

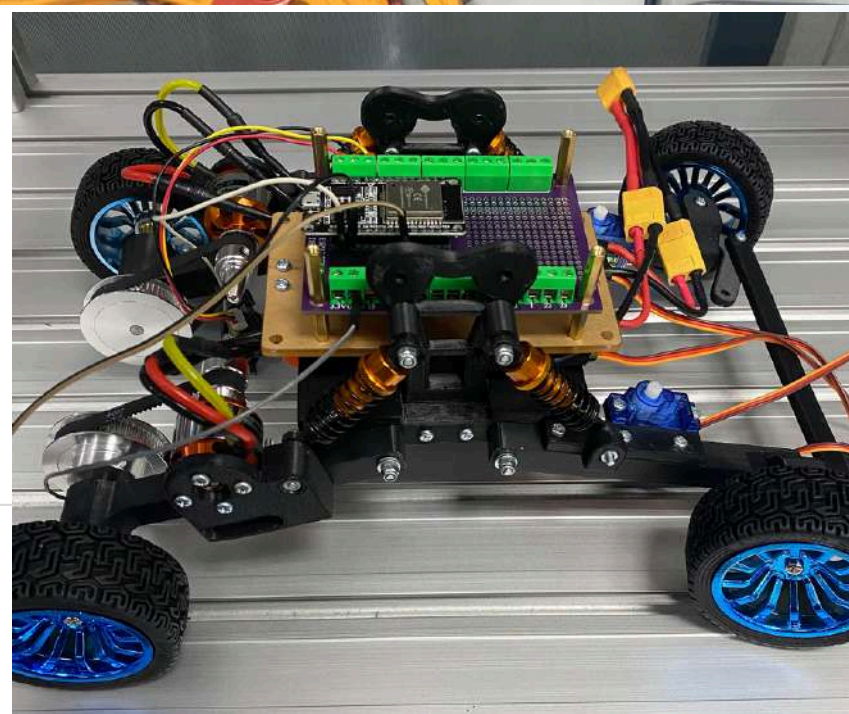
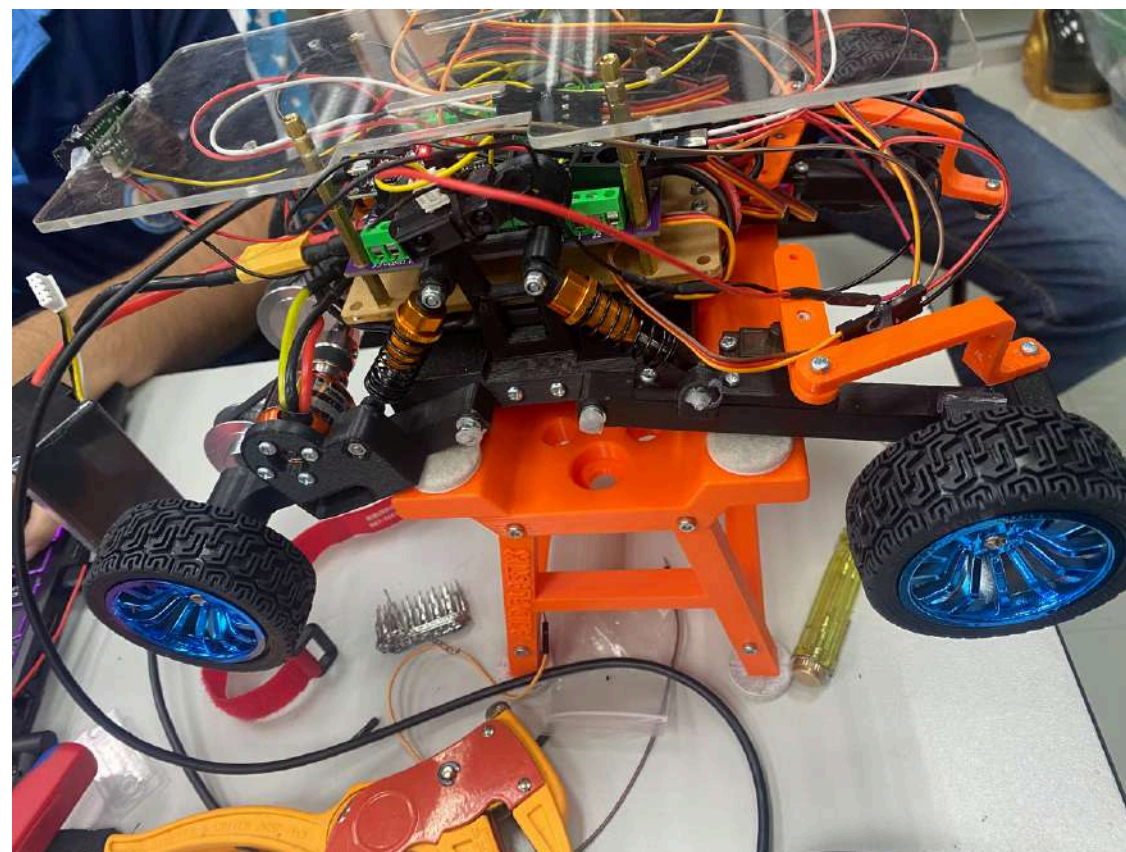


