสเส้นสีเหลือง และเมื่อสัมผัสกับเส้นสีดำ(ขอบสนาม) จะทำการตั้งหลักใหม่ จะมีเซนเซอร์ตรวจจับโดยแบ่งเป็นชุดคำสั่งให้รถดังนี้

- เซนเซอร์ หน้ารถ ให้ตรวจจับสิ่งกีดขวางด้านหน้า และให้ทำการวิ่งตรงหรือถอยหลังหรือชะลอหรือหยุด
- เซนเซอร์ ซ้าย,ขวา ให้ตรวจจับสิ่งกีดขวางซ้าย ขวา และให้ทำการเลี้ยวซ้ายขวา
- เซนเซอร์ หลัง ให้ตรวจจับสิ่งกีดขวางด้านหลัง และให้ทำการวิ่งตรงหรือถอยหลังถอยหลังหรือชะลอหรือหยุด
- เมื่อไม่มีสัญญาณเซนเซอร์ให้รถวิ่งไปข้างหน้าหรือถอยหลังเมื่อเป็นขากลับ

แสดงผังรูปที่ 3-1 Flow Chart การออกแบบกลยุทธ์รถ





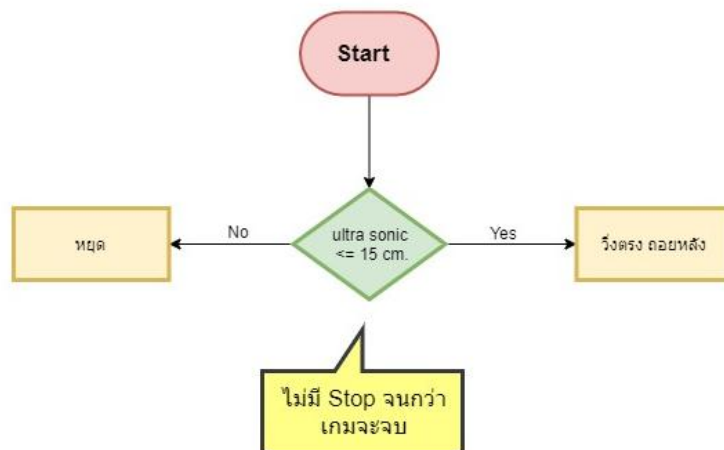
รูปที่ 3-1 Flow Chart การออกแบบกลยุทธรถ

### 3.1.2 กลยุทธ์รับ

เมื่อเป็นฝั่งรับ จะหันรถเป็นแนวขวาง จะมีเซนเซอร์ตรวจจับโดยแบ่งเป็นชุดคำสั่งให้รถดังนี้

- เซนเซอร์ หนักรถ ให้รถวิ่งตรง
- เซนเซอร์ หลังรถ ให้รถถอยหลัง
- เมื่อ ไม่มีสัญญาณเซนเซอร์ให้รถอยู่เฉยๆ

