

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC SYLVIO DE MATTOS CARVALHO

Curso Técnico em Mecatrônica M-TEC

**Luana Beatriz Souza Carlos
Poliana Victória Barbosa dos Santos
Thais Vitória de Oliveira
Thiago Luiz de Oliveira**

**Robótica Competitiva: Desenvolvimento e Testes de um Protótipo
de Robô Sumô**

Matão, SP

2024

SUMÁRIO

1 INTRODUÇÃO	3
2 OBJETIVOS (GERAL E ESPECÍFICOS)	4
2.1 Objetivo Geral	4
2.2 Objetivos Específicos	4
3 DESCRIÇÃO DO PROJETO	5
3.1 Apresentação Geral do Projeto	5
3.2 Materiais Utilizados e suas Especificações Técnicas	6
4.1 Atividades e Metas 1º Semestre de 2024	14
4.2 Atividades e Metas 2º Semestre de 2024	15
5 DESENVOLVIMENTO DO PROJETO	16
5.1 Descrição do Funcionamento	16
5.2 Montagem Física do Projeto	17
5.3 Desenvolvimento do programa do robô	22
6 CONSIDERAÇÕES FINAIS	24

1 INTRODUÇÃO

Este documento apresenta o Projeto Técnico Robótica Competitiva: Desenvolvimento e Testes de um Protótipo de Robô Sumô referente ao projeto de Trabalho de Conclusão de Curso elaborado pela equipe.

Esse projeto foi escolhido por proporcionar aos estudantes uma oportunidade de desenvolver habilidades e competências relativas ao curso de Mecatrônica, contribuindo para sua formação e preparando-os para novas jornadas.

Um robô autônomo é uma máquina que executa funções sem intervenção externa, utilizando sistemas microprocessados, sensores e lógica computacional para tomar decisões e agir no ambiente em que está inserido. Esses sistemas autônomos podem variar desde sensores até microprocessadores e podem ser alimentados por diferentes fontes de energia.

No contexto de um projeto de robô para competições de sumô, existem regras específicas a serem seguidas. O principal objetivo da competição é empurrar o adversário para fora de uma área delimitada. As regras determinam as dimensões do robô, como largura, comprimento, limite de massa e altura livre, e proíbem características destrutivas. O combate tem uma duração definida, conhecida como round, que geralmente é de 90 segundos. Como o robô não é controlado externamente, ele deve ser capaz de internamente realizar a contagem do tempo e automaticamente parar quando o tempo acabar. Além disso, as normas internacionais exigem que a borda da arena de combate seja branca e a parte interna seja preta.

2 OBJETIVOS (GERAL E ESPECÍFICOS)

2.1 Objetivo Geral

O objetivo geral do projeto é:

- Construir um robô sumô.

2.2 Objetivos Específicos

Os objetivos específicos do projeto são:

- Definir os componentes ideais para o robô;
- Elaborar o projeto estrutural do robô;
- Montar o robô, efetuando as conexões dos seus componentes;
- Desenvolver a programação do robô.

3 DESCRIÇÃO DO PROJETO

3.1 Apresentação Geral do Projeto

O robô tem o formato de um veículo, contendo duas rodas de borracha em uma das extremidades do chassi acionadas cada uma através de um motor CC, e uma roda fixa no centro da outra extremidade.

Sobre o chassi foi instalado uma placa de Arduino modelo Nano, o qual possui o microcontrolador Atmega 328P programado para que o robô se movimente através de comandos via celular, que está conectado ao Arduino através de um módulo bluetooth.

O robô é alimentado por uma bateria recarregável de 12V, instalado também no chassi.

A Figura 1 apresenta uma foto do nosso projeto final montado.



Figura 1 – Foto do projeto montado
FONTE: Elaborado pelos autores (2024).

3.2 Materiais Utilizados e suas Especificações Técnicas

Os materiais utilizados em nosso projeto estão apresentados na Tabela 1.

Tabela 1 – Lista de Materiais

Material	Quantidade
Arduino nano Atmega 328P	1
Ponte H LM2984C	1
Bluetooth Hc-05	1
Motor de corrente contínua	2
Bateria 12v 4400mah	1
Pneu de borracha 68mm	2
Roda esfera	1
Chapa Alumínio	1

FONTE: Elaborado pelos autores (2024).

A Ponte H é um circuito usado em motores de corrente contínua para controlar a direção do movimento. Composta por quatro chaves controladas eletronicamente, permite que o motor gire tanto no sentido horário quanto no sentido anti-horário, usando uma quantidade mínima de energia do circuito de controle.

