

# Descrição Detalhada do Código do Robô Seguidor de Linha

Equipa 9

29 de Janeiro de 2025

## 1 Introdução

Este documento fornece uma descrição detalhada do código de um robô seguidor de linha, que utiliza um microcontrolador Arduino para controlar motores e sensores. O robô é projetado para seguir uma linha, contar voltas e entrar em boxes após completar um número específico de voltas. O código é estruturado de forma modular, facilitando a compreensão e a manutenção.

## 2 Estrutura do Código

O código é dividido em várias secções, cada uma com uma função específica. As principais secções incluem a definição de pinos, parâmetros PID, estados do robô, variáveis de controlo e as funções principais que controlam o comportamento do robô.

### 2.1 Definição de Pinos

Os pinos do Arduino são definidos no início do código. Cada pino é atribuído a uma função específica, como controlo de motores, LEDs e leitura de sensores. Abaixo estão os pinos definidos:

- M1 e M2: Pinos para controlo dos motores.
- dir1 e dir2: Pinos para definir a direção dos motores.
- RED\_pin, GREEN\_pin, BLUE\_pin: Pinos para controlo de LEDs RGB.
- pin\_S1 a pin\_S5: Pinos para leitura dos sensores de linha.
- ldrPin: Pino para o sensor LDR que detecta luz para iniciar a corrida.

## 2.2 Parâmetros PID

Os parâmetros do controlador PID (Proporcional, Integral, Derivativo) são definidos para ajustar a velocidade dos motores com base nas leituras dos sensores. Os parâmetros são:

- $K_p = 9$ : Ganho proporcional.
- $K_i = 0$ : Ganho integral.
- $K_d = 11$ : Ganho derivativo.

## 2.3 Estados do Robô

O robô pode estar em um dos quatro estados, definidos por um enum:

- **ESPERA**: O robô aguarda a partida.
- **CORRIDA**: O robô segue a linha.
- **BOXES**: O robô entra em boxes após completar a volta.
- **FINALIZADO**: O robô para após completar o número de voltas.

## 2.4 Variáveis de Controle

Várias variáveis são utilizadas para controlar o comportamento do robô:

- `foiBoxes`: Indica se o robô já entrou nos boxes.
- `passouMeta`: Indica se o robô já passou pela linha de chegada.
- `contadorBifurcacoes`: Contador de bifurcações após a meta.
- `tempoInicialBoxes`: Tempo desde que o robô passou a meta na volta 2.
- `TEMPO_PARA_BOXES`: Tempo mínimo após passar a meta para entrar nos boxes.
- `voltas`: Contador de voltas completadas.
- `debounceVolta`: Tempo de debounce para contagem de voltas.
- `ultimaVolta`: Armazena o tempo da última volta registrada.

## 2.5 Setup

A função `setup()` é chamada uma vez no início do programa. Nela, os pinos são configurados como entrada ou saída, e o sistema é inicializado. O LED vermelho é aceso para indicar que o robô está em espera.

## 2.6 Loop Principal

A função `loop()` é chamada repetidamente e contém a lógica principal do robô. O estado atual do robô é verificado e as ações apropriadas são executadas com base nesse estado. O robô inicia a corrida quando a luz é detectada pelo sensor LDR.

### 2.6.1 Estado ESPERA

No estado **ESPERA**, o robô aguarda a luz. Quando a luz é detectada (valor do LDR acima do limiar), o estado muda para **CORRIDA** e o LED verde é aceso.

### 2.6.2 Estado CORRIDA

No estado **CORRIDA**, o robô chama a função `seguirLinha()` para seguir a linha. Ele verifica se passou pela linha de chegada e se atende às condições para entrar nos boxes. Se o robô completar 4 voltas, o estado muda para **FINALIZADO**.

### 2.6.3 Estado BOXES

No estado **BOXES**, a função `executarBoxes()` é chamada, onde o robô realiza uma série de movimentos e paradas. Após completar as ações nos boxes, o robô retorna ao estado **CORRIDA**.

