



## Assignment 4 Robot Car

รถหุ่นยนต์ (Robot car)

จัดทำโดย

นายชนสรณ์ ศิริวงศ์ รหัสนักศึกษา 62010153

วิชา 01076001 INTRODUCTION TO COMPUTER ENGINEERING

ภาคเรียนที่ 1 ปีการศึกษา 2566

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

# คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อศึกษาการสร้างรถหุ่นยนต์ที่สามารถเดินตามเส้นสีดำและสามารถตรวจจับไฟกระพริบได้โดยนำความรู้เรื่อง Line tracking มาประยุกต์ใช้และพัฒนาบนรถหุ่นยนต์ที่ควบคุมด้วยบอร์ดไมโครคอนโทรลเลอร์และเพื่อได้ฝึกอัลกอริทึมในการเขียนโปรแกรมเพื่อควบคุมมอเตอร์ซึ่งเป็นล้อของรถหุ่นยนต์ผ่านโมดูล L298N และสามารถตรวจจับไฟกระพริบได้

โดยผู้จัดทำหวังอย่างถึงที่สุดว่ารายงานฉบับนี้จะเป็นประโยชน์และเป็นแรงบันดาลใจให้แก่ผู้ที่กำลังศึกษาการสร้างรถหุ่นยนต์ที่สามารถเดินตามเส้นสีดำและสามารถตรวจจับไฟกระพริบได้

ผู้จัดทำ

นายชนสรณ์ ศิริวงศ์

# สารบัญ

	หน้า
อุปกรณ์ที่ใช้	1
ทฤษฎีที่เกี่ยวข้อง	6
การออกแบบและพัฒนา	8
ปัญหาและอุปสรรค	13
สรุปผลการดำเนินงาน	14
แหล่งอ้างอิง	15

## อุปกรณ์ที่ใช้

### 1. บอร์ด Arduino uno r3



ใช้สำหรับพัฒนาเกมส์และเป็น Hardware หัวใจสำหรับการเชื่อมต่อกับอุปกรณ์อื่นโดยภาษาที่ใช้พัฒนานั้นจะเป็นภาษา C++

โดย Arduino Uno R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ยอดนิยมซึ่งเป็นส่วนหนึ่งของแพลตฟอร์ม Arduino ซึ่งเป็นแพลตฟอร์มอิเล็กทรอนิกส์แบบโอเพ่นซอร์สที่ออกแบบมาสำหรับมือสมัครเล่น ผู้ผลิต และนักเรียนเพื่อสร้างโครงการอิเล็กทรอนิกส์เชิงโต้ตอบ

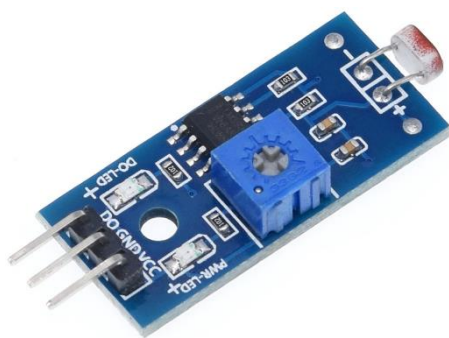
### 2. เซนเซอร์ TCRT 5000



เซนเซอร์ TCRT5000 เป็นเซนเซอร์ที่ใช้ตรวจจับวัตถุโดยใช้แสงอินฟราเรด โดยจะมี led แบบอินฟราเรดยิงแสงอินฟราเรดออกไป และมีตัวรับแสงอินฟราเรดรับค่าแสงที่สะท้อนกลับมา เมื่อวัตถุอยู่ใกล้จะมีแสง

สะท้อนกลับมากกว่าวัตถุที่อยู่ไกล จึงสามารถนำมาใช้วัดวัตถุผ่าน หรือใช้ตรวจจับเส้นสีขาว/ดำได้ โดยเส้นขาวจะให้แสงสะท้อนกลับมากกว่าสีดำ

### 3. เซนเซอร์ LDR Module



LDR Module เป็นโมดูลเซ็นเซอร์วัดความสว่างความเข้มแสง โดยใช้เซ็นเซอร์ LDR ในการตรวจจับ เมื่อมีการเปลี่ยนแปลงความเข้มแสงจะทำให้ความต้านทานเปลี่ยนแปลงตามความเข้มแสง โดยให้สัญญาณ Output ออกมา 2 แบบคือ