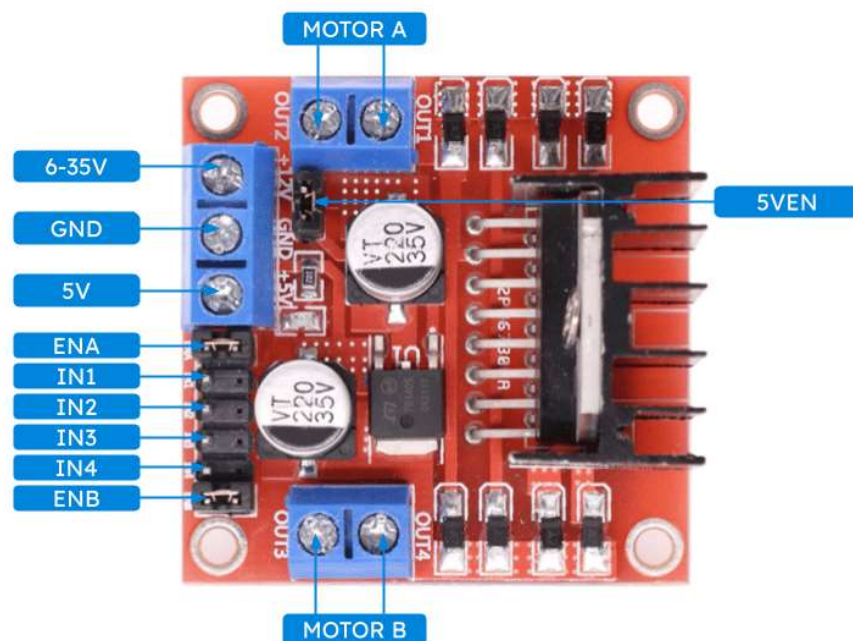


Figura 2: Ponte H Dupla L298N

FONTE: MakerHero

O Arduino Nano é uma placa microcontroladora compacta com 14 pinos digitais que operam a 5V, cada um capaz de fornecer até 40mA de corrente. Ele é amplamente utilizado em projetos de eletrônica e robótica, sendo programável via computador usando a IDE do Arduino. Suas aplicações incluem controle de dispositivos como motores e sensores, além de permitir comunicação via Bluetooth para envio e recebimento de dados em aplicações robóticas e outros projetos eletrônicos.

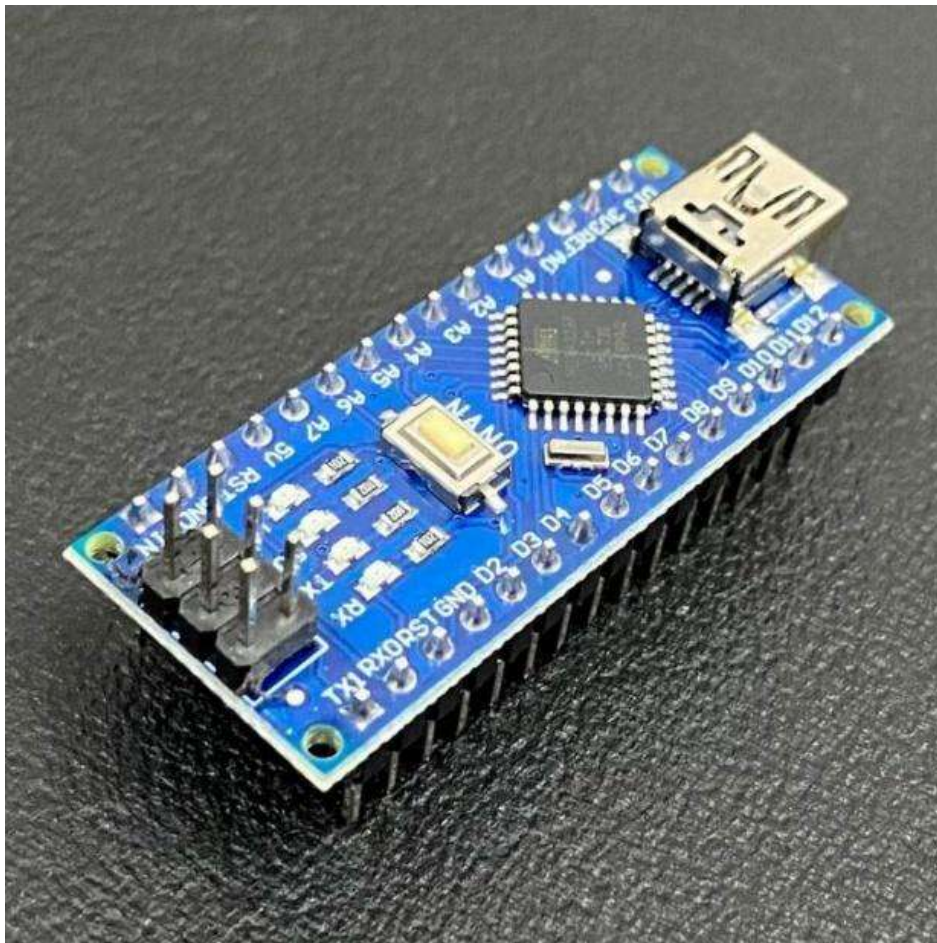


Figura 3 Arduino Nano ATmega328
FONTE: Autoria Própria (2024)

O módulo Bluetooth HC-05 é um dispositivo que utiliza tecnologia Bluetooth para comunicação sem fio entre microcontroladores, como o Arduino, e outros dispositivos. É amplamente usado em projetos de eletrônica e robótica para enviar comandos, como direções de movimento e ajustes de velocidade, facilitando a interação e controle remoto em diversas aplicações.