แสดงดังรูปที่ 3-2 Flow Chart การออกแบบกลยุทธรับ

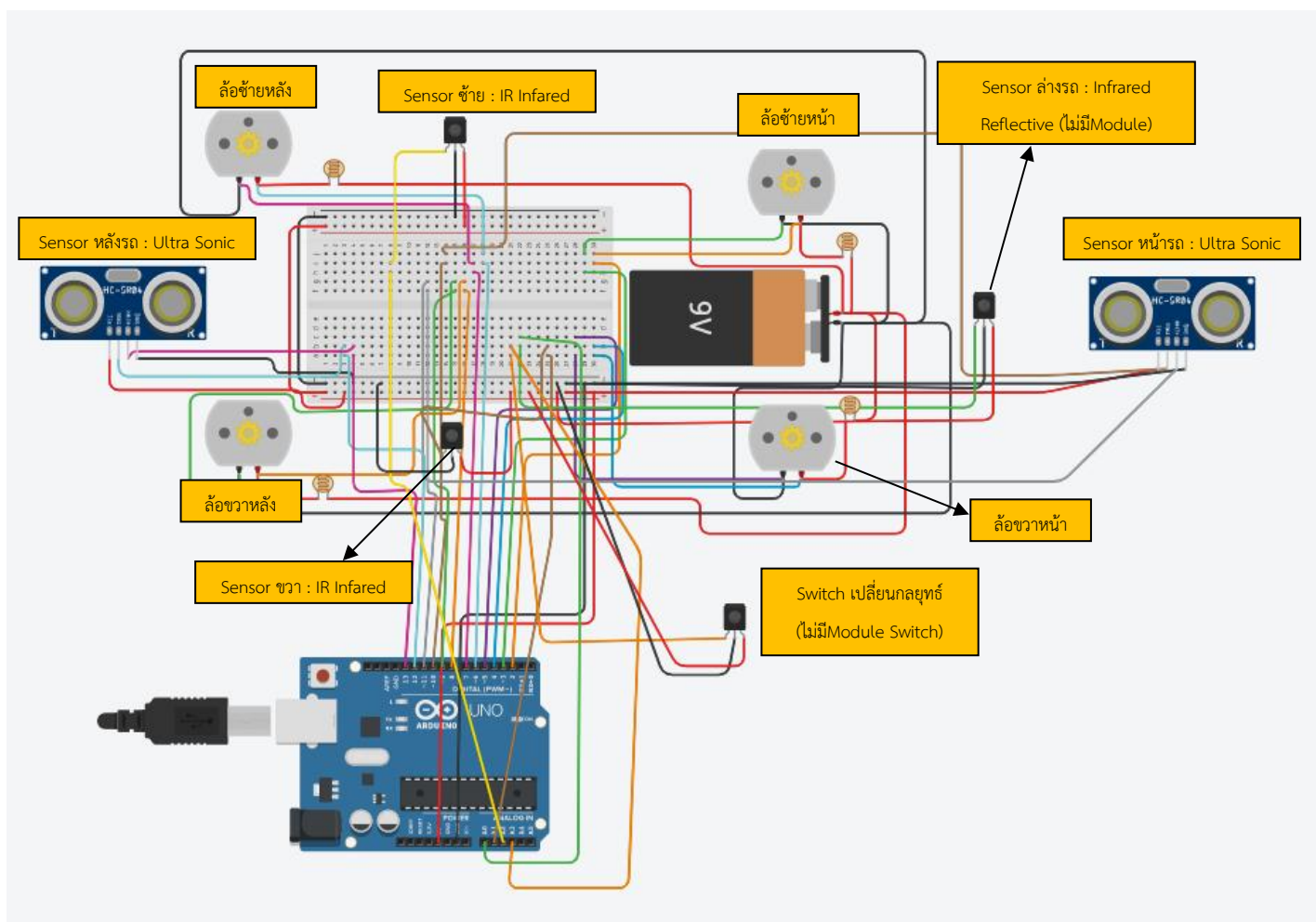


รูปที่ 3-2 Flow Chart การออกแบบกลยุทธรับ

### 3.2 การออกแบบการทำงานของวงจร

อุปกรณ์ที่ใช้ในการทำโครงงาน

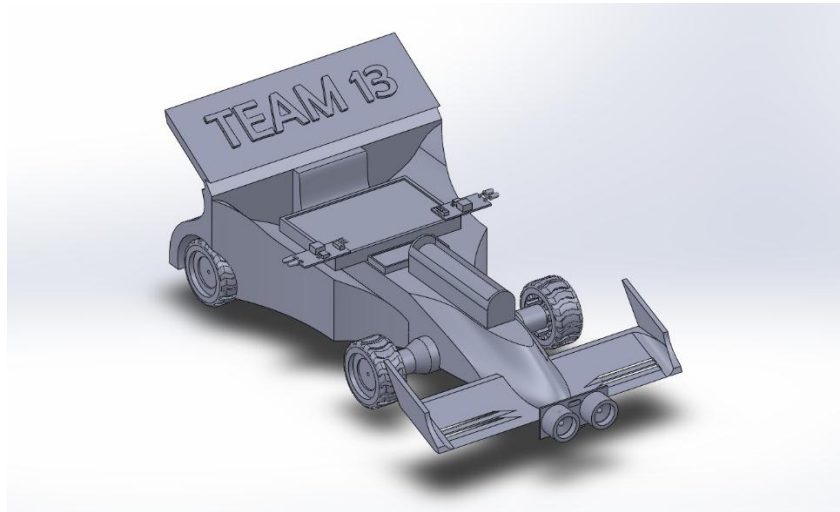
1. Arduino board 1 ตัว
2. ล้อ 4 ล้อ
- 3.มอเตอร์ 4 ตัว
- 4.ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7V 1 ก้อน
- 5.เซนเซอร์ 5 ตัว
- 6.โครงรถ
- 7.ที่ชาร์จแบตเตอรี่
- 8.สายไฟ