# PROJECT RC CAR

6552500017 จิตต์ศุภางค์ จริยพาณิชย์  
6552500076 พชรกฤษฎ์ พรหมมี



# Overview





```

1  #include <ESP32Servo.h>
2  #include "ESC.h"
3
4  int escPin1 = 18; // PWM pin for ESC 1 (BLDC 1)
5  int escPin2 = 19; // PWM pin for ESC 2 (BLDC 2)
6  int escPin3 = 22; // PWM pin for SG90 servo 1
7  int escPin4 = 21; // PWM pin for SG90 servo 2
8
9  // ตัวแปรสำหรับเซ็นเซอร์
10 int frontSensorPin = 34; // IR sensor for the front
11 int leftSensorPin = 35; // IR sensor for the left
12 int rightSensorPin = 32; // IR sensor for the right
13
14 // ประกาศตัวแปรสำหรับการควบคุม BLDC และ SG90
15 Servo esc1;
16 Servo esc2;
17 Servo esc3;
18 Servo esc4;
19
20 int pwmTarget = 1060; // ตั้งความเร็วสูงสุด
21 int pwmIncrement = 0.03; // ตั้งค่าความเร่งเพื่อให้รถวิ่งด้วยความ smooth
22 unsigned long previousMillis = 0;
23
24 const unsigned long pwmInterval = 50;
25 const int frontThreshold = 99; // กำหนด irfront ในระยะ 99 cm
26 const int leftSensorThreshold = 30; // กำหนด irleft ในระยะ 30 cm
27 const int rightSensorThreshold = 30; // กำหนด irright ในระยะ 30 cm
28
29 #define SENSOR_HISTORY_SIZE 5 // ลดขนาด Sliding Window 5

```

โค้ดนี้ถูกเขียนขึ้นเพื่อควบคุมรถหุ่นยนต์ที่มี มอเตอร์ BLDC และ เซอร์โวมอเตอร์ SG90 โดยใช้ ESP32 เพื่อควบคุมการเคลื่อนไหวยและหลบหลีกสิ่งกีดขวางที่ตรวจจับได้จาก เซ็นเซอร์ IR ที่ติดตั้งไว้ด้านหน้า ซ้าย และขวา โค้ดมีรายละเอียดดังนี้:

#### 1. การตั้งค่าพิน

- escPin1 ถึง escPin4: พินที่ใช้ควบคุมมอเตอร์ BLDC (escPin1, escPin2) และเซอร์โวมอเตอร์ SG90 (escPin3, escPin4)
- frontSensorPin, leftSensorPin, rightSensorPin: พินที่เชื่อมต่อกับเซ็นเซอร์ IR