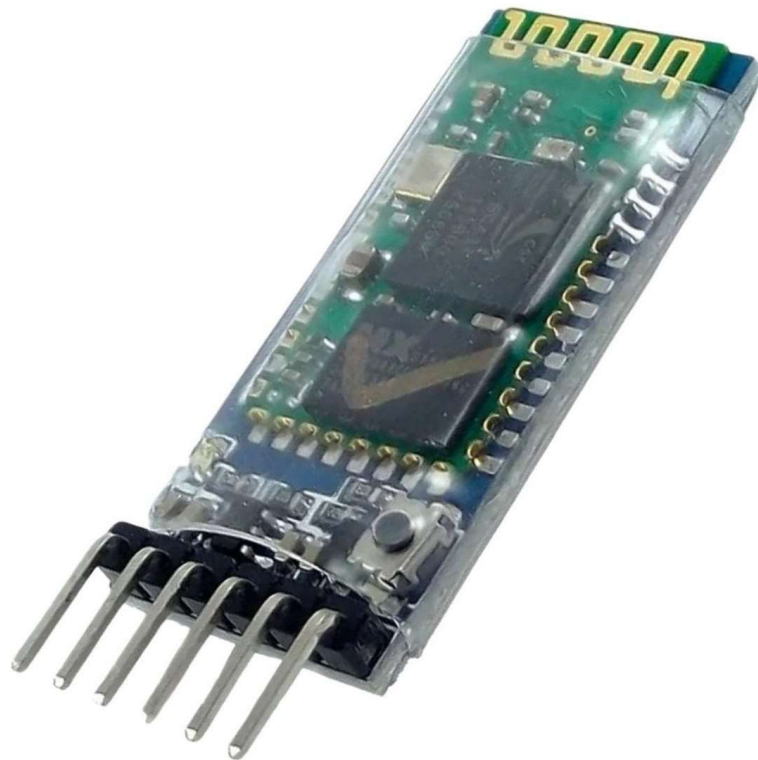


Figura 4 - Bluetooth HC-05

Disponível em <<https://www.smartcomponentes.com/produto/modulo-bluetooth-master-hc05.html>> Acessado em 23 maio 2024

O motor de corrente contínua (CC) converte energia elétrica em movimento mecânico através da interação de campos magnéticos. Em robótica, é usado para fornecer movimento e locomoção aos robôs, sendo controlado via Bluetooth para receber comandos sem fio que direcionam seu movimento em diferentes direções.



Figura 5 – Motor de corrente contínua
FONTE: alldatasheet Acessado em 23 maio 2024

Uma bateria de 12 volts converte energia química em elétrica através de reações eletroquímicas. Ela alimenta motores DC e dispositivos como o Arduino Nano, fornecendo energia elétrica para seu funcionamento autônomo.



Figura 6 – Bateria 12v 44000mah
FONTE: Alldatasheet Acessado em 23 mai 2024

Um pneu de borracha com diâmetro de 68mm é indicado para projetos de robótica, especialmente na construção de carrinhos autônomos. Feito de borracha, é compatível com o Arduino em projetos de controle remoto. Pode ser acoplado diretamente ao Motor DC 3V-6V "Amarelinho" com Caixa de Redução, facilitando sua integração em sistemas robóticos.



Figura 7 – Pneu de borracha 68mm
FONTE: Saravati Acessado em 23 maio 2024

A Roda esfera é crucial em projetos de robótica, especialmente em carrinhos, oferecendo movimento independente do carrinho, independentemente da direção. A peça oferece maior estabilidade ao robô, proporcionando apoio adicional para equilibrá-lo. Facilita movimentos suaves e mudanças de direção, permitindo locomoção em qualquer direção sem ajustes nas outras rodas. É feita de aço, com furos de fixação de 4 mm de diâmetro e distância de 38 mm entre eles. Suas dimensões são 44,8 x 28 x 17 mm e pesa 40 g.



Figura 8 – Roda esfera
FONTE: Larstore Acesso em: 18 nov. 2024.

A chapa de alumínio é um produto feito de alumínio com dimensões de 100 cm x 50 cm. Possui uma superfície plana e lisa, sem relevo, sendo um material leve, resistente à corrosão, com boa condutividade térmica e elétrica.

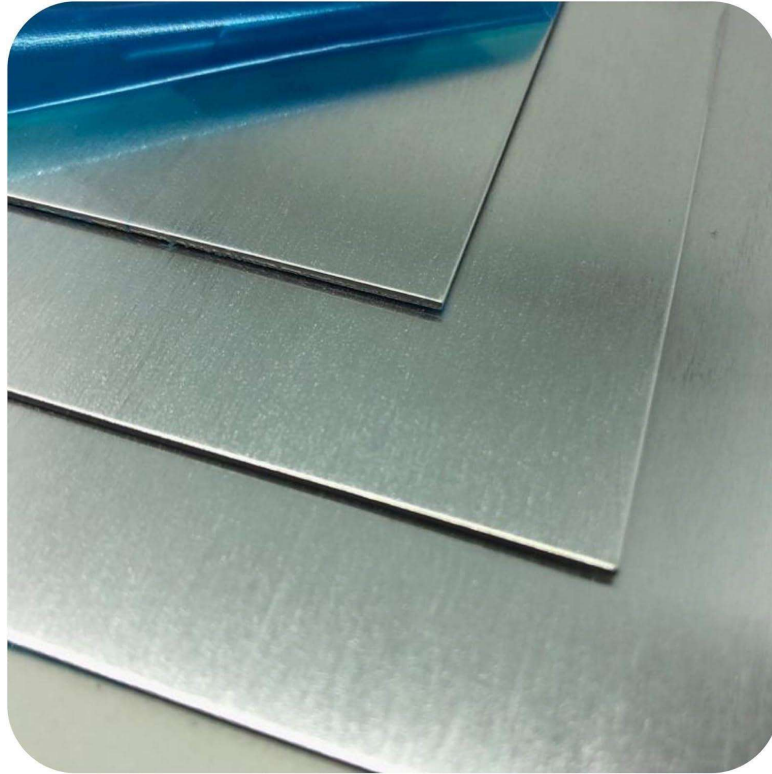


Figura 9 – Chapa de alumínio
FONTE: Shockmetais Acessado 23 maio 2024

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2024

Nossas atividades para o primeiro semestre começaram em fevereiro com a montagem da equipe e o início das pesquisas sobre o robô de sumô, também demos início a elaboração do nosso diário de bordo, que se estendeu por todo o semestre. Ao decorrer de março e abril definimos alguns objetivos do robô e realizamos o preenchimento dos capítulos do boneco, discutimos sobre o robô, em maio e fizemos uma prévia sobre o projeto, solicitada pela escola e logo após em junho fizemos as correções necessárias de acordo com o que foi dito a prévia.

