

# Report

## Robot Car Contest

contributer

นายันทพงศ์	พงศ์ธำรงค์ดี	รหัสนักศึกษา	67011475
นายดิณณ์	สูงเมฆ	รหัสนักศึกษา	67011448

Introduction to Computer Engineering (ICE)  
Semester 1 Year 1 | Bachelor of Computer Engineering  
King Mongkut's Institute of Technology Ladkrabang, KMITL

---

### Table of Content

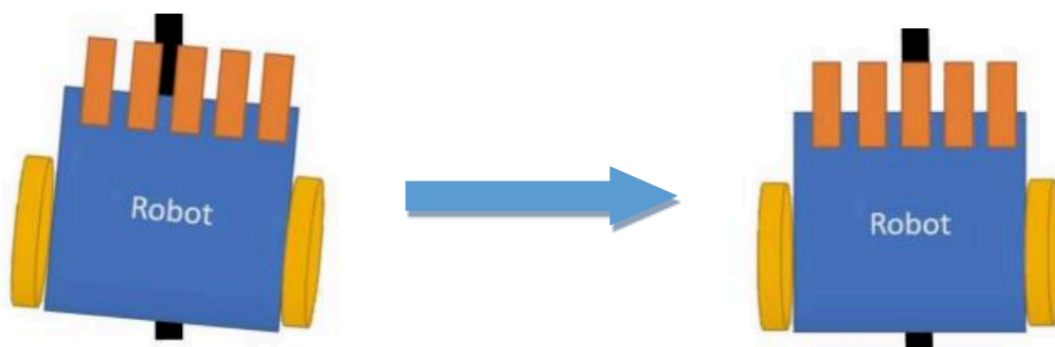
แนวคิดและหลักการทำงานของ Robot Car	2
การออกแบบ Hardware ของ Robot Car	3
Circuit Diagram ของ Robot Car	5
การออกแบบ Software Design ของ Robot Car	6
Source Code ของ Robot Car	10

## Robot Car Contest - Report

### แนวคิดและหลักการทำงานของ Robot Car

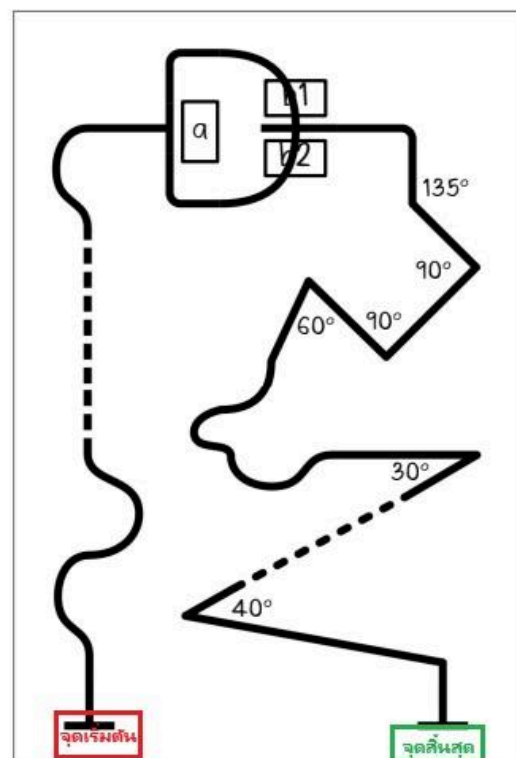
การทำงานของ Robot Car โดยสังเขป คือ หุ่นยนต์จะเคลื่อนที่ตามเส้นสีดำบนสนาม ซึ่งประกอบด้วยเส้นทึบและเส้นประ ในกรณีที่หุ่นยนต์เคลื่อนที่ออกนอกเส้นทาง (เกิดข้อผิดพลาด) ระบบจะพยายามปรับทิศทางกลับเข้าสู่เส้นทางเดิมโดยอัตโนมัติ หุ่นยนต์จะเริ่มเดินจากจุดเริ่มต้นจนไปถึงจุดสิ้นสุด

อีกหนึ่งฟังก์ชันการทำงานที่สำคัญคือ เมื่อหุ่นยนต์พบทางแยก จะหยุดและรอรับสัญญาณไฟ หากไฟกระพริบ 1 ครั้ง หุ่นยนต์จะเลี้ยวซ้าย และหากไฟกระพริบ 2 ครั้ง หุ่นยนต์จะเลี้ยวขวา