#### 2. การประกาศตัวแปร

- pwmTarget: ความเร็วเป้าหมายของมอเตอร์ BLDC
- pwmIncrement: กำหนดความเร่งให้รถวิ่งอย่างราบรื่น
- SENSOR\_HISTORY\_SIZE: ขนาดของ Sliding Window เพื่อเก็บประวัติค่าจากเซ็นเซอร์
- frontSum, leftSum, rightSum: เก็บผลรวมของค่าประวัติจากเซ็นเซอร์แต่ละตัว



```

#define SENSOR_HISTORY_SIZE 5 // ลดขนาด Sliding Window 5

int frontSensorHistory[SENSOR_HISTORY_SIZE] = {0};
int leftSensorHistory[SENSOR_HISTORY_SIZE] = {0};
int rightSensorHistory[SENSOR_HISTORY_SIZE] = {0};
int historyIndex = 0;
int frontSum = 0, leftSum = 0, rightSum = 0;

// ฟังก์ชันการกรองค่าของเซ็นเซอร์ด้วย Moving Average
int filterSensorRead(int pin, int* sensorHistory, int& sum) {
    int newValue = analogRead(pin); // อ่านค่าจากเซ็นเซอร์

    // หักค่าที่จะถูกแทนที่ออกจากผลรวม
    sum -= sensorHistory[historyIndex];

    // อัปเดตค่าใหม่ในประวัติและเพิ่มเข้าผลรวม
    sensorHistory[historyIndex] = newValue;
    sum += newValue;

    // หมุนตำแหน่ง index
    historyIndex = (historyIndex + 1) % SENSOR_HISTORY_SIZE;

    // คำนวณค่าเฉลี่ย
    return sum / SENSOR_HISTORY_SIZE;
}

```

3. การกรองข้อมูลเซ็นเซอร์ด้วย Moving Average  
ฟังก์ชัน filterSensorRead() ใช้ Moving Average เพื่อกรองค่าจากเซ็นเซอร์:
1. อ่านค่าจากเซ็นเซอร์ (ผ่าน analogRead()).
  2. ปรับปรุงผลรวมโดยลบค่าเก่าที่จะถูกแทนที่ และเพิ่มค่าใหม่เข้าไป.
  3. คำนวณค่าเฉลี่ยโดยหารผลรวมด้วยขนาดของ Sliding Window.

```

void setup() {
  // Attach ESCs และเซอร์โว
  esc1.attach(escPin1);
  esc2.attach(escPin2);
  esc3.attach(escPin3);
  esc4.attach(escPin4);

  esc1.writeMicroseconds(1000); // ตั้งค่า BLDC ให้หยุดด้วย 1000 (1000 = หยุด)
  esc2.writeMicroseconds(1000); // ตั้งค่า BLDC ให้หยุดด้วย 1000 (1000 = หยุด)
  esc3.write(90); // ตั้งค่า sg90 ที่ 90 องศา
  esc4.write(90); // ตั้งค่า sg90 ที่ 90 องศา
  pinMode(frontSensorPin, INPUT);
  pinMode(leftSensorPin, INPUT);
  pinMode(rightSensorPin, INPUT);

  // setCpuFrequency(240);
  Serial.begin(115200);
  delay(2000);

  // สร้าง function ในหาค่า average
  for (int i = 0; i < SENSOR_HISTORY_SIZE; i++) {
    frontSensorHistory[i] = analogRead(frontSensorPin); // อ่านค่าเซ็นเซอร์หน้า 5 ครั้งแรก
    leftSensorHistory[i] = analogRead(leftSensorPin); // อ่านค่าเซ็นเซอร์ซ้าย 5 ครั้งแรก
    rightSensorHistory[i] = analogRead(rightSensorPin); // อ่านค่าเซ็นเซอร์ขวา 5 ครั้งแรก
    frontSum += frontSensorHistory[i];
    leftSum += leftSensorHistory[i];
    rightSum += rightSensorHistory[i];
    delay(10);
  }
}