Tabela 2: Cronograma de atividades e metas no 1º Semestre de 2024

Atividade	Fev.		Março		Abril		Maio		Junho	
	1ª Qui nze na	2ª Qui nze na	1ª Qui nze na	2ª Qui nze na	1ª Qui nze na	2ª Qui nze na	1ª Q u i n z e n a	2ª Q u i n z e n a	1ª Qui nze na	2ª Qui nze na
montagem da equipe	x	x								
definição do nome do robô			x							
definição dos objetivos do projeto				x						
redação do capítulo 3					x	x				
preparação para a prévia							x	x		
correções após prévia									x	x
diário de bordo	x	x	x	x	x	x	x	x	x	x

FONTE: Elaborado pelos autores (2024).

4.2 Atividades e Metas 2º Semestre de 2024

Nossas atividades para o segundo semestre começaram em julho com o preenchimento do boneco e analisamos chassis para comprar, em setembro fizemos a compra do chassi, realizamos a montagem e programação do robô a partir da segunda quinzena de setembro, também realizamos a descrição do funcionamento e o fluxograma a partir de outubro, elaboração do nosso diário de bordo, que se estendeu por todo o semestre e na última quinzena apresentamos o tcc e foi realizado a luta de robô sumo.

Tabela 3: Cronograma de atividades e metas no 2º Semestre de 2024

Atividade	Julho		Agosto		Setembro		Outubro		Novembro	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
preenchimento do boneco	X	X								
análise de chassis para compra			X	X						
compra do chassi					X					
montagem do robô						X				
programação do robô							X			
descrição do funcionamento								X		
fluxograma									X	
batalha de robôs										X
diário de bordo	X	X	X	X	X	X	X	X	X	X

FONTE: Elaborado pelos autores (2024).

5 DESENVOLVIMENTO DO PROJETO

5.1 Descrição do Funcionamento

A Figura 10 apresenta o esquema elétrico de toda a parte eletrônica do projeto.

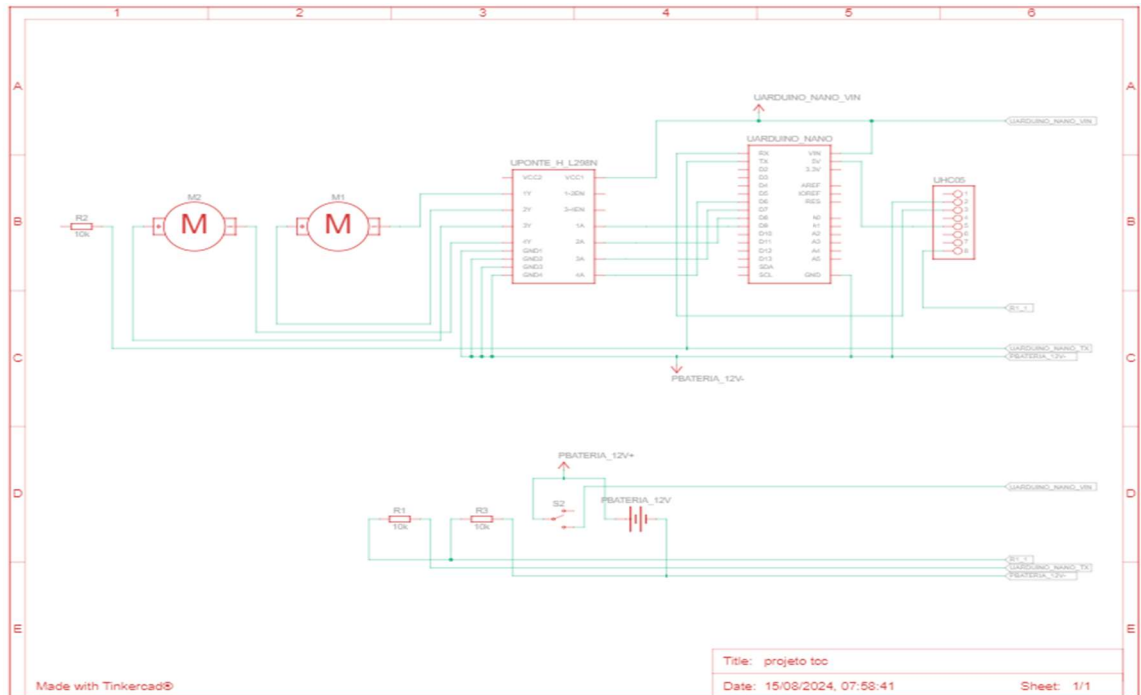


Figura 10 – Esquema elétrico do robô
FONTE: Elaborado pelos autores (2024).

O funcionamento da parte eletrônica é descrito a seguir.

O Arduino Nano é o microcontrolador responsável por atuar como o "cérebro" do robô. Ele recebe os comandos enviados pelo módulo Bluetooth e os converte em sinais que serão utilizados para controlar os motores. O módulo Bluetooth possibilita a comunicação sem fio entre o robô e um dispositivo externo, como um smartphone. Através de um aplicativo, o usuário pode enviar comandos de movimentação, como avançar, retroceder e girar, permitindo o controle remoto do robô.

O L293D, conhecido como ponte H, é responsável por controlar a direção e a velocidade dos motores de corrente contínua. Esse componente recebe os sinais do Arduino Nano e fornece a potência necessária para acionar os motores. Os motores de corrente contínua são os atuadores que realizam o movimento do robô. Eles

respondem aos comandos enviados pelo L293D e geram o movimento desejado, seja para avançar, retroceder ou girar.