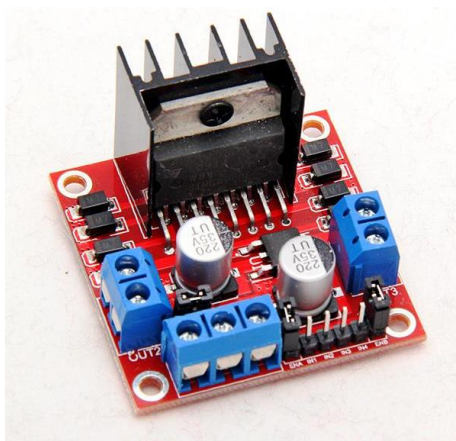
- Analog ระดับความเข้มของแสง ที่วัดได้เป็นค่า 0-1023
- Digital สัญญาณที่ได้คือ Logic 1 และ Logic 0 โดยหมุนตัว R ปรับค่าได้บนบอร์ดเพื่อตั้งระดับความต้องการของความเข้มแสงว่าจะให้สว่างเท่าใดจึงจะส่งค่าเอาต์พุตออกมา

### 4. USB Cable for Arduino UNO



สายเชื่อมต่อระหว่างบอร์ด Arduino และคอมพิวเตอรืเพื่ออัปโหลด Source code จากคอมพิวเตอรืลงบนบอร์ด Arduino

## 5. L298N Motor Driver Module



L298N Motor Driver Module เป็นโมดูลขับมอเตอร์ราคาถูก ใช้ชิพ L298N สามารถขับมอเตอร์ได้ 2 ตัวแบบแยกอิสระ สามารถควบคุมความเร็วมอเตอร์ได้ ใช้ไฟ 5 โวลต์ สามารถรับไฟเข้า 7-35 โวลต์ได้ มีวงจรเรกูเรเตอร์ในตัว ขับมอเตอร์กระแสสูงสุดได้ 2A

## 6. DC Motor



มอเตอร์ไฟฟ้ากระแสตรง หรือ ดี.ซี.มอเตอร์ (D.C. Motor) เป็นเครื่องกลชนิดหนึ่งที่เปลี่ยนจากพลังงานไฟฟ้าให้เป็นพลังงานกล โดยปกติมีขั้วต่อไฟฟ้าอยู่สองขั้ว คือ ขั้วบวกและขั้วลบ เมื่อต่อขั้ว ไฟฟ้าทั้ง

สองเข้าแบตเตอรี่โดยตรงมอเตอร์จะหมุนไปทางหนึ่ง แต่ถ้าสลับขั้วต่อไฟฟ้ากับแบตเตอรี่ มอเตอร์จะหมุนในทิศทางตรงกันข้าม

## 7. โครงรถ Robot Car



โครงรถ Robot car นี้จะประกอบไปด้วยโครงหุ่นยนต์ ล้อ และ DC motor ที่มาในตัว

## 8. ถ่านชาร์จ



ไว้สำหรับเป็นพลังงานให้รถหุ่นยนต์ทำงานโดยมีแรงดันเฉลี่ยอยู่ที่ 8 โวลต์

## 10. ไม้ไอติม



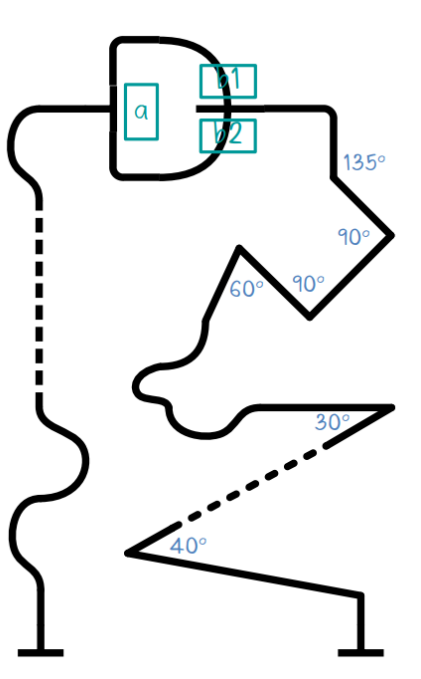
ไม้ไอติมไว้สำหรับเป็นส่วนเสริมเพิ่มเติมที่เอาไว้ยึดระหว่างโมดูล TCRT 5000 และตัวรถหุ่นยนต์



