



R A E

FIRE-SELECT
R E C A R

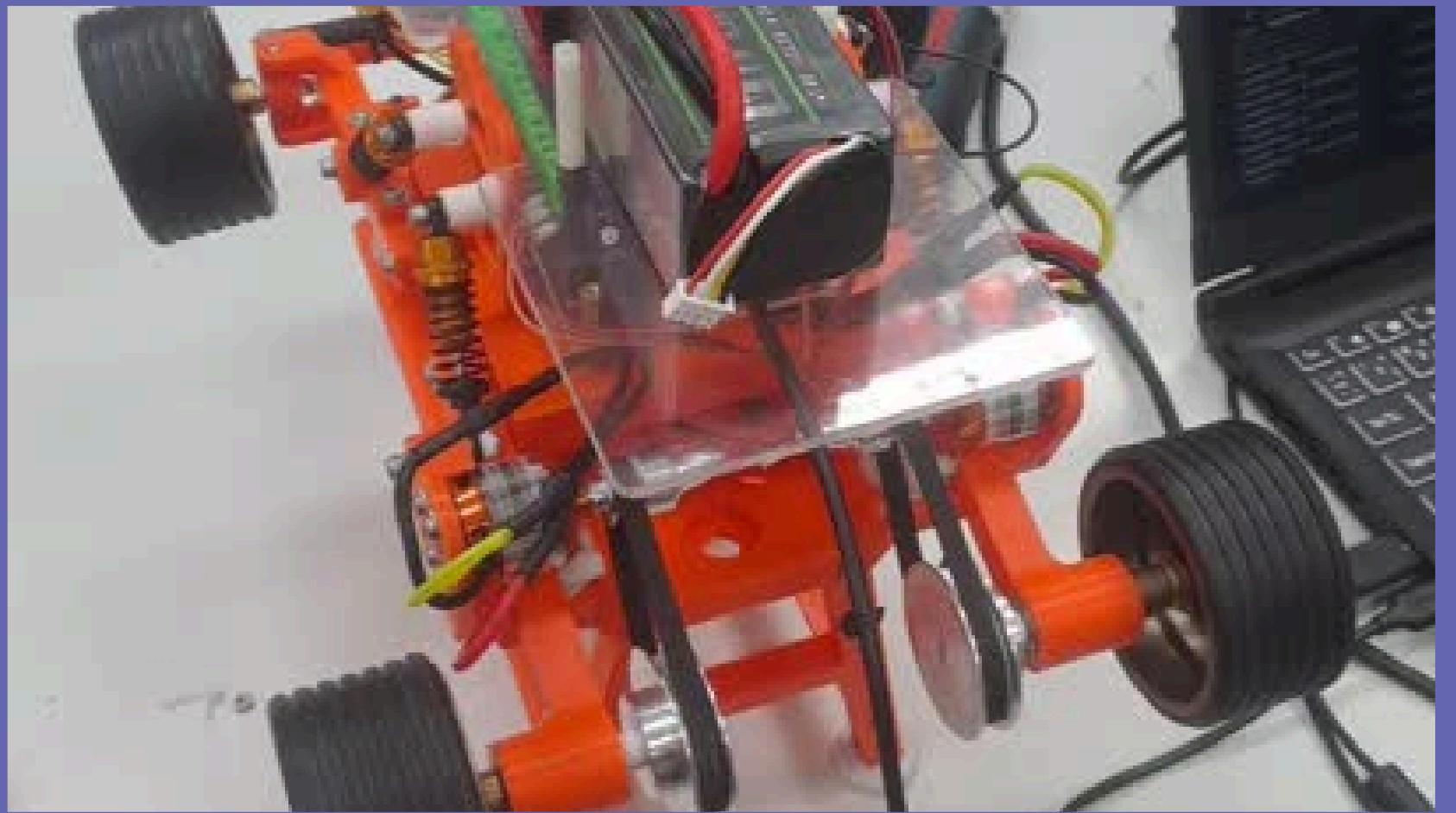
Joshua Jacques Suriya Boinier
6552500033 RAE