Para a montagem do circuito, utiliza-se uma placa de protótipo (breadboard), que conecta todos os componentes de forma organizada, facilitando a construção e ajustes do sistema.

O funcionamento do robô ocorre da seguinte maneira: o usuário envia comandos ao robô por meio do aplicativo Bluetooth, que é transmitido para o módulo Bluetooth conectado ao Arduino Nano. O Arduino, por sua vez, processa esses comandos e envia sinais ao L293D, determinando a direção e a velocidade dos motores. O L293D amplifica os sinais recebidos e aciona os motores de acordo com as ordens enviadas.

5.2 Montagem Física do Projeto

As Figuras 11 a 14 apresentam a sequência da montagem física do nosso projeto.



Figura 11 – Peças utilizadas no robô
FONTE: Elaborado pelos autores (2024).

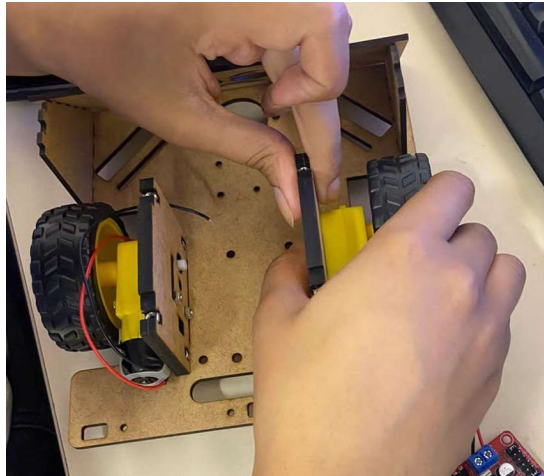


Figura 12 – Instalação das rodas e motores no chassi
FONTE: Elaborado pelos autores (2024).

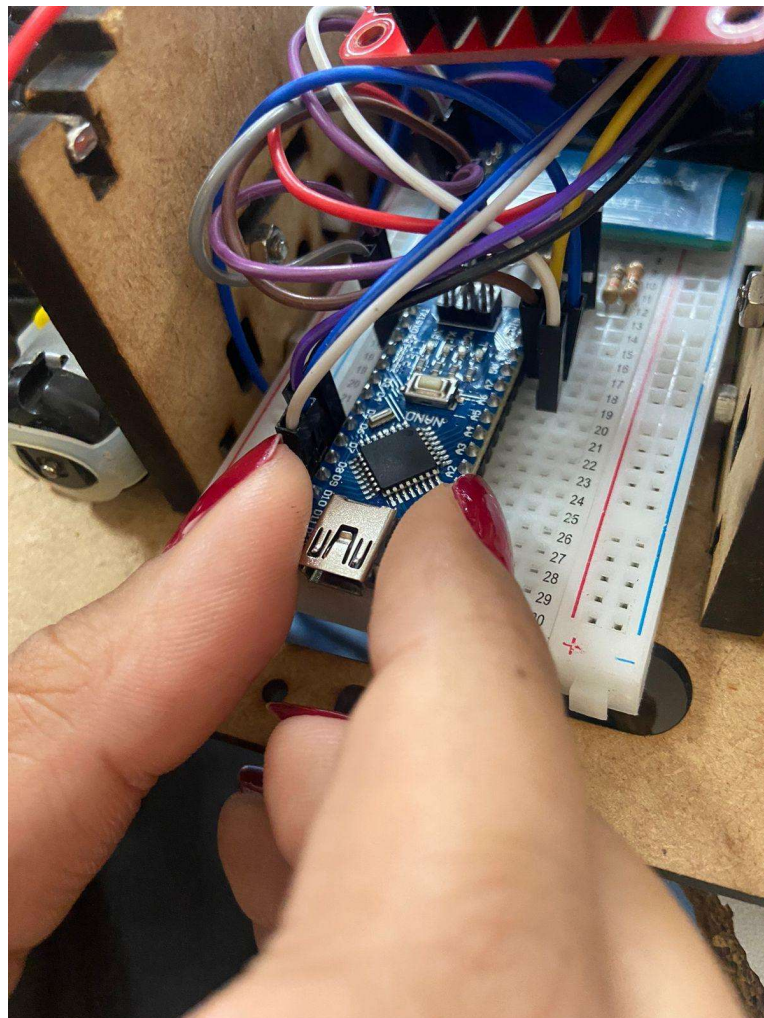


Figura 13 – Instalação do Arduino e módulos no chassi
FONTE: Elaborado pelos autores (2024).



Figura 14 – Instalação da bateria no chassi
FONTE: Elaborado pelos autores (2024).

As Figuras 15 e 16 apresentam os testes finais efetuados em nosso projeto.