## ทฤษฎีที่เกี่ยวข้องและ Library

### 1. Line Tracking sensor

เป็นการอ่านค่าจากเซนเซอร์ TCRT 5000 ที่ได้จากสีขาวและสีดำทึบแล้วนำมาเก็บเป็นค่า error เพื่อนำมาใช้ในการบังคับตัวหุ่นยนต์ซึ่งค่า error ที่ได้จะมีตั้งแต่ -5,-4,-3 ไปเรื่อยๆจนถึง 5 เนื่องจากการเดินตามเส้นสีดำทึบนั่นไม่ได้มีแค่เส้นตรงเพียงอย่างเดียวยังมีการเข้าโค้ง การเลี้ยวหักศอกในองศาที่ต่างกัันดังนั้นจึงจำเป็นที่จะต้องเก็บค่า error มาวิเคราะห์อีกทีว่า ณ ปัจจุบันรอยู่ในตำแหน่งใดของเส้นทึบและจะไปได้ได้อย่างไรดังรูปภาพประกอบ



### 2. เทคนิคการบังคับหุ่นแบบ PID

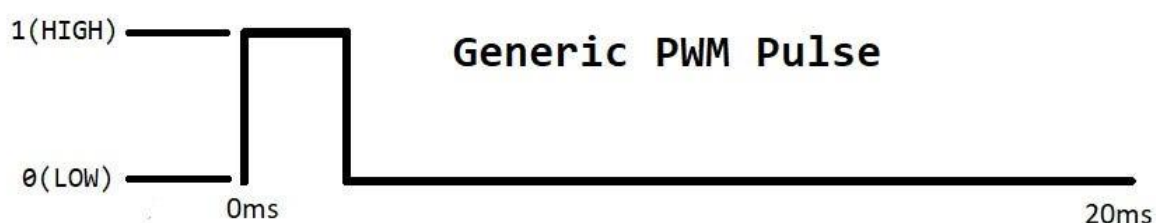
เป็นระบบควบคุมแบบป้อนกลับ ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หาได้จากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการเพื่อให้หุ่นยนต์อยู่ตรงกลางเส้นขณะวิ่ง

### 3. การตรวจจับไฟกระพริบ

เป็นการอ่านค่าอนาล็อกจาก Module LDR ว่าเมื่อมีไฟกระพริบจะมีค่าสัญญาณเป็นเท่าไรเมื่อเทียบกับตอนที่ยังไม่กระพริบเพื่อนำมาเขียนเงื่อนไขและอัลกอริทึมในการนับจำนวนครั้งที่ไฟกระพริบ

### 4. สัญญาณ PWM

Pulse Width Modulation(PWM) เป็นเทคนิคที่ทำให้สามารถอ่าน/เขียนข้อมูลแบบ analog ด้วยสัญญาณ digital ได้ โดยตัวควบคุมการสร้างสัญญาณดิจิทัล (Digital control) จะสร้างสัญญาณคลื่นสี่เหลี่ยมออกมามีรูปที่ 1 โดยสัญญาณที่สร้างออกมาจะสลับกันระหว่าง เปิด(HIGH) กับ ปิด(LOW) รูปแบบสัญญาณเปิด-ปิดนี้สามารถจำลองเป็นแรงดันไฟฟ้าระหว่าง เปิด (5 Volts) กับ ปิด (0 Volts)



เราเรียกช่วงที่เป็นปิดหรือเปิดตามการทำงาน เพราะช่วงที่เป็นปิดแรงดันไฟฟ้าจะเป็น 0 ทำให้ไม่มีการทำงาน ส่วนช่วงที่เป็นเปิดคือช่วงที่มีแรงดันไฟฟ้า ทำให้อุปกรณ์ทำงานได้