### 2.6.4 Estado FINALIZADO

No estado **FINALIZADO**, os motores são parados e o LED azul é aceso, indicando que a corrida foi concluída.

## 3 Funções Principais

### 3.1 seguirLinha()

A função `seguirLinha()` é responsável por controlar os motores do robô para que ele siga a linha. Ela lê os valores dos sensores e ajusta a velocidade dos

motores com base nas leituras, utilizando um controlador PID para garantir que o robô mantenha a trajetória correta.

### **3.1.1 Leitura dos Sensores**

Os sensores são lidos e a média das leituras é calculada para suavizar os dados. Se todos os sensores detectarem a linha (valores abaixo de 400), o robô conta a volta e reinicia o contador de bifurcações.

### **3.1.2 Detecção de Bifurcações**

A função também verifica se o robô detectou bifurcações após passar pela linha de chegada. Se as condições forem atendidas, o contador de bifurcações é incrementado.

### **3.1.3 Recuperação da Linha**

Se todos os sensores detectarem a cor clara, o robô tenta recuperar a linha utilizando a função `recuperarLinha()`.

## **3.2 contarVolta()**

A função `contarVolta()` é chamada quando o robô detecta que cruzou a linha de chegada. Ela registra a volta e altera a cor do LED de acordo com o número de voltas completadas. O tempo de debounce é utilizado para evitar contagens múltiplas.

## **3.3 executarBoxes()**

A função `executarBoxes()` é chamada quando o robô entra na fase de boxes. Ela realiza um pequeno movimento inicial, segue a linha por 2 segundos, para por 3 segundos e, em seguida, sai dos boxes.

## **3.4 setColor()**

A função `setColor(int red, int green, int blue)` é utilizada para controlar a cor do LED RGB, permitindo que o robô indique seu estado atual através de diferentes cores.

## 4 Diagrama de Estados

O diagrama de estados abaixo ilustra as transições entre os diferentes estados do robô durante a execução do código.

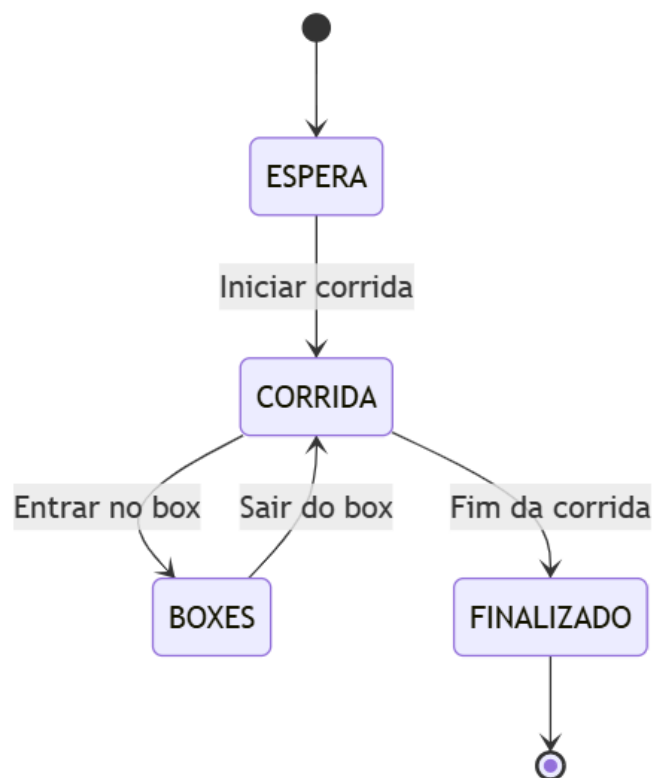


Figure 1: Diagrama de estados do robô seguidor de linha.

## 5 Conclusão

O código apresentado implementa um robô seguidor de linha com funcionalidades de contagem de voltas e entrada em boxes. A estrutura modular do código facilita a compreensão e a manutenção, permitindo futuras melhorias e adaptações. Este documento fornece uma visão abrangente do funcionamento do robô, detalhando cada parte do código e suas respectivas funções.