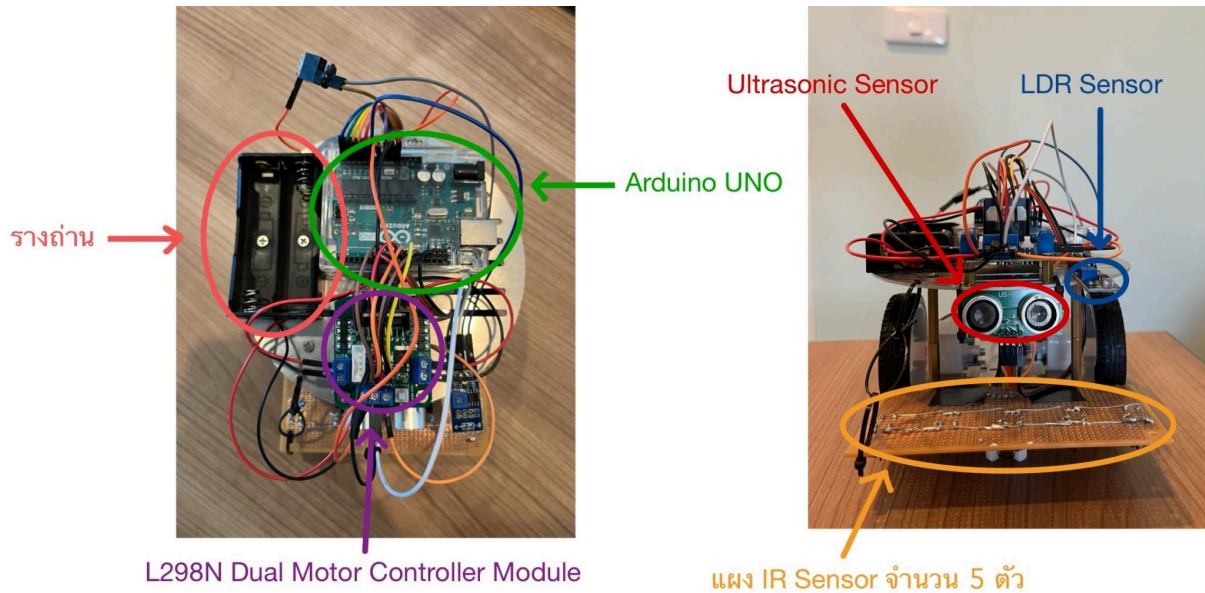
#### • Field Mission

- สนามขนาดประมาณ 2x3 m
- เส้นหนาประมาณ 3 cm
- มุมโดยประมาณ แสดงในสนาม
- จุดตั้งสัญญาณไฟ a และ b1 หรือ b2
  - ถ้า A กระพริบ 1 ครั้ง (ไปทางซ้าย)
    - ตั้งสัญญาณไฟที่ b2
  - ถ้า A กระพริบ 2 ครั้ง (ไปทางขวา)
    - ตั้งสัญญาณไฟที่ b1



## การออกแบบ Hardware ของ Robot Car

ในการออกแบบตัว Robot ได้ทำการจัดวางอุปกรณ์ต่างๆให้เหมาะสมได้แก่ บอร์ด Arduino UNO, Ultrasonic Sensor, IR Sensor (TCRT5000), LDR Sensor, L298N Dual Motor Controllerและรางถ่าน เพื่อให้เกิดความสมดุล