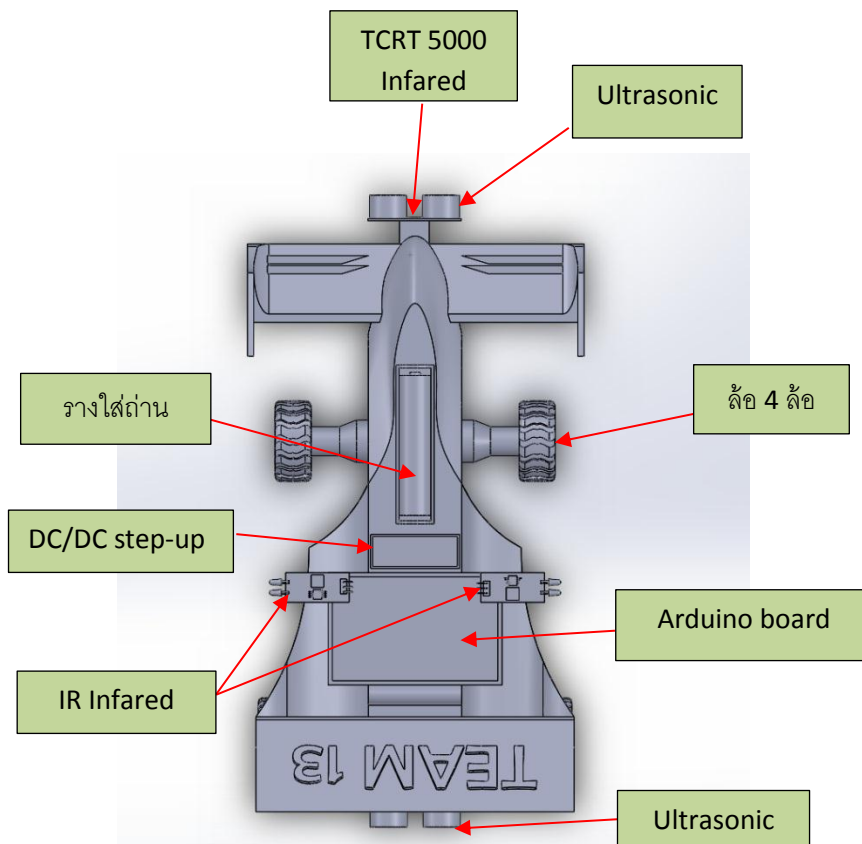
รูปที่ 3-3 ลักษณะการต่อวงจร

### 3.3 การออกแบบโครงสร้างรถ

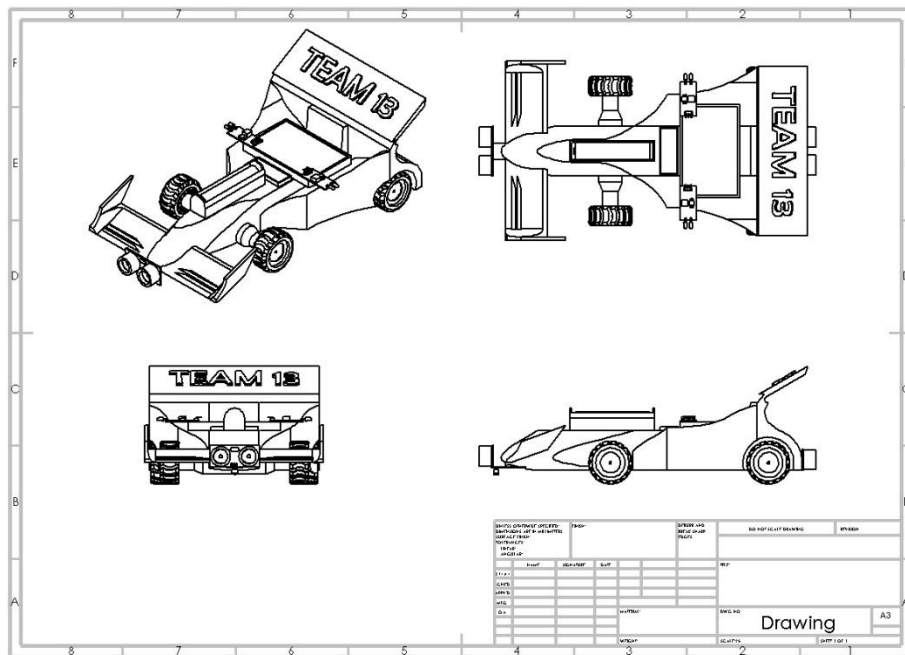
การออกแบบโครงสร้างรถ “Team 13” มีแรงบันดาลใจมาจาก Racing car หรือรถแข่งทางมีย่า ที่เคยเป็นของเล่นที่ยอดนิยมในช่วงยุค 90 ซึ่งมีการใช้มอเตอร์ต่อกับเฟืองโดยตรงในการขับเคลื่อน มีหลักการคล้ายกับโครงงานนี้ จึงออกแบบลักษณะรางถ่วงรวมถึงตำแหน่งการวางมอเตอร์คล้ายกับรถแข่งทางมีย่า โดยตัวรถที่ออกแบบจะมีโหนดต่ำใกล้พื้นเพื่อป้องกันการพลิกคว่ำ และช่วยให้ TCRT5000 อ่านค่าจากเส้นสีได้แม่นยำขึ้น แสดงดังรูปที่ 3-4



รูปที่ 3-4 โครงสร้างรถโดยรวม



รูปที่ 3-5 ตำแหน่งที่จัดวางอุปกรณ์บนตัวรถ



รูปที่ 3-6 Top view Front view Side view

### 3.4 การออกแบบและอธิบายโค้ด

ผู้จัดทำอธิบายโค้ดเป็น 4 ส่วน ได้แก่ 1. ประกาศตัวแปร, 2. Setup, 3. Function ต่างๆ และ 4. Loop ซึ่งอธิบายได้ดังนี้

#### 1. ประกาศตัวแปร

`#include <HCSR04.h>` เป็นการประกาศ library ให้ sensor ultra sonic ประมวลค่าที่รับและส่งเป็นหน่วย cm

`HCSR04 hc(10,11);` ultra sonic sensor ตัวด้านหน้ารถ

`HCSR04 hc2(12,13);` ultra sonic sensor ตัวด้านหลังรถ

`#define SwitchPin A3` Switch ในการสับเปลี่ยนคำสั่งรถหรือรับ

`#define ia1 2` `#define ia2 3` ล้อซ้ายหน้า

`#define ib1 4` `#define ib2 5` ล้อขวาหน้า

#define ic1 6      #define ic2 7      ล้อซ้ายหลัง

#define id1 8      #define id2 9      ล้อขวาหลัง

#define S2 A2      Sennsor Infared ด้านซ้าย

#define S3 A1      Sennsor Infared ด้านขวา

#define SBT A0      Sensor จับเทปสี่ล้อที่จอดรถ

#define maxSpd 255      ความเร็วของมอเตอร์

int SwitchState = 0;      ตัวแปรในการเปลี่ยนคำสั่งในการเลือกกลยุทธ์ของรถ

int T=0;      ตัวแปรในการเปลี่ยนคำสั่งในการเคลื่อนที่ ของกลยุทธ์ฝ่ายรุก กับเทปสีแดงและเหลือง

int B=0;      ตัวแปรในการเปลี่ยนคำสั่งในการเคลื่อนที่ ของกลยุทธ์ฝ่ายรุก กับเทปสีดำ

int sensorValue = analogRead(A0);      ตัวแปร sensorValue รับค่ามาจาก Sensor จับเทปสี่ล้อที่จอดรถ

