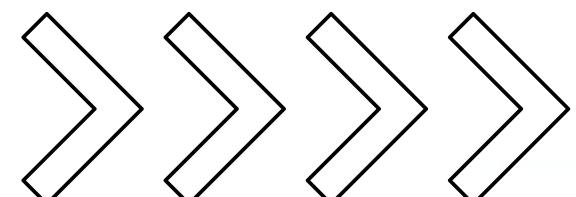


# ROBÔ SUMÔ

## PROGRAMAÇÃO DE MICROCONTROLADORES



### Integrantes:

Jardel Souza

José Arcanjo

João Morais

Monique Salles

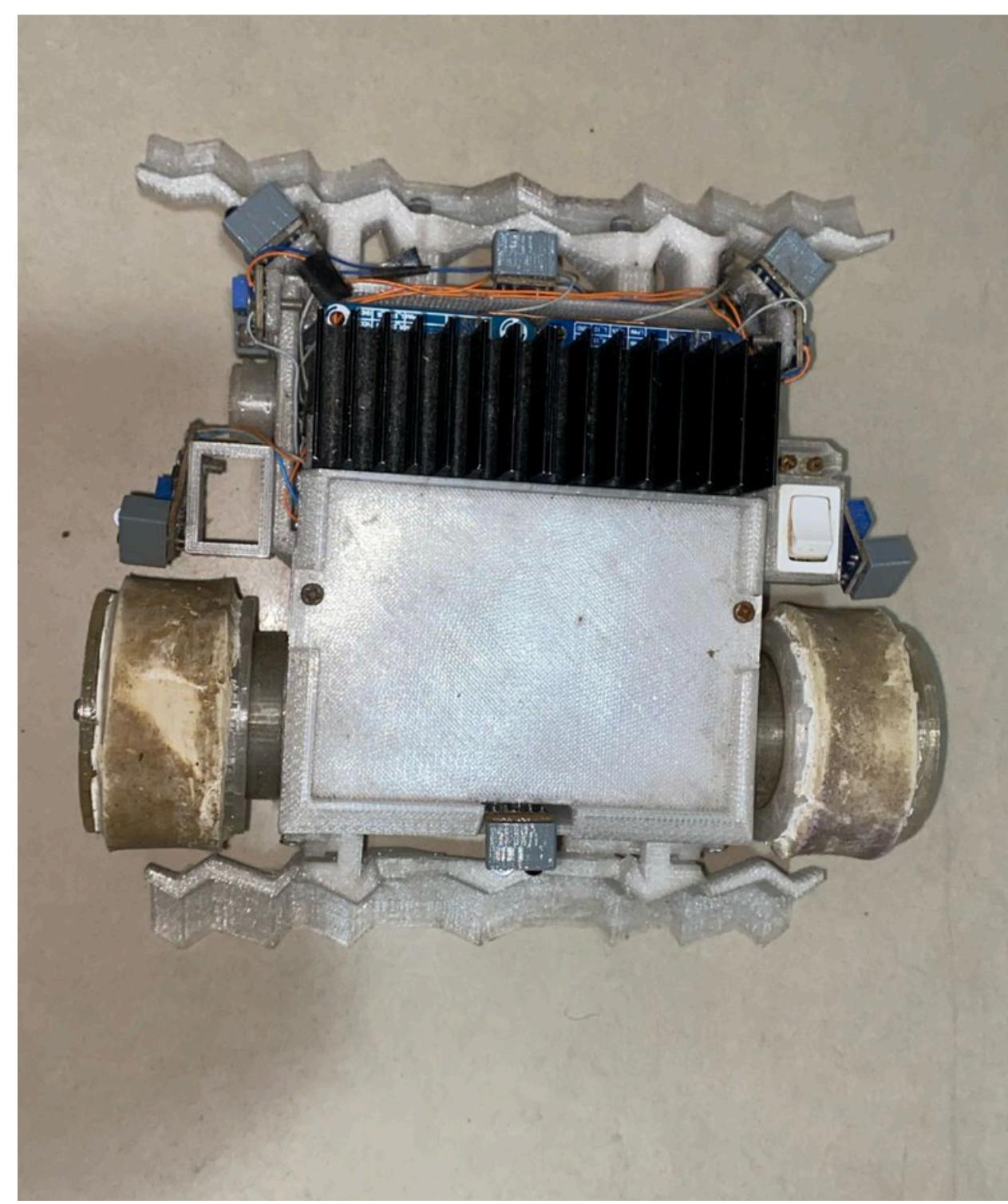
Vinicius Bomfim

# Sumário

- 03** Definição
- 04** Materiais
- 07** Controlador

- 08** Código
- 11** Visão esquemática

# Definição



# Materiais

## 2 motores DC 12V

# 8 sensores ópticos

# Um controlador )Mega2560+Wifi R3

# CH340G USB-TTL ATMega2560+

# ESP8266 32Mb)

# 2 módulos Driver Ponte H 43a Bts7960

Filamento Petg para impressão 3d da estrutura completa quanto carcaça, engrenagens planetárias de tração das rodas.