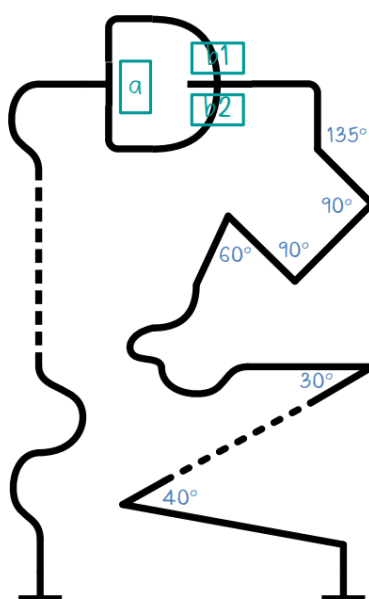
## การออกแบบและพัฒนา

เริ่มต้นด้วยการออกแบบถ้าเราบอกว่าเราต้องการสร้างรถหุ่นยนต์ที่สามารถเดินตามเส้นดำทึบและสามารถตรวจจับไฟกระพริบได้เช่นกัน โดยมีเงื่อนไขดังนี้

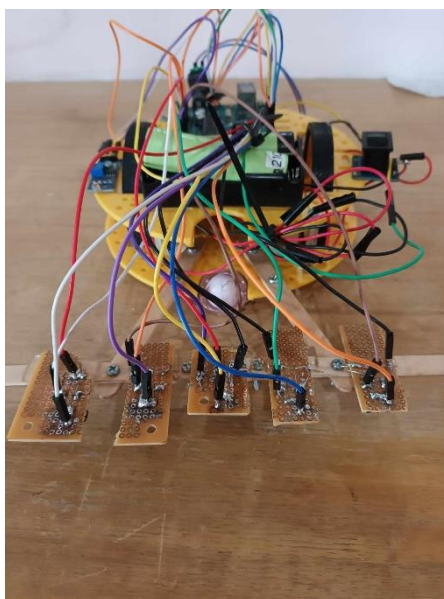
- 1) สามารถเดินตามเส้นตรง เส้นโค้ง มุม และเส้นประได้
- 2) สามารถหยุดรอสัญญาณไฟตรงแยก เพื่อเลี้ยวตามสัญญาณไฟ
  - กระพริบ 1 ครั้งให้ไปทางซ้าย
  - กระพริบ 2 ครั้งให้ไปทางขวา
- 3) การควบคุมหุ่นยนต์เป็นแบบอัตโนมัติ
  - เริ่มทำงานด้วยสวิตช์เปิด-ปิด อยู่ในจุดที่ใช้งานง่าย
  - วางส่วนประกอบให้มีความมั่นคงแข็งแรงสามารถอัปโหลดโปรแกรมได้ง่าย
  - จัดสายให้เรียบร้อยไม่ให้พันกันกับอุปกรณ์อื่นๆ

โดยสนามแข่งรถจะเป็นพื้นไวนิลสีขาวล้วนและมีเส้นสีดำทึบไว้สำหรับรถเดินตามเส้นซึ่งจะมีรายละเอียดดังนี้

- 1) สนามขนาดประมาณ 2X3 เมตร
- 2) เส้นหนาประมาณ 3 เซนติเมตร
- 3) มุมต่างๆในสนามจะมีตั้งแต่ 30,40,60,90 และ 135 องศา
- 4) จุดตั้งสัญญาณไฟจะมีทั้งหมด 3 จุดได้แก่ จุด A จุด b1 และจุด b2



มาถึงในส่วนของขั้นตอนการประกอบตัวรถหุ่นยนต์โดยในที่นี้นักศึกษาออกแบบให้วางบอร์ด Arduino uno r3 ไว้บนชั้นที่ 2 ของตัวรถ(ชั้นเดียวกันกับรางถ่าน) และ L298N ไว้ชั้นที่ 1 ของตัวรถ(ชั้นเดียวกันกับ DC Motor) และนำไม้ไอติม 3 อันมาเจาะรูแล้วนำไปยึดกับตัวรถเพื่อใช้เป็นฐานสำหรับวางโมดูล เซนเซอร์ TCRT 5000 และใช้กาวนาโนแบบเหนียวมากสำหรับยึดตัวโมดูลกับไม้ไอติมส่วนโครงไม้ไอติมจะใช้น็อตตัวผู้-เมียในการยึดแทนเนื่องจากมีความแข็งแรงกว่าดังรูป