int speed = maxSpd;      ตัวแปรspeedเท่ากับความเร็วของมอเตอร์

## 2. setup

pinMode(ia1, OUTPUT);      pinMode(ia2, OUTPUT);      ล้อซ้ายหน้าแสดงเป็น output

pinMode(ib1, OUTPUT);      pinMode(ib2, OUTPUT);      ล้อขวาหน้าแสดงเป็น output

pinMode(ic1, OUTPUT);      pinMode(ic2, OUTPUT);      ล้อซ้ายหลังแสดงเป็น output

pinMode(id1, OUTPUT);      pinMode(id2, OUTPUT);      ล้อขวาหลังแสดงเป็น output

pinMode(S2, INPUT);      เซนเซอร์ด้านซ้ายเป็นตัวรับค่า

pinMode(S3, INPUT);      เซนเซอร์ด้านขวาเป็นตัวรับค่า

pinMode(SBT, INPUT);      เซนเซอร์ใต้ที่จอดรถเป็นตัวรับค่า

Serial.begin(9600);      ความเร็วในการแสดงค่าของ Monitor



pinMode(SwitchPin, INPUT);    Switch เป็นตัวรับค่า

### 3. Fucntion คำสั่งต่างๆ

โดย แบ่งเป็น 5 ส่วน ดังนี้

ส่วนที่1 Function 3.1-3.9 จะเป็น Function เกี่ยวกับการขับเคลื่อนมอเตอร์

3.1 void Break()                    ให้อัดทั้งสี่เบรก

{

digitalWrite(ia1, HIGH);

digitalWrite(ia2, HIGH);

digitalWrite(ib1, HIGH);

digitalWrite(ib2, HIGH);

digitalWrite(ic1, HIGH);

digitalWrite(ic2, HIGH);

digitalWrite(id1, HIGH);

digitalWrite(id2, HIGH);

}

3.2 void Forward(int speed)    ให้อัดทั้งสี่วิ่งไปข้างหน้าด้วยความเร็ว speed

{

digitalWrite(ia1, LOW);

analogWrite(ia2, speed);

digitalWrite(ib1, LOW);

```

analogWrite(ib2, speed);

digitalWrite(ic1, LOW);

analogWrite(ic2, speed);

digitalWrite(id1, LOW);

analogWrite(id2, speed);

}

```

3.3 void Slow(int speed) ให้ล้อทั้งสี่วิ่งไปข้างหน้าด้วยความเร็ว speed\*0.3(ช้า)

```

{

digitalWrite(ia1, LOW);

analogWrite(ia2, speed*0.3);

digitalWrite(ib1, LOW);

analogWrite(ib2, speed*0.3);

digitalWrite(ic1, LOW);

analogWrite(ic2, speed*0.3);

digitalWrite(id1, LOW);

analogWrite(id2, speed*0.3);

}

```

3.4 void Reward(int speed) ให้ล้อทั้งสี่ถอยหลัง ด้วยความเร็ว speed

```

{

digitalWrite(ia2, LOW);

```

```

analogWrite(ia1, speed);

digitalWrite(ib2, LOW);

analogWrite(ib1, speed);

digitalWrite(ic2, LOW);

analogWrite(ic1, speed);

digitalWrite(id2, LOW);

analogWrite(id1, speed);

}

```

3.5 void RSlow(int speed)      ให้ล้อทั้งสี่ถอยหลัง ด้วยความเร็ว speed\*0.3(ช้า)

```

{

digitalWrite(ia2, LOW);

analogWrite(ia1, speed*0.3);

digitalWrite(ib2, LOW);

analogWrite(ib1, speed*0.3);

digitalWrite(ic2, LOW);

analogWrite(ic1, speed*0.3);

digitalWrite(id2, LOW);

analogWrite(id1, speed*0.3);

}

```

3.6 void Turnleft(int speed) ให้สองล้อหน้าทำการเลี้ยวซ้าย และสองล้อหลังทำการวิ่งไปข้างหน้าด้วย speed\*0.5

```
{  
  
    digitalWrite(ia1, LOW);  
  
    analogWrite(ia2, speed*0.7);  
  
    digitalWrite(ib2, LOW);  
  
    analogWrite(ib1, speed*0.3);  
  
    digitalWrite(ic1, LOW);  
  
    analogWrite(ic2, speed*0.5);  
  
    digitalWrite(id1, LOW);  
  
    analogWrite(id2, speed*0.5);  
  
}
```

3.7 void TurnRight(int speed) ให้สองล้อหน้าทำการเลี้ยวขวา และสองล้อหลังทำการวิ่งไปข้างหน้าด้วย speed\*0.5

```
{  
  
    digitalWrite(ia2, LOW);  
  
    analogWrite(ia1, speed*0.3);  
  
    digitalWrite(ib1, LOW);  
  
    analogWrite(ib2, speed*0.7);  
  
    digitalWrite(ic1, LOW);
```

```

analogWrite(ic2, speed*0.5);

digitalWrite(id1, LOW);

analogWrite(id2, speed*0.5);

}

```

3.8 void RTurnleft(int speed) ให้สองล้อหลังทำการถอยหลังซ้าย และสองล้อหลังทำการถอยหลังด้วย speed\*0.5

```

{

digitalWrite(ia1, LOW);

analogWrite(ia2, speed*0.5);

digitalWrite(ib2, LOW);

analogWrite(ib1, speed*0.5);

digitalWrite(ic1, LOW);

analogWrite(ic2, speed*0.7);

digitalWrite(id1, LOW);

analogWrite(id2, speed*0.3);

}