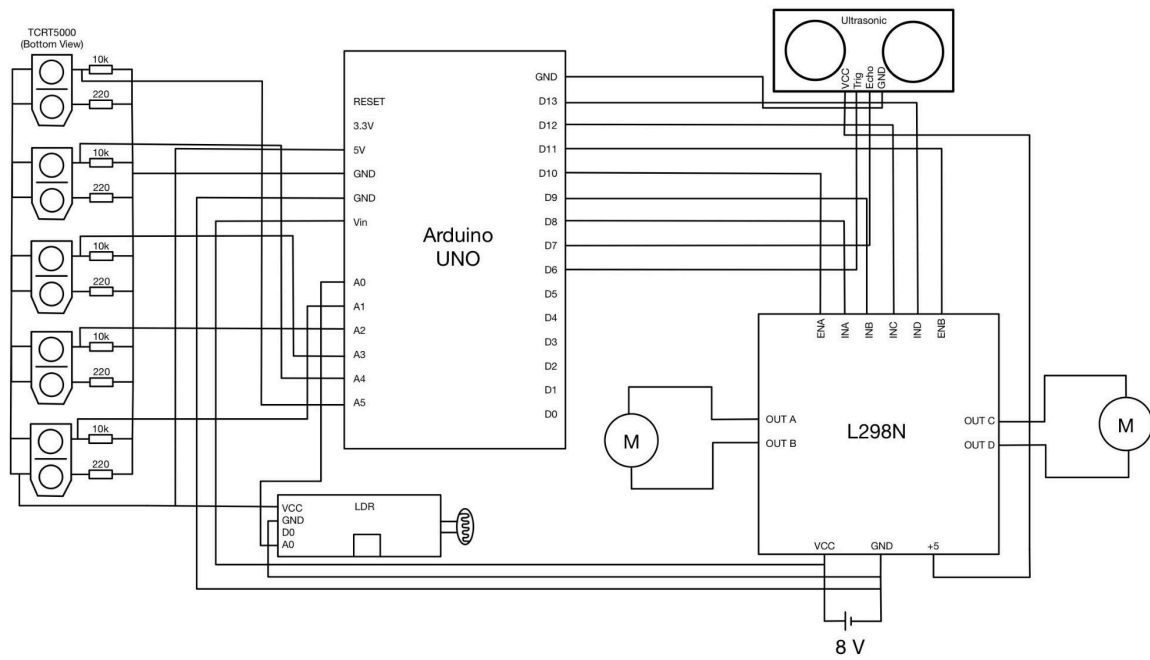


โดยอุปกรณ์ที่ใช้มีดังนี้

ลำดับ	ชื่ออุปกรณ์	จำนวนชิ้น
1	Robot car	1
2	Arduino UNO R3	1
3	Ultrasonic Sensor	1
4	LDR Sensor	1
5	IR Sensor	5
6	ตัวต้านทาน 220 โอห์ม	5
7	ตัวต้านทาน 10000 โอห์ม	5
8	บอร์ดไขปลา	1
9	Pin header	7
10	8 Pin Female Header PCB Connectors	1

ลำดับ	ชื่ออุปกรณ์	จำนวนชิ้น
11	L298N Dual Motor Controller Module	1
12	รางถ่าน	1
13	ถ่าน	2
14	DC Motor	2
15	Male to Male Cable Jumper	7
16	Male to Female Cable Jumper	20

## Circuit Diagram ของ Robot Car

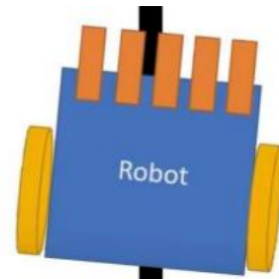


## การออกแบบ Software Design ของ Robot Car

### - กลไกการเดินตามเส้น

ใช้ IR Sensor ทั้งหมด 5 ตัว ในการอ่านข้อมูล เส้นบนสนาม ณ ขณะเวลา หนึ่ง ๆ โดยค่าเซ็นเซอร์ที่อ่านเส้นสีดำได้จะแตกต่างกันเนื่องจาก IR Sensor แต่ละตัวมีประสิทธิภาพแตกต่างกัน โดยเรียงจากซ้าย ไป ขวา ดังภาพ

IR sensor 1 < 600 จะพบเส้นดำ  
 IR sensor 2 < 650 จะพบเส้นดำ  
 IR sensor 3 < 800 จะพบเส้นดำ  
 IR sensor 4 < 600 จะพบเส้นดำ  
 IR sensor 5 < 650 จะพบเส้นดำ



เพื่อให้ง่ายต่อความเข้าใจได้สะดวก แทนเลข 1 เมื่อเซ็นเซอร์ เมื่อพบ เส้นดำ  
 แทนเลข 0 เมื่อเซ็นเซอร์ ไม่พบ เส้นดำ

โดยสามารถอธิบายพฤติกรรมการเดินทางโดยทั่วไปของ Robot car ได้ตามตารางนี้

IR SENSOR (จาก ซ้าย ไป ขวา)	พฤติกรรมของ Robot Car
0 0 1 0 0	เดินตรง
0 0 1 1 0	เอียงขวาเล็กน้อย
0 1 1 0 0	เอียงซ้ายเล็กน้อย
0 0 0 1 1	เอียงขวา
1 1 0 0 0	เอียงซ้าย
0 0 1 1 1	พบโค้งหักศอกทางขวา
1 1 1 0 0	พบโค้งหักศอกทางซ้าย
1 0 1 1 1	พบโค้งหักมุมทางขวา
1 1 1 0 1	พบโค้งหักมุมทางซ้าย
1 1 1 1 1	พบทางแยก
0 0 0 0 0	หลุดออกจากเส้นทาง

โดยที่พฤติกรรมของหุ่นที่แตกต่างกันในแต่ละกรณี จะนำไปสู่การตัดสินใจเดินที่แตกต่างกัน เช่น หาก หุ่นอยู่ในสถานะเดินตรงตามเส้น ก็จะพยายามเดินตรงตามเส้นต่อไป แต่หากในกรณีที่ หุ่นมีการเอียงออกจาก เส้น ก็จะพยายามเดินกลับเข้าสู่เส้น

**- การควบคุมทิศทางของ Robot Car**

ในการควบคุม Robot Car จะใช้ DC motor จำนวนสองตัว (ซ้ายและขวา) โดยสั่งงานผ่าน Motor Driver Module (L298N) โดยหากต้องการควบคุมให้ Robot Car เคลื่อนที่ไปในทิศทางที่ต่างกัน อาจมีหลักการในการสั่งงานโดยถ้าต้องการเลี้ยวจะให้มอเตอร์ ข้างใดข้างหนึ่งหมุนช้ากว่ามอเตอร์อีกข้างหนึ่งเพื่อให้การเลี้ยวมีประสิทธิภาพมากที่สุด โดยความเร็วปกติที่ Robot Car ตั้งไว้คือ 65