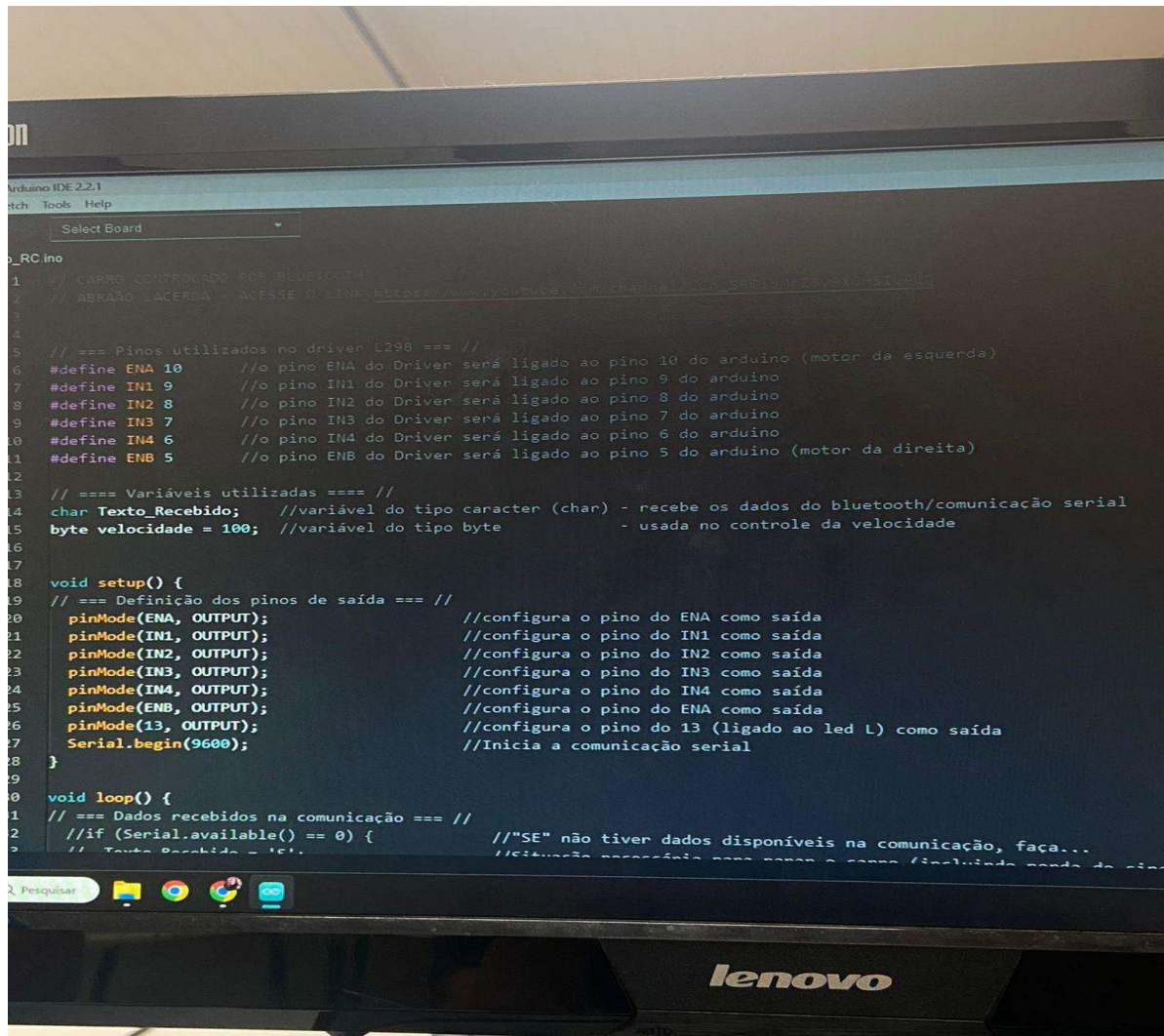


Figura 15 – Teste do hardware
FONTE: Elaborado pelos autores (2024).

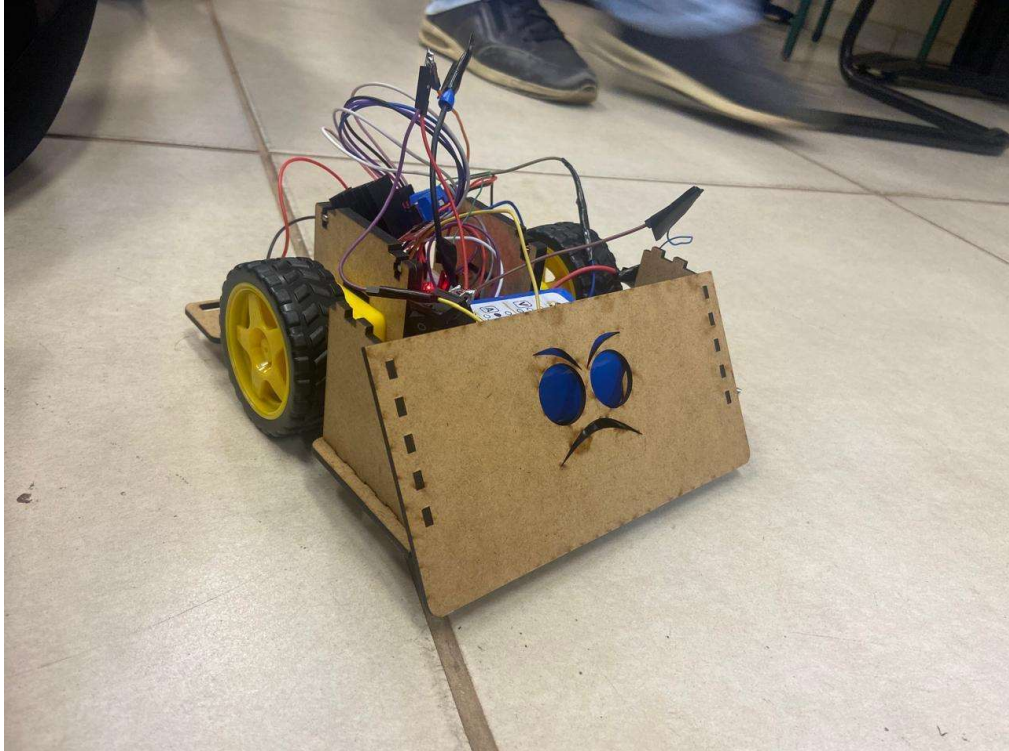


Figura 16 – Teste geral do robô
FONTE: Elaborado pelos autores (2024).