```

3.9 void RTurnRight(int speed) ให้สองล้อหลังทำการถอยหลังซ้าย และสองล้อหลังทำการถอยหลังด้วย speed\*0.5

```
{  
  
    digitalWrite(ia1, LOW);  
  
    analogWrite(ia2, speed*0.5);  
  
    digitalWrite(ib2, LOW);  
  
    analogWrite(ib1, speed*0.5);  
  
    digitalWrite(ic1, LOW);  
  
    analogWrite(ic2, speed*0.3);  
  
    digitalWrite(id1, LOW);  
  
    analogWrite(id2, speed*0.7);  
  
}
```

ส่วนที่ 2 Function 3.10-3.17 จะเป็น Function เกี่ยวกับชุดคำสั่งรับค่าของเซนเซอร์พร้อมการทำงานต่างๆ

3.10 void Sensor1() ใช้ ultra sonic ในการตรวจจับวัตถุอยู่ข้างหน้ารถ

```
{  
  
    int speed = maxSpd;  
  
    Serial.println(hc.dist() ); //showdistantfromfront  
  
    if((hc.dist()>=10) && (hc.dist()<=15)){  
  
        Slow(speed);}      ถ้าวัตถุ อยู่ในระยะ 10-15 cm ให้ชะลอรถ  
  
    else if ((hc.dist()>=5) && (hc.dist()<10)){
```

```

    Break();}      ถ้าวัตถุ อยู่ในระยะ 5-10 cm ให้หยุดรถ

else if ((hc.dist()>=0) && (hc.dist(<5))){

    Reward(speed);}   ถ้าวัตถุ อยู่ในระยะ 0-5 cm ให้รถถอยหลัง

else{

    Forward(speed);}  ถ้าวัตถุ ไม่อยู่ในระยะ 15 cm ให้รถวิ่งไปข้างหน้า

}

```

3.11 void Sensor2(){

```

    int speed = maxSpd;

    if(digitalRead(S2)==LOW){

        Serial.println("sensor detected object from left side");

        delay(20);

        Turnleft(speed);}  ถ้า sensor 2 จับได้ รถจะเลี้ยวซ้าย

    else{

        Forward(speed); ถ้า sensor 2 จับไม่ได้ รถจะวิ่งไปข้างหน้า

    }

}

```

3.12 void RSensor2(){

```

    int speed = maxSpd;

    if(digitalRead(S2)==LOW){

        Serial.println("sensor detected object from left side");

```

```

    delay(20);

    RTurnRight(speed);} ถ้า sensor 2 จับได้ รถจะถอยขวา

else{

    Reward(speed); ถ้า sensor 2 จับไม่ได้ รถจะถอยหลัง

}

}

```

3.13 void Sensor3(){

```

    int speed = maxSpd;

    if(digitalRead(S3)==LOW){

        Serial.println("sensor detected object from right side");

        delay(20);

        TurnRight(speed);} ถ้า sensor 3 จับได้ รถจะเลี้ยวขวา

    else{

        Forward(speed); ถ้า sensor 3 จับไม่ได้ รถจะวิ่งไปข้างหน้า

    }

}

```

3.14 void RSensor3(){

```

    int speed = maxSpd;

    if(digitalRead(S3)==LOW){

        Serial.println("sensor detected object from right side");
    }
}

```



```

    delay(20);

    RTurnleft(speed);} ถ้า sensor 3 จับได้ รถจะถอยซ้าย

    else{

Reward(speed); ถ้า sensor 3 จับไม่ได้ รถจะถอยหลัง

    }

}

```

3.15 void Sensor4() ใช้ ultra sonic ในการตรวจจับวัตถุอยู่ข้างหลังรถ

```

{

int speed = maxSpd;

Serial.println(hc2.dist() );

if((hc2.dist())>=10) && (hc2.dist()<=15)){

    RSlow(speed);} ถ้าวัตถุ อยู่ในระยะ 10-15 cm ให้ชะลอรถ

else if ((hc2.dist())>=5) && (hc2.dist()<10)){

    Break();} ถ้าวัตถุ อยู่ในระยะ 5-10 cm ให้หยุดรถ

else if ((hc2.dist())>=0) && (hc2.dist()<5)){

    Forward(speed);} ถ้าวัตถุ อยู่ในระยะ 0-5 cm ให้รถวิ่งไปข้างหน้า

else{

    Reward(speed);} ถ้าวัตถุไม่อยู่ในระยะ 15 cm ให้รถถอยหลัง

}

```

3.16 void DSensor1()

```

{

int speed = maxSpd;

Serial.println(hc2.dist() );

if((hc2.dist()>=0) && (hc2.dist()<=15)){

    Forward(speed);}    ถ้าวัตถุ อยู่ในระยะ 0-15 cm ให้รถวิ่งไปข้างหน้า

else{

    Break();}          ถ้าวัตถุไม่อยู่ในระยะ 0-15 cm ให้รถหยุด

}

```

3.17 void DSensor2() //ultrasonic

```

{

int speed = maxSpd;

Serial.println(hc2.dist() );

if((hc2.dist()>=0) && (hc2.dist()<=15)){

    Reward(speed);}    ถ้าวัตถุ อยู่ในระยะ 0-15 cm ให้รถวิ่งถอยหลัง

else{

    Break();}          ถ้าวัตถุไม่อยู่ในระยะ 0-15 cm ให้รถหยุด

}

```

ส่วนที่ 3 Function 3.18-3.21 จะเป็น Function เกี่ยวกับชุดคำสั่งรับค่าของเซนเซอร์ได้ทั้งรถพร้อมการทำงานต่างๆ

3.18 void DetectRed(){

if(sensorValue>=1700&&sensorValue<=1850)

{

Reward(speed);                    ถ้าsensor สัมผัสเทปสีแดง ให้ทำการถอยรถและ return T=1

delay(100);