## 1 interruptor liga/desliga

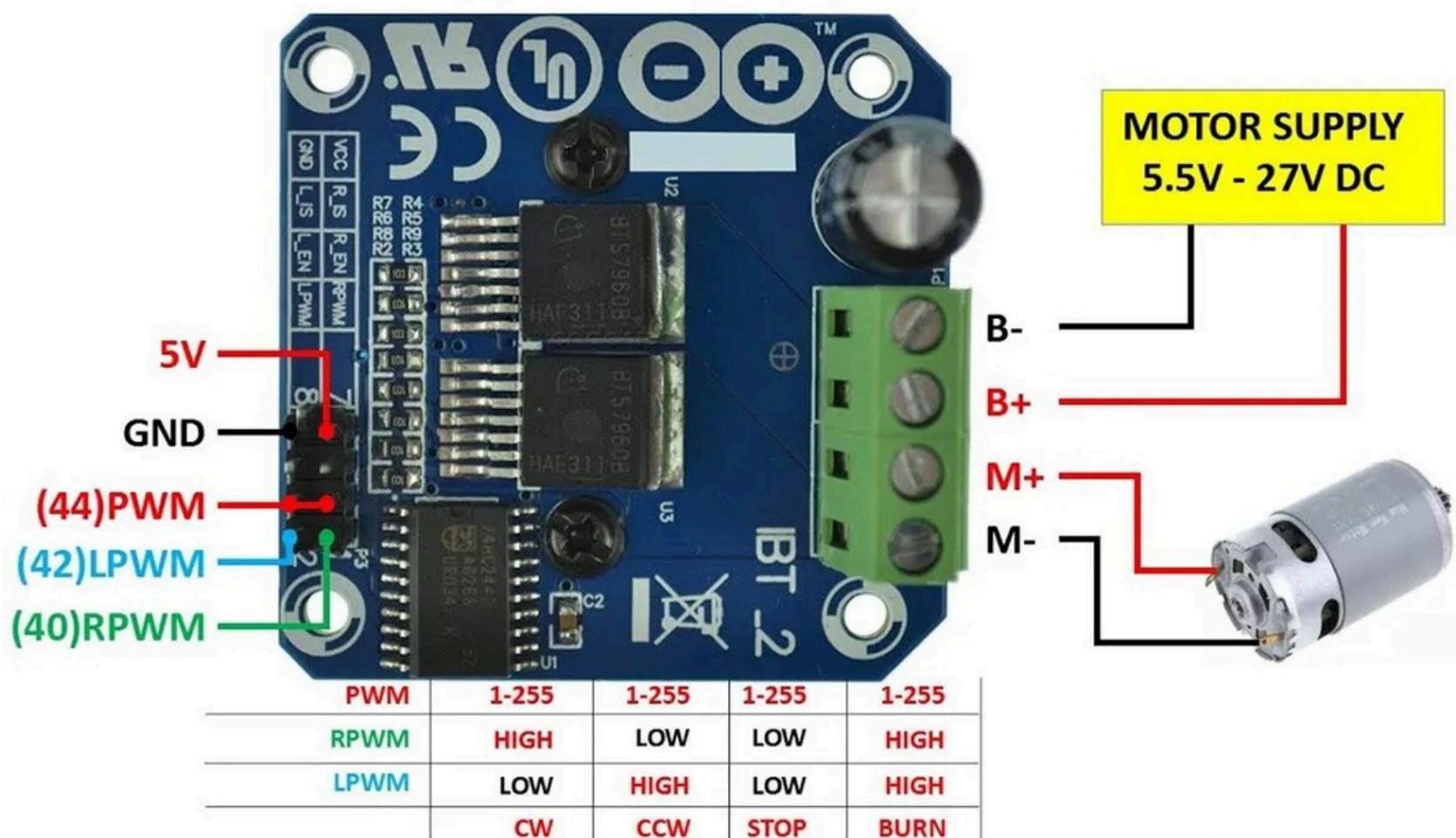
14 Parafusos de fixação dos sensores,  
rodas impressas, módulos Drivers,  
controladores ATMega.