```

#### 4. การตั้งค่าในฟังก์ชัน setup()

1. เชื่อมต่อ ESC และเซอร์โวด้วย attach().
2. กำหนดค่ามอเตอร์ BLDC ให้หยุด (1000 ไมโครวินาที).
3. กำหนดเซอร์โวให้อยู่ในตำแหน่งเริ่มต้น (90 องศา).
4. อ่านค่าเซ็นเซอร์เบื้องต้นเพื่อตั้งค่าประวัติสำหรับ Moving Average.



```

void loop() {
    unsigned long currentMillis = millis();

    // ใช้ฟังก์ชันกรองค่าแยกกันสำหรับแต่ละเซ็นเซอร์
    int frontDistance = filterSensorRead(frontSensorPin, frontSensorHistory, frontSum);
    int leftDistance = filterSensorRead(leftSensorPin, leftSensorHistory, leftSum);
    int rightDistance = filterSensorRead(rightSensorPin, rightSensorHistory, rightSum);
    Serial.print(frontDistance);
    Serial.print(" : ");

    // ตั้งค่า map เพื่อแปลงค่า analog เป็นหน่วย cm
    int frontDistance_G = map(frontDistance, 0, 4095, 100, 0);
    int leftDistance_G = map(leftDistance, 0, 4000, 100, 0);
    int rightDistance_G = map(rightDistance, 0, 4000, 100, 0);

    frontDistance_G = constrain(frontDistance_G, 0, 100);
    leftDistance_G = constrain(leftDistance_G, 0, 100);
    rightDistance_G = constrain(rightDistance_G, 0, 100);

```

## 5. ฟังก์ชัน loop()

1. อ่านค่าจากเซ็นเซอร์ทั้ง 3 ตัว (ด้านหน้า, ซ้าย, ขวา) ผ่าน filterSensorRead() และแปลงค่าเป็นหน่วยเซนติเมตรด้วย map().
2. ตรวจสอบเงื่อนไขการทำงาน:
  - กรณีมีสิ่งกีดขวางด้านหน้า:
    - หยุดมอเตอร์ BLDC (esc1 และ esc2 หยุดที่ 1000 ไมโครวินาที).
    - หมุนเซอร์โวลให้หลบหลีก (45 องศา และ 135 องศา).
  - กรณีมีสิ่งกีดขวางด้านซ้าย:
    - หมุนเซอร์โวลให้เลี้ยวขวา (55 องศา).
  - กรณีมีสิ่งกีดขวางด้านขวา:
    - หมุนเซอร์โวลให้เลี้ยวซ้าย (125 องศา).
  - กรณีไม่มีสิ่งกีดขวาง:
    - ขับมอเตอร์ BLDC ไปข้างหน้าด้วยความเร็วเป้าหมาย.
    - เซอร์โวลปรับกลับไปที่ตำแหน่งตรงกลาง (90 องศา).

## 6.

1. PWM กับ ESC:
  - ใช้ writeMicroseconds() เพื่อควบคุม BLDC (1000 = หยุด, 1500 = กลาง, 2000 = เร็วสุด).
  - ใช้ write() สำหรับเซอร์โวล SG90 เพื่อกำหนดองศาการหมุน.
2. การกรองเซ็นเซอร์:
3. การใช้ Sliding Window (Moving Average) ช่วยลดสัญญาณรบกวนจากเซ็นเซอร์ IR.
4. การแปลงค่า (Mapping):
5. ใช้ map() เพื่อแปลงค่า ADC (0-4095) เป็นระยะทาง (0-100 เซนติเมตร).
6. การหน่วงเวลา (delay()):
7. ใช้ดีเลย์เล็กน้อย (30 มิลลิวินาที) เพื่อให้การเคลื่อนไหวมีเสถียรภาพ.



```
// Serial.print(" ");  
Serial.print("Front Distance: ");  
Serial.print(frontDistance_G);  
Serial.print(" cm : ");  
Serial.print("left Distance: ");  
Serial.print(leftDistance_G);  
Serial.print(" cm : ");  
Serial.print("right Distance: ");  
Serial.print(rightDistance_G);  
Serial.print(" cm ");
```

คำสั่งและการทำงาน:

- 1.Serial.print("Front Distance: ");
  - พิมพ์ข้อความ "Front Distance: " ลงใน Serial Monitor เพื่อระบุว่าในบรรทัดนี้จะแสดงผลการวัดระยะห่างจากเซ็นเซอร์ด้านหน้า
- 2.Serial.print(frontDistance\_G);
  - พิมพ์ค่าระยะห่างจากเซ็นเซอร์ด้านหน้าที่ได้จากการคำนวณและแปลง (ผ่านฟังก์ชัน map() และ constrain() เพื่อให้เป็นหน่วยเซนติเมตร) ซึ่งเก็บอยู่ในตัวแปร frontDistance\_G
- 3.Serial.print(" cm : ");
  - พิมพ์ข้อความ " cm : " เพื่อแสดงหน่วยของระยะห่างเป็นเซนติเมตร (cm) และเว้นช่องว่างเพื่อให้อ่านง่าย
- 4.Serial.print("left Distance: ");
  - พิมพ์ข้อความ "left Distance: " ลงใน Serial Monitor เพื่อระบุว่าในบรรทัดถัดไปจะแสดงผลการวัดระยะห่างจากเซ็นเซอร์ด้านซ้าย
- 5.Serial.print(leftDistance\_G);
  - พิมพ์ค่าระยะห่างจากเซ็นเซอร์ด้านซ้ายที่ได้จากการคำนวณและแปลงเก็บไว้ในตัวแปร leftDistance\_G
- 6.Serial.print(" cm : ");
  - พิมพ์ข้อความ " cm : " เพื่อแสดงหน่วยของระยะห่างเป็นเซนติเมตร (cm) และเว้นช่องว่างเพื่อให้อ่านง่าย
- 7.Serial.print("right Distance: ");
  - พิมพ์ข้อความ "right Distance: " ลงใน Serial Monitor เพื่อระบุว่าในบรรทัดถัดไปจะแสดงผลการวัดระยะห่างจากเซ็นเซอร์ด้านขวา
- 8.Serial.print(rightDistance\_G);
  - พิมพ์ค่าระยะห่างจากเซ็นเซอร์ด้านขวาที่ได้จากการคำนวณและแปลงเก็บไว้ในตัวแปร rightDistance\_G
- 9.Serial.print(" cm ");
  - พิมพ์ข้อความ " cm" เพื่อแสดงหน่วยของระยะห่างเป็นเซนติเมตร (cm) ตามด้วยช่องว่างจำนวนมาก เพื่อให้การแสดงผลดูสะอาดตา



```

if (frontDistance_G < frontThreshold) {
  // Stop if an object is detected in front
  esc1.writeMicroseconds(1000);
  esc2.writeMicroseconds(1000);
  esc3.write(45); // Center position 90
  esc4.write(135); // Center position 90
  Serial.println("Stopping - Obstacle ahead");
  // Serial.println("");
} else if (leftDistance_G < leftSensorThreshold) {
  // Turn right if an obstacle is detected on the left
  esc3.write(55); // SG90 servo 1: Turn to 65 degrees
  esc4.write(55); // SG90 servo 2: Turn to 65 degrees
  Serial.println("Turning right - Obstacle on left");
  // Serial.println("");
} else if (rightDistance_G < rightSensorThreshold) {
  // Turn left if an obstacle is detected on the right
  esc3.write(125); // SG90 servo 1: Turn to 115 degrees
  esc4.write(125); // SG90 servo 2: Turn to 115 degrees
  Serial.println("Turning left - Obstacle on right");
  // Serial.println("");
} else {

  esc3.write(90); // Center position 90
  esc4.write(90); // Center position 90
  esc1.writeMicroseconds(pwmTarget);
  esc2.writeMicroseconds(pwmTarget);
  Serial.println("Moving straight");
}

```

```

delay(30);

```

1. กรณีมีสิ่งกีดขวางด้านหน้า (frontDistance\_G < frontThreshold)