5.3 Desenvolvimento do programa do robô

A programação do robô foi desenvolvida no aplicativo Arduino IDE versão 2.2.1. A Figura 17 apresenta o fluxograma do programa.

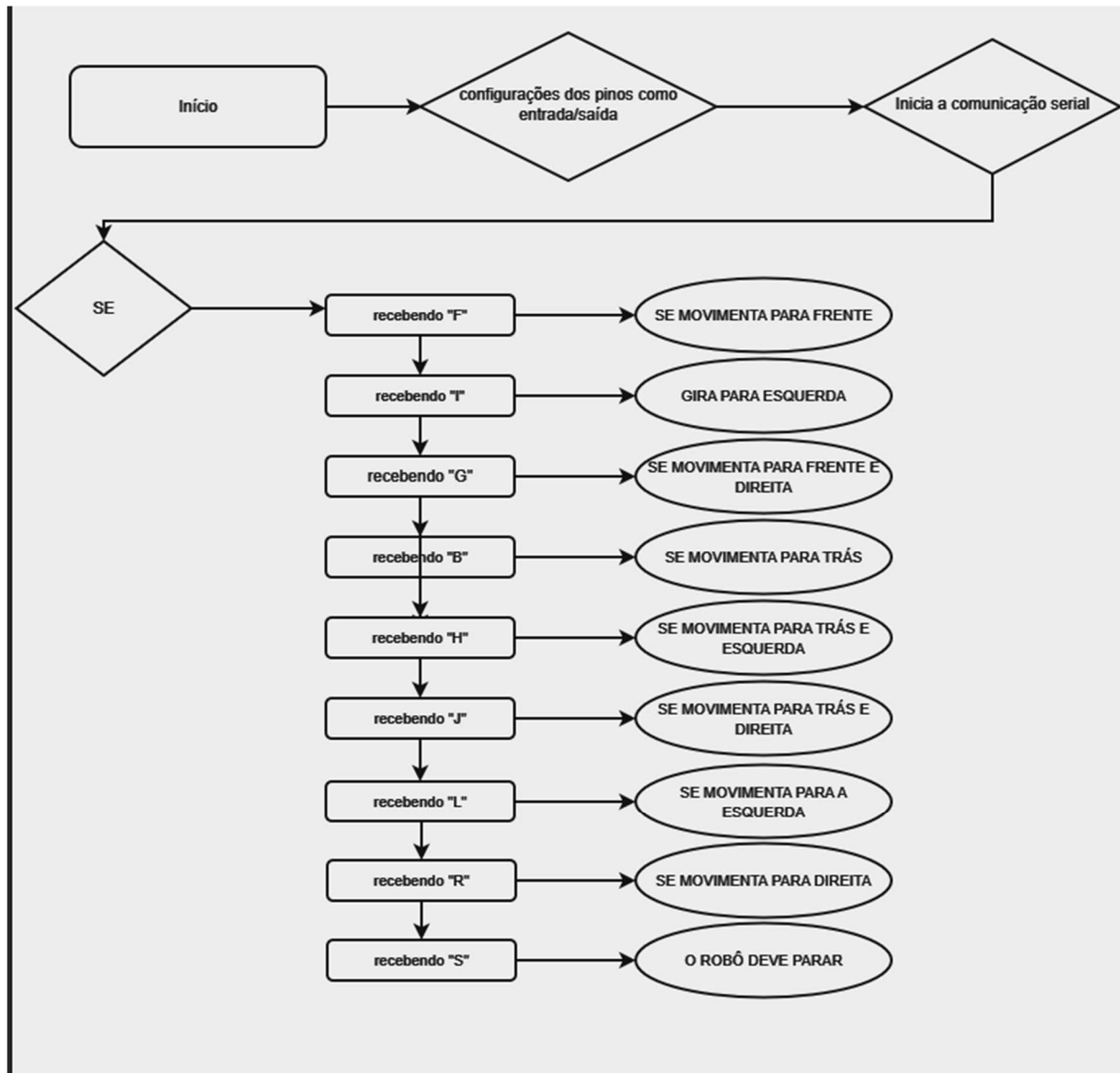


Figura 17 – Fluxograma do programa do robô
 FONTE: Elaborado pelos autores (2024).

Conforme apresentado no fluxograma

Início: O processo começa com a inicialização do sistema, incluindo as configurações dos pinos de entrada/saída.

Comunicação Serial: A comunicação com o robô é estabelecida através de uma porta serial.

SE (Condição): Um comando é recebido pela porta serial. Dependendo do comando recebido, o robô executa uma ação diferente.

Comandos e Ações:

- "F": O robô se move para frente.
- "I": O robô gira para a esquerda.
- "G": O robô se move para frente e para a direita.
- "B": O robô se move para trás.
- "H": O robô se move para trás e para a esquerda.
- "J": O robô se move para trás e para a direita.
- "L": O robô gira para a esquerda.
- "R": O robô se move para a direita.
- "S": O robô para.