T = 1;

}

}

3.19 void RDetectYellow(){

if(sensorValue>=1900&&sensorValue<=2100)

{

Reward(speed);                    ถ้าsensor สัมผัสเทปสีเหลือง ให้ทำการถอยรถและเพิ่มค่า T ขึ้น 1

delay(100);

T+=1;

}

}

3.20 void DetectBlack(){

if(sensorValue>=2400&&sensorValue<=2600)    ถ้าsensor สัมผัสเทปสีดำ

{ if(B%2==0){            ถ้า B หาร 2 ลงตัว รถจะถอยซ้าย

RTurnleft(speed);

```

delay(100);

if(!(sensorValue>=2400&&sensorValue<=2600 )){           และหาก sensor ไม่สัมผัสเทปสีดำ ให้เพิ่มค่า
B ขึ้น1

    B+=1;

}

}

else if(B%2==1){           ถ้า B หาร 2 เหลือเศษ 1 รถจะถอยขวา

    RTurnRight(speed);

    delay(100);

    if(!(sensorValue>=2400&&sensorValue<=2600 )){           และหาก sensor ไม่สัมผัสเทปสีดำ
ให้เพิ่มค่า B ขึ้น1

        B+=1;

    }

}

else{

    Forward(speed);           ถ้า sensor ไม่สัมผัสเทปสีดำ รถจะวิ่งไปข้างหน้า

}

}

```

```
}
```

```
3.21 void RDetectBlack(){
```

```
if(sensorValue>=2400&&sensorValue<=2600) ถ้า sensor สัมผัสเทปสีดำ
```

```
{ if(B%2==0){      ถ้า B หาร 2 ลงตัว รถเลี้ยวซ้าย
```

```
    Turnleft(speed);
```

```
    delay(100);
```

```
        if(!(sensorValue>=2400&&sensorValue<=2600 )){      และหาก sensor ไม่สัมผัสเทปสีดำ  
ให้เพิ่มค่า B ขึ้น1
```

```
        B+=1;
```

```
    }
```

```
}
```

```
else if(B%2==1){      ถ้า B หาร 2 เหลือเศษ 1 รถจะเลี้ยวขวา
```

```
    TurnRight(speed);
```

```
    delay(100);
```

```
        if(!(sensorValue>=2400&&sensorValue<=2600 )){      และหาก sensor ไม่สัมผัสเทปสีดำ  
ให้เพิ่มค่า B ขึ้น1
```

```
        B+=1;
```

```
    }
```

```
}
```

```
else{
```

```

    Reward(speed); ถ้า sensor ไม่สัมผัสเทปสีดำ รถจะถอยหลัง

}

}

}

```

ส่วนที่ 4 Function 3.22-3.23 จะเป็น Function เกี่ยวกับชุดคำสั่งกลยุทธฝ่ายรุก

```

3.22 void GoF(){      เปิดใช้ sensor 1,2,3 และsensor ได้ห้องรถ

    Sensor1();

    Sensor2();

    Sensor3();

    DetectRed();

    DetectBlack();

    if(T=1 && (sensorValue>=1700&&sensorValue<=1850))

    {

        GoR();      ถ้า T=1 และ sensor สัมผัสเทปสีแดง ให้ใช้ฟังก์ชัน GoR()

    }

}

```

```

3.23 void GoR(){

    Sensor4();      เปิดใช้ sensor 4,2,3 และsensor ได้ห้องรถ

    RSensor2();

```

```
RSensor3();
```

```
RDetectYellow();
```

```
RDetectBlack();
```

```
if(T=6)          ถ้า T=6 ให้รถหยุด
```

```
{
```

```
    Break();
```

```
}
```

```
}
```

ส่วนที่ 5 Function 3.24 จะเป็น Function เกี่ยวกับชุดคำสั่งกลยุทธฝ่ายรับ

```
3.24 void Defend(){
```

```
    DSensor1();   เปิดใช้ sensor หน้ารถและหลังรถ
```

```
    DSensor2();
```

```
}
```

4. loop

```
void loop() {
```

```
    if (SwitchState == HIGH)    ถ้า switch เปิด ให้ใช้ function GoF()
```

```
{
```

```
    GoF();
```

```
}
```

```
    if (SwitchState == LOW)    ถ้า switch ปิด ให้ใช้ function Defend()
```

```
{  
  
    Defend();  
  
}  
  
}
```



## ภาคผนวก

```
#include <HCSR04.h>

HCSR04 hc(10,11);

HCSR04 hc2(12,13); //initialisation class HCSR04 (trig,echo);

#define SwitchPin A3

#define ia1 2

#define ia2 3

#define ib1 4

#define ib2 5

#define ic1 6

#define ic2 7

#define id1 8

#define id2 9

#define S2 A2

#define S3 A1

#define SBt A0

#define maxSpd 255 // motor max speed

int SwitchState = 0;

int T=0;

int B=0;
```

```
int sensorValue = analogRead(A0);
```

```
int speed = maxSpd;
```

```
void setup() {
```

```
    pinMode(ia1, OUTPUT);
```

```
    pinMode(ia2, OUTPUT);
```

```
    pinMode(ib1, OUTPUT);
```

```
    pinMode(ib2, OUTPUT);
```

```
    pinMode(ic1, OUTPUT);
```

```
    pinMode(ic2, OUTPUT);
```

```
    pinMode(id1, OUTPUT);
```

```
    pinMode(id2, OUTPUT);
```

```
    pinMode(S2,INPUT);
```

```
    pinMode(S3,INPUT);
```

```
    pinMode(SBt,INPUT);
```