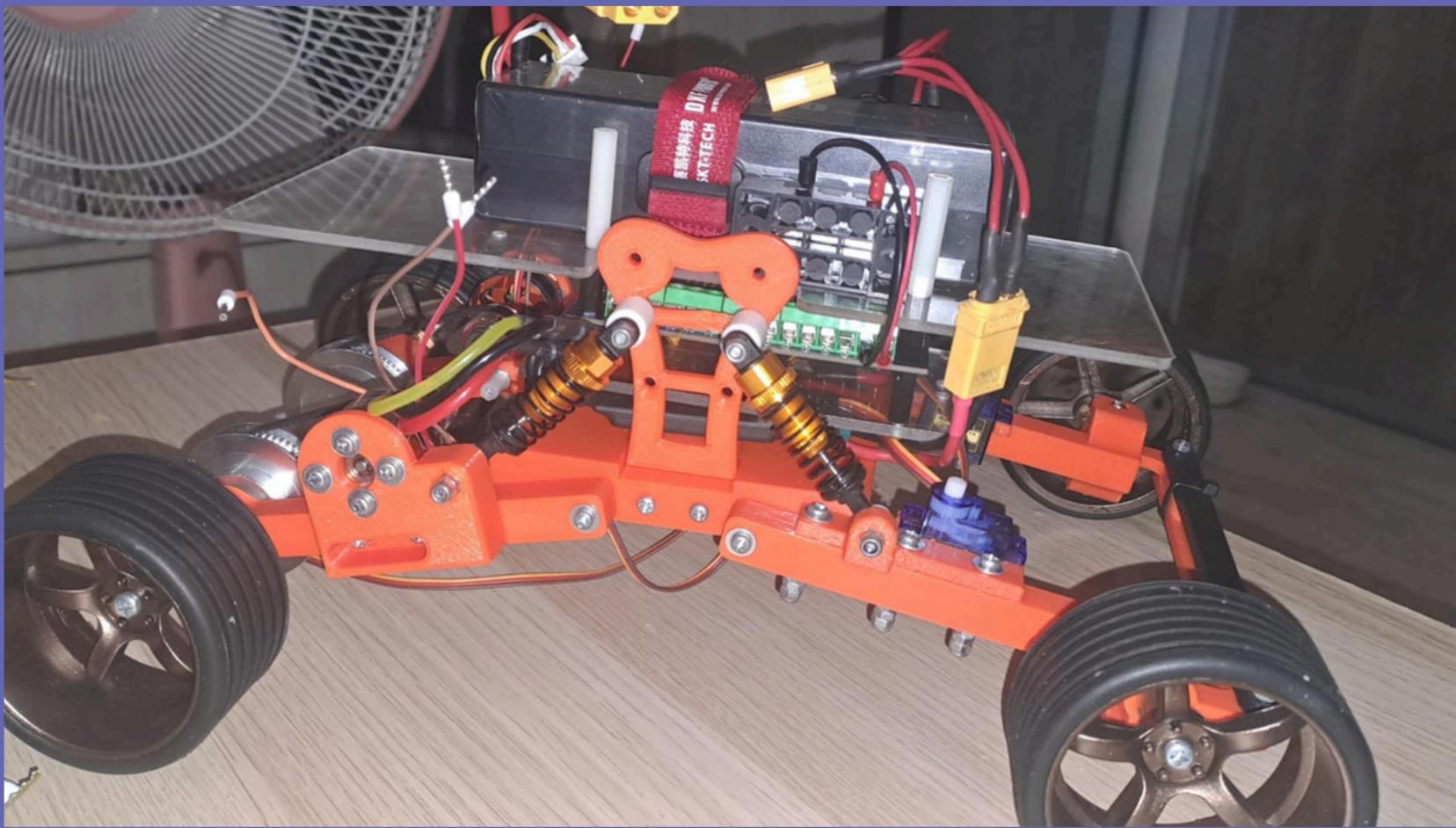
R A E

OVERVIEW

X □ -



X □ -





R A E

```
#include "ESC.h"
```

```
#define LED_PIN (2)
#define SPEED_MIN (1000)
#define SPEED_MAX (1100)
#define MOTORPIN_L 15
#define MOTORPIN_R 13
#define ANALOGPIN_F 34
#define SPEED (1220)
```

```
ESC myESCL(MOTORPIN_L, SPEED_MIN, SPEED_MAX, 500);
```

```
ESC myESCR(MOTORPIN_R, SPEED_MIN, SPEED_MAX, 500);
```

