

**UNIVERSIDADE TIRADENTES**

Ciência da Computação

Paulo Ricardo Gomes Gois Silva

Vitor Feitosa Almeida

**Carro Controlado por Bluetooth com Modo Segue Linha**

Código do Projeto

Aracaju - Sergipe  
2019

PAULO RICARDO GOMES GOIS SILVA  
VITOR FEITOSA ALMEIDA

CÓDIGO DO PROJETO  
Carro Controlado por Bluetooth com Modo Segue Linha

Código do projeto referente a  
segunda unidade da matéria  
Fundamentos e Arquitetura de  
Computadores.

Aracaju – Sergipe  
2019

```
#include <NewPing.h>
```

```
// Define o Pino 2 como o pino de comunicação do Bluetooth.
```

```
#define BTState 2
```

```
// Define os pinos de utilização do motor.
```

```
#define IN1 13
```

```
#define IN2 12
```

```
#define IN3 11
```

```
#define IN4 10
```

```
// Define os pinos de utilização dos sensores ultrassônicos frontal e traseiro
```

```
#define echo_f 2
```

```
#define trigger_f 3
```

```
#define echo_t 8
```

```
#define trigger_t 9
```

```
#define max_distancia 100
```

```
// Cria os sonares
```

```
NewPing sonar_f (trigger_f, echo_f, max_distancia);
```

```
NewPing sonar_t (trigger_t, echo_t, max_distancia);
```

```
// Variáveis
```

```
int i = 0;
```

```
int j = 0;
```

```
int state_rec;
```

```
int vSpeed = 255;
```

```
char state;
```

```
bool manual = true;
```

```
float distancia_f;
```

```
float distancia_t;

void setup() {
    // Inicializa a comunicação serial em 9600 bits.
    Serial.begin(9600);

    // Inicializa as portas como entrada e saída.
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(BTState, INPUT);
}

void loop() {
    distancia_f = sonar_f.ping_cm();
    distancia_t = sonar_t.ping_cm();

    // Para o carro quando a conexão com Bluetooth é perdida ou desconectada.
    if (digitalRead(BTState) == LOW) {
        state_rec = 'S';
    }

    // Salva os valores da variável 'state'
    if (Serial.available() > 0) {
        state_rec = Serial.read();
        state = state_rec;
    }

    // Altera a velocidade de acordo com valores especificados.
    if (state == '0')    { vSpeed = 0; }
```

```

else if (state == '4') { vSpeed = 100; }
else if (state == '6') { vSpeed = 155; }
else if (state == '7') { vSpeed = 180; }
else if (state == '8') { vSpeed = 200; }
else if (state == '9') { vSpeed = 230; }
else if (state == 'q') { vSpeed = 255; }

// Imprime no monitor serial o state atual caso seja diferente de 'S'.
if (state != 'S') { Serial.print(state); }

// Verifica se o carro está em modo manual ou seguidor de linha.
if (state == 'x')    { manual = true; }
else if (state == 'X') { manual = false; }

if (manual == true) {
    // Se o estado recebido for igual a 'F', o carro se movimenta para frente.
    if (state == 'F') {
        if (distancia_f < 10 && distancia_f > 0) {
            analogWrite(IN1, 0);
            analogWrite(IN2, 0);
            analogWrite(IN3, 0);
            analogWrite(IN4, 0);
        }
        else {
            analogWrite(IN1, 0);
            analogWrite(IN2, vSpeed);
            analogWrite(IN3, 0);
            analogWrite(IN4, vSpeed);
        }
    }
}

```

else if (state == 'I') { // Se o estado recebido for igual a 'I', o carro se movimenta para Frente Direita.

```
    analogWrite(IN1, 0);  
    analogWrite(IN2, 100);  
    analogWrite(IN3, 0);  
    analogWrite(IN4, vSpeed);  
}
```

else if (state == 'G') { // Se o estado recebido for igual a 'G', o carro se movimenta para Frente Esquerda.

```
    analogWrite(IN1, 0);  
    analogWrite(IN2, vSpeed);  
    analogWrite(IN3, 0);  
    analogWrite(IN4, 100);  
}
```

else if (state == 'B') { // Se o estado recebido for igual a 'B', o carro se movimenta para trás.

```
    if (distancia_t < 10 && distancia_t > 0) {  
        analogWrite(IN1, 0);  
        analogWrite(IN2, 0);  
        analogWrite(IN3, 0);  
        analogWrite(IN4, 0);  
    }  
    else{  
        analogWrite(IN1, vSpeed);  
        analogWrite(IN2, 0);  
        analogWrite(IN3, vSpeed);  
        analogWrite(IN4, 0);  
    }  
}
```

else if (state == 'H') { // Se o estado recebido for igual a 'H', o carro se movimenta para Trás Esquerda.

```
    analogWrite(IN1, vSpeed);  
    analogWrite(IN2, 0);  
    analogWrite(IN3, 100);  
    analogWrite(IN4, 0);  
}
```

else if (state == 'J') { // Se o estado recebido for igual a 'J', o carro se movimenta para Trás Direita.

```
    analogWrite(IN1, 100);  
    analogWrite(IN2, 0);  
    analogWrite(IN3, vSpeed);  
    analogWrite(IN4, 0);  
}
```

else if (state == 'L') { // Se o estado recebido for igual a 'L', o carro se movimenta para esquerda.

```
    analogWrite(IN1, 0);  
    analogWrite(IN2, vSpeed);  
    analogWrite(IN3, vSpeed);  
    analogWrite(IN4, 0);  
}
```

else if (state == 'R') { // Se o estado recebido for igual a 'R', o carro se movimenta para direita.

```
    analogWrite(IN1, vSpeed);  
    analogWrite(IN2, 0);  
    analogWrite(IN3, 0);  
    analogWrite(IN4, vSpeed);  
}
```

```
else if (state == 'S') { // Se o estado recebido for igual a 'S', o carro permanece parado.
    analogWrite(IN1, 0);
    analogWrite(IN2, 0);
    analogWrite(IN3, 0);
    analogWrite(IN4, 0);
}
}

else {

}

}
```