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

```
    pinMode(SwitchPin, INPUT);
```

```
}
```

```

void loop() {

    if (SwitchState == HIGH) //attack

    {

        GoF();

    }

    if (SwitchState == LOW) //defend

    {

        Defend();

    }

}

void Defend(){

    DSensor1();

    DSensor2();

}

void GoF(){

    Sensor1();

```

```

Sensor2();

Sensor3();

DetectRed();

DetectBlack();

if(T=1 && (sensorValue>=1700&&sensorValue<=1850)) //Red

{

    GoR();

}

}

void GoR(){

    Sensor4();

    RSensor2();

    RSensor3();

    RDetectYellow();

    RDetectBlack();

    if(T=6)

    {

        Break();

    }

}

```

```

void DetectRed(){

    if(sensorValue>=1700&&sensorValue<=1850)

        {

            Reward(speed);

            delay(100);

            T = 1;

        }

}

```

```

void RDetectYellow(){

    if(sensorValue>=1900&&sensorValue<=2100)

        {

            Reward(speed);

            delay(100);

            T+=1;

        }

}

```

```

void DetectBlack(){

```

```
if(sensorValue>=2400&&sensorValue<=2600) // detect black
```

```
{ if(B%2==0){
```

```
    RTurnleft(speed);
```

```
    delay(100);
```

```
    if(!(sensorValue>=2400&&sensorValue<=2600 )){
```

```
        B+=1;
```

```
    }
```

```
}
```

```
else if(B%2==1){
```

```
    RTurnRight(speed);
```

```
    delay(100);
```

```
    if(!(sensorValue>=2400&&sensorValue<=2600 )){
```

```
        B+=1;
```

```
    }
```

```
}
```

```
else{
```

```
    Forward(speed);
```

```
}
```

```

    }

}

void RDetectBlack(){

    if(sensorValue>=2400&&sensorValue<=2600) // detect black

    { if(B%2==0){

        Turnleft(speed);

        delay(100);

        if(!(sensorValue>=2400&&sensorValue<=2600 )){

            B+=1;

        }

    }

    else if(B%2==1){

        TurnRight(speed);

        delay(100);

        if(!(sensorValue>=2400&&sensorValue<=2600 )){

            B+=1;

        }

    }

    else{

        Reward(speed);
    }
}

```

```

    }

}

}

void Sensor1() //ultrasonic

{

    int speed = maxSpd;

    Serial.println(hc.dist() ); //showdistantfromfront

    if((hc.dist()>=10) && (hc.dist()<=15)){

        Slow(speed);}

    else if ((hc.dist()>=5) && (hc.dist()<10)){

        Break();}

    else if ((hc.dist()>=0) && (hc.dist()<5)){

        Reward(speed);}

    else{

        Forward(speed);}

}

void Sensor2(){

    int speed = maxSpd;

```



```

if(digitalRead(S2)==LOW){

    Serial.println("sensor detected object from left side");

    delay(20);

    Turnleft(speed);}

else{

    Forward(speed);

}

}

void RSensor2(){

    int speed = maxSpd;

    if(digitalRead(S2)==LOW){

        Serial.println("sensor detected object from left side");

        delay(20);

        RTurnRight(speed);}

    else{

        Reward(speed);

    }

}

void Sensor3(){

    int speed = maxSpd;

```

```

if(digitalRead(S3)==LOW){

    Serial.println("sensor detected object from right side");

    delay(20);

    TurnRight(speed);}

else{

    Forward(speed);

    }

}

void RSensor3(){

    int speed = maxSpd;

    if(digitalRead(S3)==LOW){

        Serial.println("sensor detected object from right side");

        delay(20);

        RTurnleft(speed);}

    else{

        Reward(speed);

        }

    }

void Sensor4() //ultrasonic

{

```

```

int speed = maxSpd;

Serial.println(hc2.dist() );

if((hc2.dist()>=10) && (hc2.dist()<=15)){

    RSlow(speed);}

else if ((hc2.dist()>=5) && (hc2.dist()<10)){

    Break();}

else if ((hc2.dist()>=0) && (hc2.dist()<5)){

    Forward(speed);}

else{

    Reward(speed);}

}

void DSensor1() //ultrasonic

{

int speed = maxSpd;

Serial.println(hc2.dist() );

if((hc2.dist()>=0) && (hc2.dist()<=15)){

    Forward(speed);}

else{

    Break();}

}

```

```

void DSensor2() //ultrasonic

{

    int speed = maxSpd;

    Serial.println(hc2.dist() );

    if((hc2.dist()>=0) && (hc2.dist()<=15)){

        Reward(speed);}

    else{

        Break();}

}

```

```

void Break() // motor break

{

    digitalWrite(ia1, HIGH);

    digitalWrite(ia2, HIGH);

    digitalWrite(ib1, HIGH);

    digitalWrite(ib2, HIGH);

    digitalWrite(ic1, HIGH);

    digitalWrite(ic2, HIGH);

    digitalWrite(id1, HIGH);

    digitalWrite(id2, HIGH);

```

```

}

void Forward(int speed) //goforward

{

    digitalWrite(ia1, LOW);

    analogWrite(ia2, speed);

    digitalWrite(ib1, LOW);

    analogWrite(ib2, speed);

    digitalWrite(ic1, LOW);

    analogWrite(ic2, speed);

    digitalWrite(id1, LOW);

    analogWrite(id2, speed);

}

void Slow(int speed) //goforwardwithSlowspeed

{

    digitalWrite(ia1, LOW);

    analogWrite(ia2, speed*0.3);

    digitalWrite(ib1, LOW);

    analogWrite(ib2, speed*0.3);

    digitalWrite(ic1, LOW);

    analogWrite(ic2, speed*0.3);