ทิศทาง	ความเร็ว DC motor (ซ้าย)	ความเร็ว DC motor (ขวา)	หมายเหตุ
ตรงไป	65	65	
เลี้ยวขวา แบบเลี้ยวปกติ	32.50	65	
เลี้ยวซ้าย แบบเลี้ยวปกติ	65	32.50	
เลี้ยวขวา แบบเลี้ยวเฉียบแหลม	21.67	65	
เลี้ยวซ้าย แบบเลี้ยวเฉียบแหลม	65	21.67	
เลี้ยวขวา แบบเลี้ยวเพิ่มกำลัง	21.67	75	
เลี้ยวซ้าย แบบเลี้ยวเพิ่มกำลัง	75	21.67	
หมุนซ้าย	65	65	มอเตอร์ขวาหมุนไปข้างหน้า มอเตอร์ซ้ายหมุนไปข้างหลัง
หมุนขวา	65	65	มอเตอร์ซ้ายหมุนไปข้างหน้า มอเตอร์ขวาหมุนไปข้างหลัง
หยุด	0	0	
ถอยหลัง	65	65	มอเตอร์ทั้งคู่หมุนถอยหลัง

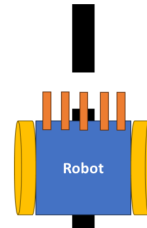
โดยในการสั่งงาน ได้สร้างฟังก์ชันการเลี้ยวแต่ละแบบไว้ เพื่อการสั่งงาน  
ซึ่งจะได้ลงรายละเอียด ต่อไปในการอธิบาย Source Code

### - Algorithm ในการจัดการกับเส้นประ

เนื่องจากในสนามแข่งมีเส้นที่เป็นลักษณะของเส้นประ ทำให้เมื่อตัว Robot Car เดินทางถึงตำแหน่ง เส้นประดังกล่าว ทำให้ IR Sensor ทั้งหมดตรวจจับเป็นสีขาว ทำให้เมื่อ Robot Car เจอกับเส้นประ ตัว Robot Car จะมองว่า หลุดออกนอกเส้นทาง จึงทำการถอยหลัง กลับจนกลับมาเจอเส้นสีดำ แล้วก็จะ เดินหน้าจนเจอสีขาวทั้งหมดอีก แล้วก็จะถอยหลังกลับมา เจอสีดำอีก วนแบบนี้ไปเรื่อย ๆ ทำให้ Robot Car ไม่สามารถผ่านเส้นประได้

จึงแก้โดยการเมื่อ IR Sensor ทั้งหมดไม่พบเส้นดำ

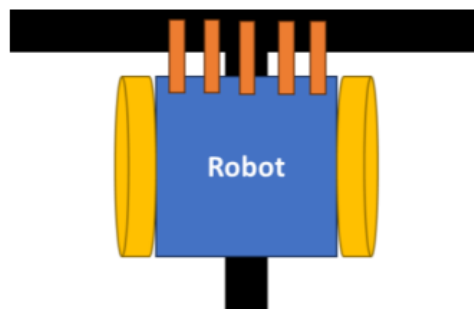
- 1.ให้ทำการเดินหน้าไปเป็นระยะเวลานึง
  - 2.ถ้าเดินไปแล้วตรวจพบเส้นดำให้เดินไปต่อแบบปกติ
- ถ้าหากไม่พบให้ทำการถอยหลังจนกว่าจะพบเส้นดำเพื่อให้เข้าสู่เส้นทางเดิม



### - Algorithm ในการจัดการกับสัญญาณไฟ

เมื่อ Robot Car อ่านค่าจาก IR Sensor ได้เป็นสีดำทั้งหมด Robot Car จะ detect ได้ว่าตอนนี้จำเป็นต้องรอรับสัญญาณไฟ เมื่อสัญญาณไฟกระพริบ 1 และ 2 ครั้ง Robot Car จะเลี้ยวไปทางซ้ายและขวาตามลำดับ

การตรวจสอบว่าสัญญาณไฟกระพริบ 1 หรือ 2 ครั้ง จะใช้ LDR Sensor ซึ่งตั้ง OUTPUT ไว้เป็น Analog โดยถ้ามีไฟเข้า LDR Sensor จะให้ค่าที่น้อยกว่า 600