**Fios**

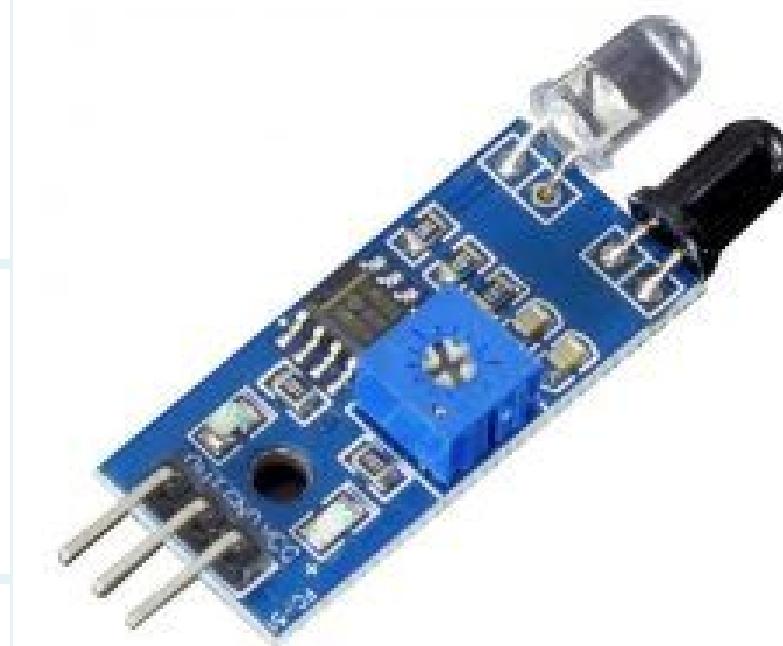
Silicone e amido de milho para fazer as rodas emborrachadas moldadas para atritar melhor ao solo.

# Bateria 14v de drone.

# Materiais



# Materiais



# Controlador



# Código

```
#define Sensor_Frente 52
#define Sensor_Direita_Frente 46
#define Sensor_Esquerda_Frente 47
#define Sensor_Direita 30
#define Sensor_Esquerda 31
#define Sensor_Tras 24
#define Sensor_Seguidor_Direita 40
#define Sensor_Seguidor_Esquerda 41
#define Motor_Direita_Frente_PWM 6
#define Motor_Direita_Tras_PWM 5
#define Motor_Esquerda_Frente_PWM 8
#define Motor_Esquerda_Tras_PWM 9
int Motores_Ligados(10);
```