```

```

digitalWrite(id1, LOW);

analogWrite(id2, speed*0.3);

}

void RSlow(int speed) //RewardwithSlowspeed

{

digitalWrite(ia2, LOW);

analogWrite(ia1, speed*0.3);

digitalWrite(ib2, LOW);

analogWrite(ib1, speed*0.3);

digitalWrite(ic2, LOW);

analogWrite(ic1, speed*0.3);

digitalWrite(id2, LOW);

analogWrite(id1, speed*0.3);

}

void Reward(int speed) //goReward

{

digitalWrite(ia2, LOW);

analogWrite(ia1, speed);

digitalWrite(ib2, LOW);

```

```

    analogWrite(ib1, speed);

    digitalWrite(ic2, LOW);

    analogWrite(ic1, speed);

    digitalWrite(id2, LOW);

    analogWrite(id1, speed);

}

void Turnleft(int speed) //turnleftandgoforward

{

    digitalWrite(ia1, LOW);

    analogWrite(ia2, speed*0.7);

    digitalWrite(ib2, LOW);

    analogWrite(ib1, speed*0.3);

    digitalWrite(ic1, LOW);

    analogWrite(ic2, speed*0.5);

    digitalWrite(id1, LOW);

    analogWrite(id2, speed*0.5);

}

void TurnRight(int speed) //turnrightandgoforward

{

    digitalWrite(ia2, LOW);

```

```

    analogWrite(ia1, speed*0.3);

    digitalWrite(ib1, LOW);

    analogWrite(ib2, speed*0.7);

    digitalWrite(ic1, LOW);

    analogWrite(ic2, speed*0.5);

    digitalWrite(id1, LOW);

    analogWrite(id2, speed*0.5);

}

void RTurnLeft(int speed) //turnleftandReward

{

    digitalWrite(ia1, LOW);

    analogWrite(ia2, speed*0.5);

    digitalWrite(ib2, LOW);

    analogWrite(ib1, speed*0.5);

    digitalWrite(ic1, LOW);

    analogWrite(ic2, speed*0.7);

    digitalWrite(id1, LOW);

    analogWrite(id2, speed*0.3);

}

void RTurnRight(int speed) //turnrightandReward

```



```
{  
  
    digitalWrite(ia1, LOW);  
  
    analogWrite(ia2, speed*0.5);  
  
    digitalWrite(ib2, LOW);  
  
    analogWrite(ib1, speed*0.5);  
  
    digitalWrite(ic1, LOW);  
  
    analogWrite(ic2, speed*0.3);  
  
    digitalWrite(id1, LOW);  
  
    analogWrite(id2, speed*0.7);  
  
}
```

## บทที่ 4

### ผลการศึกษาโครงการ

จากการศึกษาในวิชา Pre-Project ที่ได้รับมอบหมายงานในการสร้างหุ่นยนต์รถ การจัดทำโครงการครั้งนี้ ผู้จัดทำได้มีความรู้ ความเข้าใจเกี่ยวกับการโปรแกรม วงจรไฟฟ้าขั้นพื้นฐานร่วมไปถึงความคิดสร้างสรรค์ในการออกแบบตัวโครงสร้างรถ สามารถทำให้รถเคลื่อนที่ได้ตามที่ผู้จัดทำต้องการ โดยมีขอบเขตงานและอุปกรณ์ที่ใช้อย่างจำกัด ซึ่งมีวิธีการดำเนินงานดังนี้

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

หมายเหตุ W1 เริ่มวันที่ 20 มกราคม 2563

ตารางที่ 3 ระยะเวลาและวิธีการดำเนินงาน

## บทที่ 5

### สรุปผลการศึกษาโครงการ

ในการจัดทำโครงการครั้งนี้เป็นการทำงานในรูปแบบออนไลน์ เนื่องจากเกิดสถานการณ์การระบาดของโรคติดเชื้อไวรัสโคโรนา-19 ทำให้ไม่สามารถสร้างรถขึ้นมาได้จริง โครงการที่จัดทำขึ้นครั้งนี้เป็นแนวคิดที่สามารถต่อยอดการสร้างหุ่นยนต์ขึ้นมาได้ในอนาคต การจัดทำโครงการครั้งนี้สมาชิกผู้จัดทำ ได้ติดต่อสื่อสารกันทางออนไลน์ทำให้ฝึกทักษะในการสื่อสารที่สมาชิกในกลุ่มจะต้องเข้าใจและทำงานไปในทิศทางเดียวกัน มีสมาธิในการทำงานมากขึ้นและที่สำคัญประหยัดค่าเดินทางไปทำงาน

#### 5.1 สรุปค่าใช้จ่ายในการทำโครงการ

ล้อรถ 2 ล้อ  $35 \times 2 = 70$  บาท

เซนเซอร์ 1 ตัว = 65 บาท (ultra sonic)

รวมประมาณ 135 บาท

## เอกสารอ้างอิง

- [1] <https://sites.google.com/site/mikhorkhxnthorllexr1/chud-kha-sang>
- [2] <https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html>
- [3] <https://www.arduitronics.com/article/22/arduino-and-motor-control-part-1>
- [4] <http://naringroup.blogspot.com/2016/03/robot-l298n-dual-h-bridge-motor.html>
- [5] <https://www.arduinoall.com/product/698/tcrt5000-infrared-reflectance-obstacle-avoidance-line-tracking-sensor->