ส่วนต่อมาเป็นในส่วนของการพัฒนาซอฟต์แวร์ร่วมกับฮาร์ดแวร์เริ่มแรกที่สุดสำหรับการพัฒนาในหัวข้อนี้ก็คือการ Calibrate นั่นเองซึ่งจะเป็นการทดสอบค่าที่สอดคล้องกันของฮาร์ดแวร์แต่ละตัวเพื่อเวลาใช้งานจริงจะได้ทำงานที่สอดคล้องและเข้ากันได้ยกตัวอย่างเช่นการ Calibrate มอเตอร์ซึ่งจริงๆแล้วมอเตอร์แต่ละตัวนั้นมักมีรอบหมุนที่ไม่เท่ากันต่อให้ใช้สัญญาณอินพุต PWM ที่เท่ากันก็ตาม

```

1 int enA = 10;
2 int enB = 6;
3 int in1 = 9;
4 int in2 = 8;
5 int in3 = 5;
6 int in4 = 4;
7
8 void setup()
9 {
10 // set all the motor control pins to outputs
11 pinMode(enA, OUTPUT);
12 pinMode(in1, OUTPUT);
13 pinMode(in2, OUTPUT);
14 pinMode(in3, OUTPUT);
15 pinMode(in4, OUTPUT);
16
17 }
18 void loop()
19 {
20
21 digitalWrite(in1, HIGH);
22 digitalWrite(in2, LOW);
23 digitalWrite(in3, HIGH);
24 digitalWrite(in4, LOW);
25 analogWrite(enA, 80);
26 analogWrite(enB, 100);

```

โดยรูปนี้เป็นตัวอย่างของการ Calibrate มอเตอร์ล้อทั้งสองของหุ่นยนต์สำหรับเดินหน้าตรงจะเห็นได้ว่าค่า analogWrite ที่ป้อนให้กับมอเตอร์ทั้งสองนั้นมีค่าไม่เท่ากัน

ส่วนต่อมาจะเป็นส่วนของการ Calibrate โมดูลเซนเซอร์ TCRT 5000 ซึ่งปกติแล้วค่า analogRead ของเซนเซอร์แต่ละตัวที่อ่านได้ย่อมไม่เท่ากันอยู่แล้วฉะนั้น บ่อยครั้งที่นักศึกษาต้องเข้าไปเก็บค่าจริงที่สนามจริงเพื่อนำมาพัฒนาต่อไป

```

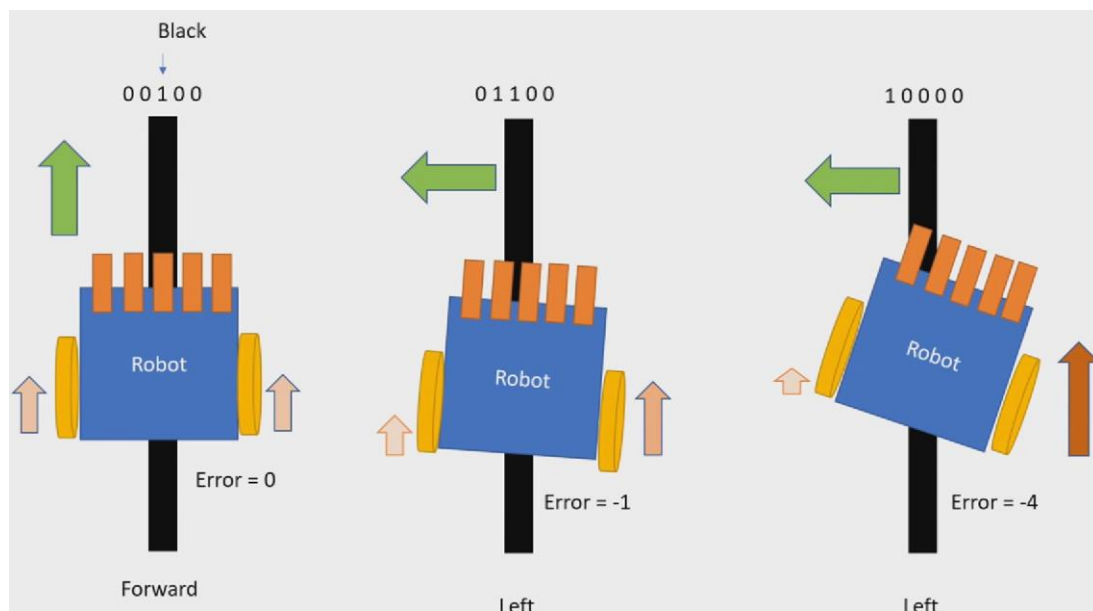
16
17 void loop() {
18   int ss0 = analogRead(sensor0); //200
19   int ss1 = analogRead(sensor1); //17
20   int ss2 = analogRead(sensor2); //250
21   int ss3 = analogRead(sensor3); //300
22   int ss4 = analogRead(sensor4); //500
23
24   Serial.print(ss0);
25   Serial.print(" ");
26   Serial.print(ss1);
27   Serial.print(" ");
28   Serial.print(ss2);
29   Serial.print(" ");
30   Serial.print(ss3);
31   Serial.print(" ");
32   Serial.println(ss4);
33   delay(500);
34 }