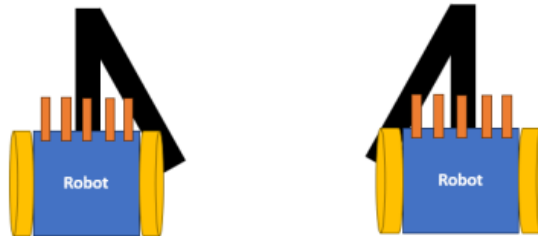


- 1.โดยถ้ามีไฟเข้าจะมาสร้างตัวแปร Temp\_light = 0 ขึ้นมา แล้วตั้งค่า Temp\_light = 1 แล้วรอเป็นระยะเวลา 8 วินาที
- 2.ถ้าระหว่างนั้นมีไฟเข้ามาอีกรอบ จะตั้ง Temp\_light = 2 พอดครบ 8 วินาที แล้วให้ทำการเลี้ยวตามเงื่อนไขที่ตั้งไว้
- 3.พอเลี้ยวแล้วจะให้ทำการลดกำลังเครื่องลงเป็นระยะเวลา 15 วินาที เพื่อให้รถสามารถจัดการหนักรถตัวเองเพื่อให้หยุดในทางแยกถัดไปได้อย่างพอดี
4. และ ในกรณีที่หากไม่มีไฟเข้ามา นั่นคือ Temp\_light = 0 จะให้ทำการถอยหลังแทนเพื่อให้รถสามารถเดินไปได้อย่างปกติ เพื่อป้องกันในกรณีเมื่อบังเอิญเซ็นเซอร์ทั้งหมดพบเส้นดำทั้งหมดแต่ไม่ใช่ทางแยก



### - Algorithm การเลี้ยวเมื่อถึงโค้งหักมุม

เมื่อเผชิญหน้ากับโค้งหักมุม บ่อยครั้งที่ Robot Car จะเกิดปัญหา ในการเลี้ยว คือ ไม่สามารถเลี้ยวได้ หรือบางครั้งถอยเข้าถอยออกจากฟังก์ชันที่จัดการเส้นประจนไม่สามารถเดินหน้าต่อไปได้ จึงคิด Algorithm ในการจัดการกับการเลี้ยวโค้งหักมุมขึ้น



เรียงลำดับ IR Sensor จากซ้ายไปขวา

โดย จะสังเกตว่า IR Sensor ตัวที่ 3 และ ตัวที่ 5 หรือ ตัวที่ 1 เมื่อเป็นกรณีเมื่อพบโค้งหักมุม

1. จะให้ทำการให้รถเดินไปข้างหน้าเป็นระยะเวลา 1 วินาที เพื่อให้ IR Sensor พ้นเส้นดำ
2. เมื่อครบ 1 วินาที แล้ว รถจะหมุนแล้ว หมุนจนกว่า IR Sensor ตัวที่ 1 หรือ ตัวที่ 5 พบเส้นดำ
3. เมื่อพบเส้นดำแล้วให้รถเดินตามปกติ

ซึ่งมีอัตราการสำเร็จมากกว่า 90 % ถ้าหากไม่สำเร็จ รถจะทำการไปข้างหน้าแล้วถอยหลังกลับตามฟังก์ชันแบบเดียวกับเส้นประ แก้ไขโดยการต้องรีเซ็ตการเดินรถใหม่

## Source Code ของ Robot Car

```

1 // LDR
2 const int ldr = A0;
3
4 // ULTRASONIC
5 const int trigPin = 6;
6 const int echoPin = 7;
7
8 // IR Sensors
9 const int IR1 = A1 ;
10 const int IR2 = A2 ;
11 const int IR3 = A3 ;
12 const int IR4 = A4 ;
13 const int IR5 = A5 ;
14
15 // Motor
16 const int enA = 10;
17 const int enB = 11;
18 const int in1 = 8; //left backward
19 const int in2 = 9 ; //left forward
20 const int in3 = 12 ; //right forward
21 const int in4 = 13; //right backward
22
23 //ตัวแปรที่ต้องใช้ในฟังก์ชันต่างๆ
24 unsigned long white_detect_startTime ;
25 unsigned long stop_startTime ;
26 unsigned long light_detect_startTime;
27 unsigned long intersection_detected_startTime;
28 unsigned long all_blackline_detected_mode_startTime;
29 bool intersection_detected = false;
30 long duration;
31 int distance;
32 long temp_light = 0;
33 int LDRvalue = 0 ;
34 int value_light_detect = 600;
35
36 //ตัวแปรควบคุมความเร็วของ มอเตอร์
37 int motorSpeed = 63;
38 int default_motorSpeed = 63;
39
40
41

```

ประกาศตัวแปรต่างๆที่ใช้การควบคุมหุ่นยนต์

```

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
}

```

setup pin ต่างๆที่ต้องใช้

```

void moveForward(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void moveBackward(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

void stopMotors() {
  analogWrite(enA, 0);
  analogWrite(enB, 0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

void turnLeft(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed/2);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void turnRight(int speed) {
  analogWrite(enA, speed/2);
  analogWrite(enB, speed);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

```

ฟังก์ชันรถเดินหน้า

ฟังก์ชันรถถอยหลัง

ฟังก์ชันรถหยุด

ฟังก์ชันรถเลี้ยวซ้าย แบบ เลี้ยวปกติ

ฟังก์ชันรถเลี้ยวขวา แบบ เลี้ยวปกติ

ใช้ในการกรณีที่เจอทางโค้งแบบปกติ

```

void turnLeftSharp(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed/3);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void turnRightSharp(int speed) {
  analogWrite(enA, speed/3);
  analogWrite(enB, speed);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void turnRightSuperSharp(int speed) {
  analogWrite(enA, speed/3);
  analogWrite(enB, speed+10);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void turnLeftSuperSharp(int speed) {
  analogWrite(enA, speed+10);
  analogWrite(enB, speed/3);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