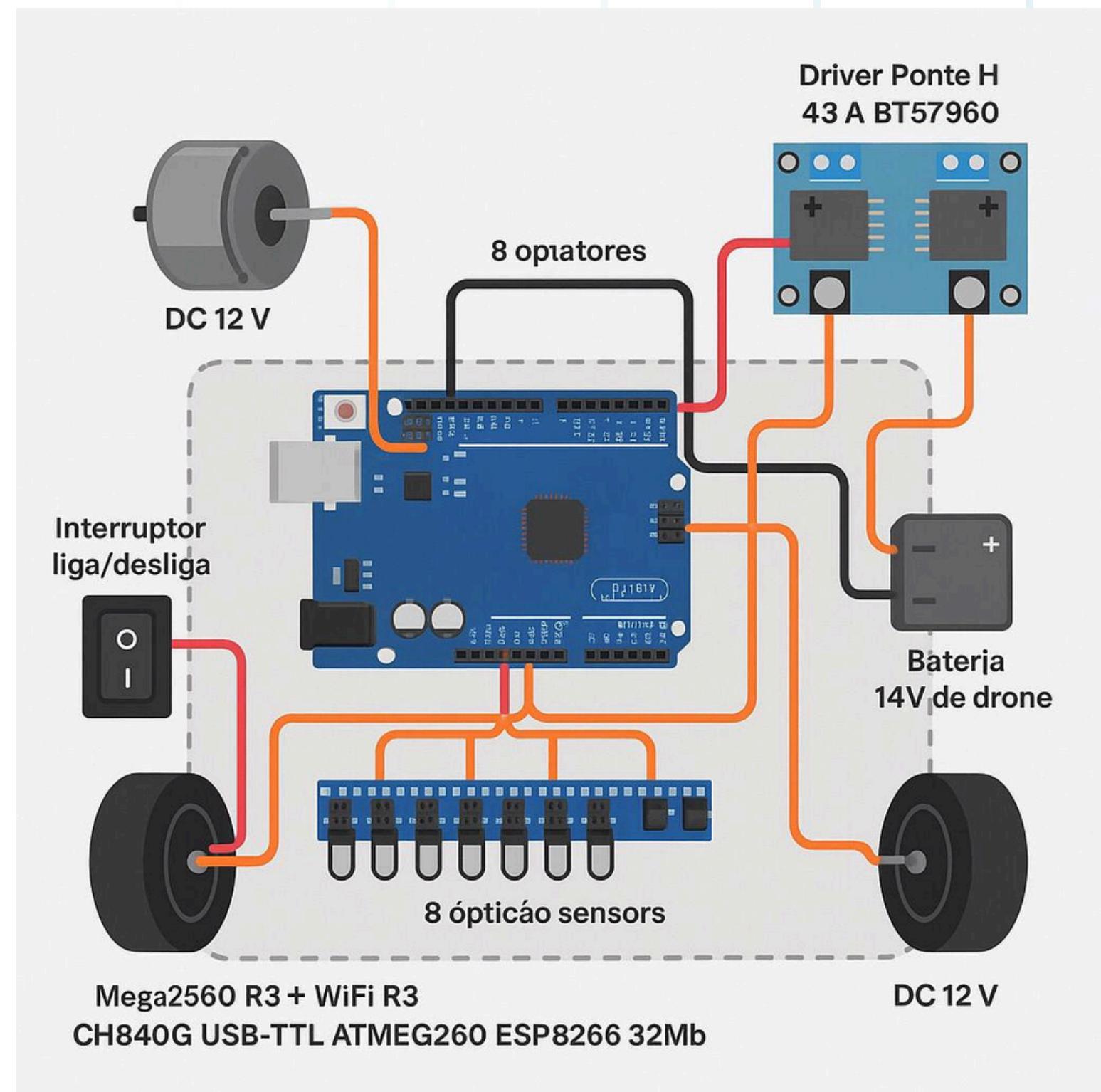
# Código

```
void setup() {
    //sensores de entrada
    pinMode(Sensor_Frente, INPUT);
    pinMode(Sensor_Direita_Frente, INPUT);
    pinMode(Sensor_Esquerda_Frente, INPUT);
    pinMode(Sensor_Direita, INPUT);
    pinMode(Sensor_Esquerda, INPUT);
    pinMode(Sensor_Tras, INPUT);
    pinMode(Sensor_Seguidor_Direita, INPUT);
    pinMode(Sensor_Seguidor_Esquerda, INPUT)
    //motores de saida
    pinMode(Motor_Direita_Frente_PWM, OUTPUT);
    pinMode(Motor_Direita_Tras_PWM, OUTPUT);
    pinMode(Motor_Esquerda_Frente_PWM, OUTPUT);
    pinMode(Motor_Esquerda_Tras_PWM, OUTPUT);
    //Set motores On
    pinMode(Motores_Ligados, OUTPUT);
    //Set motores On
    pinMode(Motores_Ligados, OUTPUT);
    //tempo antes de iniciar o combate
    delay(5000);
    digitalWrite(Motores_Ligados, true);
}
```

# Código

```
void loop() {
    //////////////Lendo valores dos sensores///////////
    bool estado_sensor_frente = digitalRead(Sensor_Frente);
    bool estado_sensor_direita_frente = digitalRead(Sensor_Direita_Frente);
    bool estado_sensor_esquerda_frente = digitalRead(Sensor_Esquerda_Frente);
    bool estado_sensor_direita = digitalRead(Sensor_Frente);
    bool estado_sensor_esquerda = digitalRead(Sensor_Esquerda);
    bool estado_sensor_tras = digitalRead(Sensor_Tras);
    bool estado_sensor_seguidor_direita = digitalRead(Sensor_Seguidor_Direita);
    bool estado_sensor_seguidor_esquerda =
        digitalRead(Sensor_Seguidor_Esquerda);
    //////////////Se sensores seguidores de linha estiverem OK, ande para frente/////////
    if(!estado_sensor_seguidor_direita & !estado_sensor_seguidor_esquerda){
        //////////////ande para frente///////////////////////////////      analogWrite(Motor_Esquerda_
    }
}
```

# Visão esquemática



# Visão esquemática