```

จากรูปนี้จะเห็นว่ามีการคอมเมนต์โค้ดด้วย “//” ซึ่งค่าที่อยู่หลังคอมเมนต์คือค่าที่โมดูลจับเส้นสีดำได้ ซึ่งจะเห็นได้ว่ามีค่าที่ไม่เท่ากันเลยนี้จึงเป็นเห็นผลหลักที่ทำไม่ก่อนการพัฒนาหุ่นยนต์เราจึงต้องทำการตรวจสอบค่าที่สอดคล้องกันของแต่ละฮาร์ดแวร์ด้วย

ส่วนต่อมาจะเป็นการพัฒนาแบบการเดินรถต่างๆที่นำค่า error ของเซนเซอร์มาคิดรวมด้วยโดยเทคนิคที่นักศึกษาใช้จะประยุกต์มาจากเทคนิค PID แต่ที่นักศึกษาไม่ใช่ PID เพราะการขับมอเตอร์ด้วยค่าต่างๆที่ส่งมาจาก Arduino นั้นไม่ค่อยต่างแบบมีนัยสำคัญเท่าไรจึงทำให้แม้จะจูนค่า PID มาดีเท่าไรรอบของการหมุนรถก็ไม่ค่อยต่างกันเนื่องจากกระแสที่ใช้ขับเคลื่อนล้อยมีไม่มากนักเพราะใช้ถ่านที่มีแรงดันเฉลี่ยอยู่ที่ 8 โวลต์ และยังต้องใช้ไฟไปเลี้ยงส่วนต่างๆของหุ่นยนต์อีกเช่น L298N หรือแม้กระทั่งโมดูลเซนเซอร์ TCRT 5000 นักศึกษาจึงตัดสินใจไม่ใช่เทคนิคดังกล่าวแต่หันมาปรับรอบหมุนและทิศทางของมอเตอร์เองโดยดูจากค่าที่ตรวจจับได้ของเซนเซอร์

โดยเมื่อเซนเซอร์จับเส้นสีดำได้เราจะให้ค่า error ของเซนเซอร์นั้นเป็น 0 และถ้าจับไม่ได้จะให้ค่าเป็น 1 ดังรูป แต่ในรูปจะสลับกันถ้าเจอสีดำจะเป็น 1 เจอสีขาวจะเป็น 0 แต่สุดท้ายแล้วใช้หลักการเดียวกัน



ที่นี้เราก็เขียนเงื่อนไขเลยถ้าเราเจอค่า error ที่ต่างกันไปเราจะให้มอเตอร์หมุนแบบไหนบ้างดังตัวอย่างโค้ด

```

else if ((ss1==0) && (ss4==1)) {                                     //0 1 1 1 1

    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(enA,150); //Left Motor Speed
    analogWrite(enB,140); //Right Motor Speed

}
else if ((ss0==1) && (ss1==1) && (ss2==0) && (ss3==1) && (ss4==1)) { //1 1 0 1 1
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enA,80); //Left Motor Speed 80
    analogWrite(enB,100); //Right Motor Speed80
}

```

จะเห็นได้ว่าเมื่อต่างเงื่อนไขกันเราก็จะให้เอาต์พุตที่ต่างกันออกไปเพื่อให้ตัวหุ่นยนต์สามารถเดินตามเส้นได้