```

ฟังก์ชันรถเลี้ยวซ้าย แบบ เลี้ยวเลียบแหลม

ฟังก์ชันรถเลี้ยวขวา แบบ เลี้ยวเลียบแหลม

ฟังก์ชันรถเลี้ยวขวา แบบ เลี้ยวเพิ่มกำลัง

ฟังก์ชันรถเลี้ยวซ้าย แบบ เลี้ยวเพิ่มกำลัง

ใช้ในการกรณีที่เจอทางโค้งแคบๆ

ใช้ในการกรณีที่เจอทางโค้งหักศอก 90 องศา

```
void rotateLeft(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void rotateRight(int speed) {
  analogWrite(enA, speed);
  analogWrite(enB, speed);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

int measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  int distance = duration * 0.034 / 2;
  return distance;
}
```

```

56 void loop() {
57     distance = measureDistance();
58     LDRvalue = analogRead(ldr);
59
60     if (distance < 15) {
61         stopMotors();
62     } else {
63         int sensor1 = analogRead(IR1), sensor2 = analogRead(IR2), sensor3 = analogRead(IR3), sensor4 = analogRead(IR4), sensor5 = analogRead(IR5);
64
65         if(sensor1 < 600 && sensor2 < 650 && sensor3 < 800 && sensor4 < 600 && sensor5 < 650){
66             stop_startTime = millis();
67             while(millis() - stop_startTime < 8000){
68                 stopMotors();
69                 LDRvalue = analogRead(ldr);
70                 sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
71                 if(LDRvalue < value_light_detect){
72                     light_detect_startTime = millis();
73                 } else {
74                     if(temp_light == 0){
75                         temp_light = 1;
76                         delay(2000);
77                         while(millis() - light_detect_startTime < 3000){
78                             LDRvalue = analogRead(ldr);
79                             if(LDRvalue < value_light_detect){
80                                 temp_light = 2;
81                             }
82                         }
83                     }
84                 }
85                 Serial.print(" temp light : "); Serial.println(temp_light);
86                 if(temp_light == 0){
87                     moveBackward(motorSpeed);
88                     delay(500);
89                 }
90                 else if(temp_light == 1){
91                     turnLeftSuperSharp(motorSpeed);
92                     delay(1000);
93                     motorSpeed = motorSpeed - 5;
94                     intersection_detected_startTime = millis();
95                     intersection_detected = true;
96                 } else if(temp_light == 2){
97                     turnRightSuperSharp(motorSpeed);
98                     delay(1000);
99                     motorSpeed = motorSpeed - 5;
100                    intersection_detected_startTime = millis();
101                    intersection_detected = true;
102                }
103                temp_light = 0;

```

ประกาศตัวแปร 2 ตัวแปร เหมือนกับ ตัวของบรรทัดจากฟังก์ชันแปลภาษาซีไปแทน และ เก็บค่า แสงที่รับจาก LDR Sensor ตามค่านี้

ถ้า distance ซึ่งมีค่าแปรผันกับระยะทางจากเซ็นเซอร์ได้มากกว่า 15 ซม (ขอจริงคือ 8 cm) ให้หยุดรถ ส่วนค่าที่น้อยกว่านั้น ให้ไปประมวลผลส่วนของการตรวจจับแสงของ IR 5 ตัว เพื่อทำการให้รถหลบตามเงื่อนไขที่กำหนด

ในส่วนนี้เป็นกรณี ที่ขณะตรวจจับรถจนเห็นตัวได้ไม่หมด จะให้ทำการหมุนเป็นเวลา 8 วินาที และตรวจว่ามีแสงเข้ามาไหม แต่ถ้ามีการตรวจจับได้ขณะตรวจจับรถจนเห็นตัวทั้งหมด จะทำการเปลี่ยนทางแยกหลบกัน