- #include "Esc.h": นำเข้าไลบรารีสำหรับควบคุม ESC (Electronic Speed Controller) ซึ่งจัดการการควบคุมความเร็วของ มอเตอร์ไร้แปรงถ่าน (Brushless Motor)
- กำหนดให้ LED_PIN เก่ากับหมายเลข 2 บน ESP32 เพื่อควบคุม LED
- กำหนดค่า SPEED_MIN เป็น 1000 ซึ่งเป็นค่าความเร็วต่ำสุดที่ ESC ยอมรับ
- กำหนดให้ MOTORPIN_L เก่ากับขา 15 บน ESP32 สำหรับควบคุม มอเตอร์ซ้าย
- กำหนดให้ MOTORPIN_R เก่ากับขา 13 บน ESP32 สำหรับควบคุม มอเตอร์ขวา
- กำหนดให้ ANALOGPIN_F เป็นขา 34 บน ESP32 ใช้สำหรับอ่านค่า จากเซ็นเซอร์ Sharp A0Y21 (ซึ่งเป็นเซ็นเซอร์อินฟราเรด)
- กำหนดค่าความเร็วเริ่มต้น SPEED เก่ากับ 1220 (ค่าความเร็วนี้อาจ ต้องอยู่ในช่วงที่ ESC รองรับ)
- สร้างออบเจกต์ myESCL สำหรับควบคุมมอเตอร์ซ้าย
 1. ขา MOTORPIN_L (15) เป็นขาที่ส่งสัญญาณ PWM ไปยัง ESC ของ มอเตอร์ซ้าย
 2. SPEED_MIN (1000): ความเร็วต่ำสุดที่ ESC ยอมรับ
 3. SPEED_MAX (1100): ความเร็วสูงสุดที่ ESC ยอมรับ
 4. 500: ค่าเริ่มต้น (initial PWM signal)
- สร้างออบเจกต์ myESCR สำหรับควบคุมมอเตอร์ขวา
 1. ขา MOTORPIN_R (13) เป็นขาที่ส่งสัญญาณ PWM ไปยัง ESC ของ มอเตอร์ขวา
 2. SPEED_MIN (1000): ความเร็วต่ำสุดที่ ESC ยอมรับ
 3. (1100): ความเร็วสูงสุดที่ ESC ยอมรับ
 4. 500: ค่าเริ่มต้น (initial PWM signal)

R A E



```
void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);

  // Arm the ESCs
  myESCL.arm();
  delay(10);
  myESCR.arm();

  // Indicate readiness with the LED
  digitalWrite(LED_PIN, HIGH);
  delay(2000);
}
```

- เริ่มต้นการสื่อสารผ่าน Serial:ใช้ Serial.begin(115200) เพื่อเปิดการสื่อสาร UART กับคอมพิวเตอร์ที่ความเร็ว baud rate สำหรับการดีบักหรือแสดงผลใน Serial Monitor
- ตั้งค่าขา LED:
 - 1.กำหนดให้ขา LED_PIN (ขา 2) เป็นขาเอาภพุต (OUTPUT) ด้วย pinMode(LED_PIN, OUTPUT) เพื่อควบคุมสถานะของ LED
- เตรียม ESC ให้พร้อมทำงาน (Arm ESC):
 - 1.เรียกคำสั่งmyESCL.arm()สำหรับ ESC ซ้าย และmyESCR.arm() สำหรับ ESC ขวา
 - 2.เพิ่ม delay(10) เพื่อหน่วงเวลา 10 มิลลิวินาทีระหว่างการ arm ESC แต่ละตัวเพื่อให้มีเวลาตอบสนอง
- แสดงสถานะพร้อมใช้งานด้วย LED:
 - 1.ใช้ digitalWrite(LED_PIN, HIGH) เพื่อเปิด LED และแสดงสถานะว่าอุปกรณ์พร้อมทำงาน
 - 2.รอเวลา 2 วินาที (delay(2000)) เพื่อให้ผู้ใช้งานเห็นว่าอุปกรณ์พร้อมแล้ว ก่อนเริ่มการทำงานหลักของโปรแกรม