มาในส่วนของขั้นตอนสุดท้ายคือการตรวจจับไฟกระพริบแน่นอนว่าเราจะต้องทำการ Calibrate โมดูล LDR เสียก่อนเพื่อจะได้รู้ว่าเมื่อมีไฟกระพริบโมดูลจะให้ค่าเท่าไรออกมาและตอนยังไม่กระพริบโมดูลจะให้ค่าเท่าไรออกมามาตรูป(โค้ดตัวอย่างการ Calibrate โมดูลเซนเซอร์ LDR)

```
void loop() {
    int inp = analogRead(A5);
    Serial.print(inp);
    Serial.print(" ");
    Serial.println(count);

    int start_time = millis();
    if(inp < 300 && toggle_count == true){//350
        count += 1;
        toggle_count = false;
    //    while(millis() - start_time < 3000){
    //        inp = analogRead(A5);
    //        if(inp > 500){
    //            toggle_count = true;
    //            break;
    //        }
    //    }
    }
    if(inp > 400){//380
        toggle_count = true;
    }
}
```

นี่เป็นตัวอย่างการ Calibrate และนับจำนวนครั้งที่ไฟกระพริบโดยจะเห็นได้ว่าเมื่อมีไฟกระพริบ ldr จะให้ค่าที่น้อยกว่าเท่ากับ 300 ออกมาแต่ถ้าไม่มีการกระพริบของไฟ ldr จะให้ค่าที่มากกว่า 400 ออกมา โดยรถจะหยุดรอรับสัญญาณไฟที่จุด A ตามในรูปสนามโดยจะหยุดก็ต่อเมื่อเซนเซอร์ทุกตัวอ่านค่าได้เท่ากับ 0 0 0 0 0 หรือก็คืออ่านสี่กำหนดนั่นเอง

### ปัญหาและอุปสรรค

1. ค่าที่เซนเซอร์ TCRT ส่งคืนกลับมามีค่าไม่เท่ากันตลอดทั้งสนาม  
วิธีแก้ไข: ทำการ Calibrate ค่าใหม่ที่ละเอียดกว่าเดิม
2. กระแสที่ใช้ขับเคลื่อนมอเตอร์ไม่เพียงพอทำให้ค่าความต่างของ PWM ไม่เห็นผลต่างเท่าที่ควร  
วิธีแก้ไข: กลับขั้วมอเตอร์สำหรับหมุนล้อเพื่อให้เกิดความต่างของ PWM ที่มากกว่า
3. แรงเสียดทานที่ล้อน้อยเกินไปทำให้เกิดการฟรีของล้อบ่อย  
วิธีแก้ไข: นำเชือกาวบางๆมาทาที่ล้อเพื่อเพิ่มสัมประสิทธิ์ความเสียดทาน



## สรุปผลการดำเนินงาน

การจำลองสร้างรถหุ่นยนต์ที่สามารถเดินตามเส้นสีดำและสามารถตรวจจับไฟกระพริบได้โดยนำความรู้เรื่อง Line tracking มาประยุกต์ใช้นั้นมีความยากพอสมควรเนื่องจากในทางปฏิบัติแล้วฮาร์ดที่นำมาใช้ย่อมเกิด Noise มากในระดับหนึ่งและรอบของการทดสอบแต่ละครั้งก็มีผลลัพธ์ที่ต่างกันทำให้การพัฒนาหุ่นยนต์หรือการเขียนโปรแกรมมีความยากพอสมควรที่จะจนค่าที่เหมาะสมให้กับหุ่นยนต์

สุดท้ายนี้ผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์และแรงบันดาลใจให้แก่ผู้ที่กำลังศึกษาการจำลองสร้างรถหุ่นยนต์ที่สามารถเดินตามเส้นสีดำและสามารถตรวจจับไฟกระพริบได้โดยนำความรู้เรื่อง Line tracking มาประยุกต์ใช้ร่วมด้วยพร้อมกับพัฒนาระบบบอร์ด Arduino uno r3

## แหล่งอ้างอิง

- สอนการใช้ PID ประยุกต์กับหุ่นยนต์เดินตามเส้นเบื้องต้น

<https://www.youtube.com/watch?v=zVbqXbINK68&list=LL&index=23&t=105s>

- พื้นฐานPIDง่ายๆ Ep.1

<https://www.youtube.com/watch?v=mfyYFsQEubc&list=LL&index=22>