ขอเข้าสู่โหมดหลบหลีก ส่วนหลบรถชนเป็น 1 รอบ พัดรอบ 8 วินาที จะทำการเก็บค่าส่วน รอบชน 2 รอบ พัดรอบ 8 วินาที จะทำการเก็บค่า และขณะเปลี่ยนไปข้างหน้า จะทำการลดกำลังเครื่องยนต์เก็บค่ารอบเป็นเวลา 15 วินาที เพื่อให้รถสามารถคุมอยู่ในเลนได้ จะทำการควบคุมทิศทางแยกที่เข้าไปโดยควบคุมรอบ แต่ในการนี้ซึ่งมีเงื่อนไขให้รถชนรับ ชี้อาจจะเป็นกรณีที่เห็นแสงเข้าไปจนเห็นแล้วหลบไม่ได้โดยหากหลบจนกระทั่งการหมุนรถเป็น 8 วินาที แล้ว ถอยหลังเพื่อทำการจับทิศทางรถเดินหน้าต่อไปได้

```

} else if (sensor3 < 800 && sensor2 < 650 && sensor1 < 600) {
    sensor5 = analogRead(IR5); sensor4 = analogRead(IR4);
    while(sensor5 > 800 || sensor4 > 750){
        sensor5 = analogRead(IR5); sensor4 = analogRead(IR4);
        rotateRight(motorSpeed);
        if(sensor5 < 800 || sensor4 < 750) {
            return;
        }
    }
} else if( sensor3 < 800 && sensor4 < 750 && sensor5 < 650){
    sensor1 = analogRead(IR5); sensor2 = analogRead(IR2);
    while(sensor1 > 600 || sensor2 > 650){
        sensor1 = analogRead(IR5); sensor2 = analogRead(IR2);
        rotateLeft(motorSpeed);
        if(sensor1 < 600 || sensor2 < 650) {
            return;
        }
    }
} else if (sensor3 < 800 && sensor2 < 650) {
    turnLeftSuperSharp(motorSpeed);
} else if( sensor3 < 800 && sensor4 < 750){
    turnRightSuperSharp(motorSpeed);
} else if( sensor3 < 800 && sensor5 < 800 && sensor2 < 650){
    moveForward(motorSpeed);
    all_blackline_detected_mode_startTime = millis();
    while(millis() - all_blackline_detected_mode_startTime < 1000){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        if(sensor1 < 600 && sensor2 < 650 && sensor3 < 800 && sensor4 < 600 && sensor5 < 650){
            return;
        }
    }
}

sensor1 = analogRead(IR1);
while(sensor1 >= 800){
    sensor1 = analogRead(IR1);
    rotateRight(motorSpeed);
}
return;

```

ในกรณีเมื่อเซ็นเซอร์ตัวตรงกลาง ตัวซ้ายสุด และ ตัวระหว่างตัวตรงกลางและตัวซ้ายสุด พบเส้นดำ ซึ่งเป็นกรณีของไฟร์ลอสตอก 90 องศา แล้วจะทำการให้รถหมุนไปทางขวาจนกว่าเซ็นเซอร์ตัวทางซ้ายสุดและตัวระหว่างตัวตรงกลางและตัวซ้ายสุดพบสีดำให้ทำการหยุดหมุนแล้วคืนหน้าต่อไปอย่างปกติ

ในกรณีเมื่อเซ็นเซอร์ตัวตรงกลาง ตัวซ้ายสุด และ ตัวระหว่างตัวตรงกลางและตัวซ้ายสุด พบเส้นดำ ซึ่งเป็นกรณีของไฟร์ลอสตอก 90 องศา แล้วจะทำการให้รถหมุนไปทางขวาจนกว่าเซ็นเซอร์ตัวทางซ้ายสุดและตัวระหว่างตัวตรงกลางและตัวซ้ายสุดพบสีดำให้ทำการหยุดหมุนแล้วคืนหน้าต่อไปอย่างปกติ

ถ้าเซ็นเซอร์ตัวตรงกลาง และ ตัวที่อยู่ระหว่างตัวตรงกลางและตัวซ้ายสุด พบเส้นดำให้ทำการเลี้ยวขวาแบบเต็มกำลัง และ ถ้าเซ็นเซอร์ตัวตรงกลาง และ ตัวที่อยู่ระหว่างตัวตรงกลางและตัวซ้ายสุด พบเส้นดำให้ทำการเลี้ยวซ้ายแบบเต็มกำลัง

ถ้าเซ็นเซอร์ตัวตรงกลางและตัวซ้ายสุด และ ตัวระหว่างตัวตรงกลางและตัวซ้ายสุด พบเส้นดำ ซึ่งเป็นกรณีของไฟร์ลอสตอก จะทำให้ทำการเดินไปข้างหน้าเป็นเวลา 1 วินาที และระหว่าง 1 วินาที นั้นถ้าพบเส้นดำทั้งหมด จะให้ทำการกลับไปเดินแบบปกติ ถ้าไม่พบระหว่างนั้นแล้วครบ 1 วินาทีแล้ว จะให้รถเดินหน้าจนกว่าเซ็นเซอร์ตัวซ้ายสุดพบเส้นดำแล้วให้กลับไปโหมดการเดินแบบปกติ

```

} else if( sensor3 < 800 && sensor1 < 800 && sensor4 < 750){
    moveForward(motorSpeed);
    all_blackline_detected_mode_startTime = millis();
    while(millis() - all_blackline_detected_mode_startTime < 1000){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        if(sensor1 < 600 && sensor2 < 650 && sensor3 < 800 && sensor4 < 600 && sensor5 < 650){
            return;
        }
    }
    sensor5 = analogRead(IR5);
    while(sensor5 >= 800){
        sensor5 = analogRead(IR5);
        rotateLeft(motorSpeed);
    }
} else if( sensor3 < 800 && sensor1 < 600){
    moveForward(motorSpeed);
    all_blackline_detected_mode_startTime = millis();
    while(millis() - all_blackline_detected_mode_startTime < 1000){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        if(sensor1 < 600 && sensor2 < 650 && sensor3 < 800 && sensor4 < 600 && sensor5 < 650){
            return;
        }
    }
}

sensor1 = analogRead(IR1);
while(sensor1 >= 800){
    sensor1 = analogRead(IR1);
    rotateRight(motorSpeed);
}
return;
} else if( sensor3 < 800 && sensor5 < 800){
    moveForward(motorSpeed);
    all_blackline_detected_mode_startTime = millis();
    while(millis() - all_blackline_detected_mode_startTime < 1000){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        if(sensor1 < 600 && sensor2 < 650 && sensor3 < 800 && sensor4 < 600 && sensor5 < 650){
            return;
        }
    }
}

sensor5 = analogRead(IR5);
while(sensor5 >= 800){
    sensor5 = analogRead(IR5);
    rotateLeft(motorSpeed);
}
return;