- เช็คนไข: ถ้าระยะห่างจากเซ็นเซอร์ด้านหน้าต่ำกว่าค่าที่ตั้งไว้ใน frontThreshold (70 ซม. ตามโค้ด)
- การกระทำ:
  - หยุดมอเตอร์ BLDC ที่ใช้พิน esc1 และ esc2 โดยการตั้งค่าให้มอเตอร์หยุด (ค่า PWM = 1000) ด้วย esc1.writeMicroseconds(1000) และ esc2.writeMicroseconds(1000)
  - ปรับตำแหน่งของเซอร์โว SG90 ด้วย esc3.write(45) และ esc4.write(135) เพื่อให้หุ่นยนต์เริ่มหลบสิ่งกีดขวาง (เซอร์โวถูกหมุนไปตำแหน่ง 45 องศา และ 135 องศา)
  - พิมพ์ข้อความ "Stopping - Obstacle ahead" ลงใน Serial Monitor เพื่อแจ้งเตือนว่ามีสิ่งกีดขวางอยู่ด้านหน้า

2. กรณีมีสิ่งกีดขวางทางซ้าย (leftDistance\_G < leftSensorThreshold)

- เช็คนไข: ถ้าระยะห่างจากเซ็นเซอร์ด้านซ้ายต่ำกว่าค่าที่ตั้งไว้ใน leftSensorThreshold (30 ซม. ตามโค้ด)
- การกระทำ:
  - หมุนเซอร์โว SG90 เพื่อเลี้ยวขวา (เพื่อหลบสิ่งกีดขวางทางซ้าย) โดยการตั้งค่าเซอร์โวทั้งสองตัว (esc3.write(55) และ esc4.write(55)) เพื่อให้เซอร์โวหมุนไปที่ 55 องศา
  - พิมพ์ข้อความ "Turning right - Obstacle on left" ลงใน Serial Monitor

3. กรณีมีสิ่งกีดขวางทางขวา (rightDistance\_G < rightSensorThreshold)

- เช็คนไข: ถ้าระยะห่างจากเซ็นเซอร์ด้านขวาต่ำกว่าค่าที่ตั้งไว้ใน rightSensorThreshold (30 ซม. ตามโค้ด)
- การกระทำ:
  - หมุนเซอร์โว SG90 เพื่อเลี้ยวซ้าย (เพื่อหลบสิ่งกีดขวางทางขวา) โดยการตั้งค่าเซอร์โวทั้งสองตัว (esc3.write(125) และ esc4.write(125)) เพื่อให้เซอร์โวหมุนไปที่ 125 องศา
  - พิมพ์ข้อความ "Turning left - Obstacle on right" ลงใน Serial Monitor

4. กรณีไม่มีสิ่งกีดขวาง

- เช็คนไข: ถ้าไม่มีสิ่งกีดขวาง (ระยะห่างจากทั้งสามเซ็นเซอร์มากกว่าค่าที่กำหนดในเช็คนไข)
- การกระทำ:
  - เซอร์โวทั้งสองตัวจะถูกตั้งค่ากลับไปตำแหน่งตรงกลาง (90 องศา) โดยใช้ esc3.write(90) และ esc4.write(90)
  - มอเตอร์ BLDC ทั้งสองตัวจะถูกตั้งค่าความเร็วที่กำหนดใน pwmTarget (ความเร็วที่ตั้งไว้ในโค้ดคือ 1060) โดยใช้ esc1.writeMicroseconds(pwmTarget) และ esc2.writeMicroseconds(pwmTarget)
  - พิมพ์ข้อความ "Moving straight" ลงใน Serial Monitor เพื่อแจ้งเตือนว่าหุ่นยนต์กำลังเคลื่อนไปข้างหน้า



# สมาชิกกลุ่ม

6552500017 จิตต์ศุภางค์ จริยพานิชย์

6552500076 พชรกฤษฎ์ พรมมิ





# THANK YOU!

HENRIETTA MITCHELL, FOUNDER & CEO  
AVERY DAVIS, FOUNDER & CTO  
9 MAY, 2023