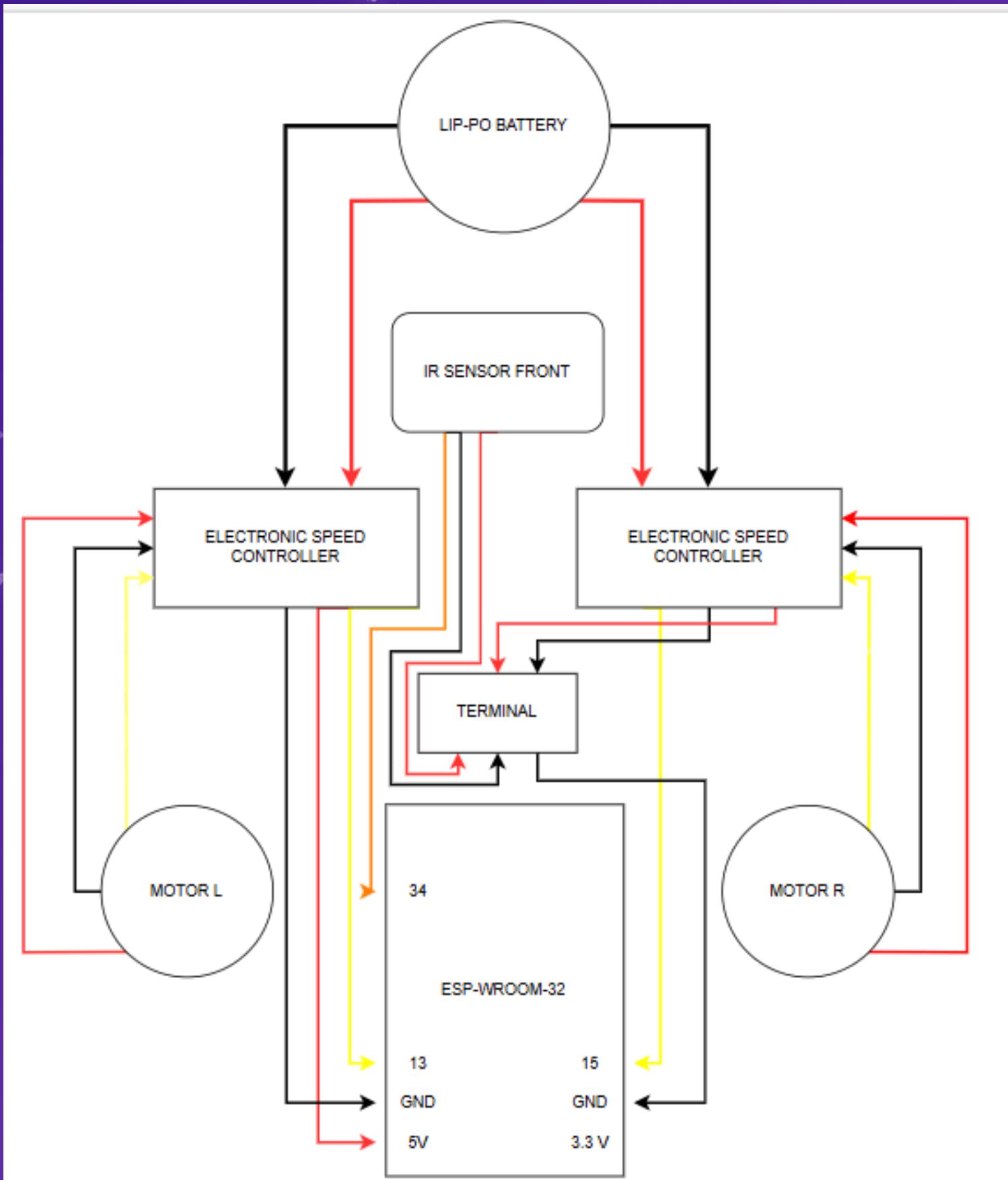
R A E

```
void loop() {  
    int analogValueF = analogRead(ANALOGPIN_F);  
  
    // Print the sensor value for debugging  
    Serial.print("Sensor Value: ");  
    Serial.println(analogValueF);  
  
    // Control motors based on sensor input  
    if (analogValueF < 800) {  
        myESCL.speed(SPEED);  
        myESCR.speed(SPEED);  
        Serial.println("Moving Forward");  
    } else {  
        myESCL.speed(SPEED_MIN);  
        myESCR.speed(SPEED_MIN);  
        Serial.println("STOP");  
    }  
  
    delay(10); // Short delay to stabilize the loop  
}
```

- อ่านค่าจากเซ็นเซอร์: อ่านค่าอ่านมาล็อกจากพิน ANALOGPIN_F และเก็บไว้ในตัวแปร analogValueF
- แสดงค่าที่อ่านได้: พิมพ์ค่าที่อ่านได้ทาง Serial Monitor เพื่อช่วยในการดูบก
- ควบคุมมอเตอร์:
หากค่าจากเซ็นเซอร์ น้อยกว่า 800:
 1. ตั้งค่าความเร็วของมอเตอร์ (myESCL และ myESCR) เป็น SPEED และพิมพ์ข้อความว่า "Moving Forward"
- หากค่ามากกว่าหรือเท่ากับ 800:
 2. ตั้งค่าความเร็วเป็น SPEED_MIN (หยุดมอเตอร์) และพิมพ์ข้อความว่า "STOP"
- หน่วงเวลาสั้น ๆ: เพิ่มการหน่วงเวลา 10 มิลลิวินาทีในแต่ละรอบของลูปเพื่อความเสถียร



R A E





R A E

THURSDAY

15:00

Joshua Jacques Suriya Boinier
6552500033 RAE