```

ได้ทั้งหมดในภาพนี้ หลักการทำงานเหมือนกับ กรณีก่อนหน้านี้ ซึ่งใช้ในการกรณีเมื่อพบไฟร์ลอสตอกทั้งเส้น โดยสังเกต นั่นคือ เมื่อเซ็นเซอร์ตรวจจับทำการเดินไปข้างหน้าเป็นเวลา 1 นาที จากนั้นหมุนจนกว่าเซ็นเซอร์ตัวซ้ายสุดหรือตัวซ้ายสุดพบเส้นดำแล้วให้กลับไปโหมดการเดินแบบปกติ



```

} else if (sensor3 < 800) {
    moveForward(motorSpeed);
}
else if (sensor2 < 650) {
    turnLeft(motorSpeed);
} else if (sensor4 < 750) {
    turnRight(motorSpeed);
} else if (sensor1 < 600) {
    turnLeftSharp(motorSpeed);
} else if (sensor5 < 800) {
    turnRightSharp(motorSpeed);
} else if (sensor1 >= 600 && sensor2 >= 650 && sensor3 >= 800 && sensor4 >= 600 && sensor5 >= 650){
    white_detect_startTime = millis();
    while(millis() - white_detect_startTime <= 600){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        Serial.println("White Detect mode (forward)");
        moveForward(motorSpeed);
        if(sensor1 < 600 || sensor2 < 650 || sensor3 < 800 || sensor4 < 600 || sensor5 < 650){
            return;
        }
    }
    while(millis() - white_detect_startTime > 600){
        sensor1 = analogRead(IR1); sensor2 = analogRead(IR2); sensor3 = analogRead(IR3); sensor4 = analogRead(IR4); sensor5 = analogRead(IR5);
        Serial.println("White Detect mode (backward)");
        moveBackward(motorSpeed);
        if(sensor1 < 600 || sensor2 < 650 || sensor3 < 800 || sensor4 < 600 || sensor5 < 650){
            return;
        }
    }
}
if(intersection_detected == true && millis() - intersection_detected_startTime > 15000){
    motorSpeed = default motorSpeed;
    intersection_detected = false ;
}
}
delay(50);
}

```

ถ้าเซ็นเซอร์ตัวกลางพบเส้นดำ ให้ทำการเดินหน้า  
 ถ้าเซ็นเซอร์ตัวระหว่างตัวตรงกลางและตัวซ้ายสุด ให้ทำการเลี้ยวขวา แบบ เลี้ยวปกติ  
 ถ้าเซ็นเซอร์ตัวขวาสุดพบเส้นดำ ให้ทำการเลี้ยวขวา แบบ เลี้ยวเพิ่มกำลัง  
 ถ้าเซ็นเซอร์ตัวซ้ายสุดพบเส้นดำ ให้ทำการเลี้ยวซ้าย แบบ เลี้ยวเพิ่มกำลัง

ถ้าการเดินเซ็นเซอร์พบเส้นขาวทั้งหมด จะให้ทำการเดินหน้าเป็นเวลา 0.6 วินาที ถ้าเดินไปแล้วพบเส้นดำให้ทำการเดินหน้าต่อไป ถ้าไม่พบในเวลา 0.6 วินาที จะให้ทำการถอยหลังกลับ จนกว่าเซ็นเซอร์ตัวใดตัวหนึ่งจะเจอเส้นดำ ถ้าพบเส้นดำแล้วให้ทำการเดินหน้าต่อไปแบบปกติ

นี่คือฟังก์ชันต่อจากในหมวดรับแสงแล้วเลี้ยว พอเลี้ยวจะทำการลดกำลังเครื่องเป็นเวลา 15 วินาที ซึ่งนี่คือฟังก์ชันเมื่อครบ 15 วินาทีแล้วจะทำให้กำลังเครื่องมาค่าสูงเท่าเดิม