O código fonte desenvolvido no Arduino IDE é apresentado no Apêndice A.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento de um robô sumô baseado em princípios de mecânica, eletrônica e programação, com foco no design eficiente para competir em uma arena de sumô. Durante o desenvolvimento, foi possível perceber a importância de um bom design mecânico, já que a estrutura e o equilíbrio do robô foram determinantes para a sua estabilidade na arena. A escolha de materiais leves e resistentes, juntamente com a construção de um chassi adequado, contribuiu para a robustez do robô em enfrentamentos físicos. Este projeto contribuiu para o aprendizado de conceitos de robótica e engenharia. O robô desenvolvido pode servir como base para futuros aprimoramentos, ampliando as possibilidades para novas inovações e desafios no campo da robótica.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Pedro. *Ponte H: O que é, como funciona?*. Disponível em: <https://www.manualdaeletronica.com.br/ponte-h-o-que-e-como-funciona/>. Acesso em: 2019.

Thayron. *Arduino Nano 3.0 - Conheça este pequeno e poderoso membro da família Arduino*. Disponível em: <https://blog.fazedores.com/arduino-nano-3-0/>. Acesso em: 18 mar. 2024.

CORDEIRO, Jade Oliveira; RODOWANSKI, Ivanoé João. Construção e análise de um robô para competição modalidade sumô 3Kg autônomo. *Revista Eletrônica de Ciências Exatas e Tecnológicas*, v. 2, n. 1, 2021.

CRAVO, Edilson. *O que é Arduino: Para que serve e como aplicar suas utilidades?*. Disponível em: <https://blog.kalatec.com.br/o-que-e-arduino/>. Acesso em: 18 mar. 2024.

MACEDO, Paulo Henrique Rezende. *Bluetooth HC06*. Disponível em: <https://dspace.uniube.br/bitstream/123456789/2498/1/TCC.pdf>.

MANTECH. *Arduino Nano*. Disponível em: <https://www.mantech.co.za/datasheets/products/a000005-6s.pdf>. Acesso em: 18 mar. 2024.

ERCÊS, Rafael Santos; RODOWANSKI, Ivanoé João. Estudo do desempenho de diferentes rodas em um robô de sumô. *Estudos em Ciências Exatas e Tecnológicas*, p. 131.

MATTÉDE, Henrique. *Motor de corrente contínua: características e aplicações!*. Disponível em: <https://www.mundodaeletrica.com.br/motor-de-corrente-continua-caracteristicas-e-aplicacoes/>. Acesso em: 16 mar. 2024.

NIEDERAUER, Gabriel N.; CIPRIANI, João P. S.; CARRILHO, Diego O. Desenvolvimento de um robô autônomo para competições de sumô robótico. *XXVIII Congresso Regional de Iniciação Científica e Tecnológica*, 2017.

OLIVEIRA, Wenderson. *Comunicação entre Módulos Bluetooth HC-05 e HC-06*. Disponível em: <https://embarcados.com.br/>. Acesso em: 02 fev. 2016.

ROMÃO, Matheus Paskevics et al. Projeto de robô sumô: guiado por Arduino via controle (celular). 2024.

SANTOS, Gabriel Almeida; PACHECO, Leonardo Freire. Motores elétricos de corrente contínua. Instituto Federal de Educação, Ciência e Tecnologia da Bahia – IFBA, 2003. Disponível em: <https://acesse.dev/nTa2s>. Acesso em: 16 mar. 2024.

SILVA, Leonardo Sousa; COLÉGIO PARAÍSO. Robô sumô de baixo custo.

APÊNDICE

Programa do Robô

```
// === Pinos utilizados no driver L298 === //
#define ENA 10    //o pino ENA do Driver será ligado ao pino 10 do arduino (motor
da esquerda)
#define IN1 9     //o pino IN1 do Driver será ligado ao pino 9 do arduino
#define IN2 8     //o pino IN2 do Driver será ligado ao pino 8 do arduino
#define IN3 7     //o pino IN3 do Driver será ligado ao pino 7 do arduino
#define IN4 6     //o pino IN4 do Driver será ligado ao pino 6 do arduino
#define ENB 5     //o pino ENB do Driver será ligado ao pino 5 do arduino (motor da
direita)

// ===== Variáveis utilizadas ===== //
char Texto_Recebido; //variável do tipo caracter (char) - recebe os dados do
bluetooth/comunicação serial
byte velocidade = 100; //variável do tipo byte          - usada no controle da
velocidade

void setup() {
// === Definição dos pinos de saída === //
pinMode(ENA, OUTPUT);           //configura o pino do ENA como saída
pinMode(IN1, OUTPUT);           //configura o pino do IN1 como saída
pinMode(IN2, OUTPUT);           //configura o pino do IN2 como saída
pinMode(IN3, OUTPUT);           //configura o pino do IN3 como saída
pinMode(IN4, OUTPUT);           //configura o pino do IN4 como saída
pinMode(ENB, OUTPUT);           //configura o pino do ENA como saída
pinMode(13, OUTPUT);            //configura o pino do 13 (ligado ao led L) como
saída
Serial.begin(9600);             //Inicia a comunicação serial
```

```
}
```

```
void loop() {
```

```
// === Dados recebidos na comunicação === //
```

```
  //if (Serial.available() == 0) {          //"SE" não tiver dados disponíveis na
  comunicação, faça...
```

```
    // Texto_Recebido = 'S';                //Situação necessária para parar o carro
  (incluindo perda de sinal)
```

```
  // }
```

```
  if (Serial.available()>0) {              //"SE" tiver dados disponíveis na comunicação,
  faça...
```

```
    Texto_Recebido = Serial.read();        //o valor da leitura serial é atribuído à
  variável Texto_Recebido
```

```
  }
```

```
// === Movimentação do carro === //
```

```
  if (Texto_Recebido == 'F') {             //"SE" a letra recebida for "F", faça...
    Serial.println("Para Frente");         //imprime o texto "Para Frente" no monitor serial
    analogWrite (ENA, velocidade);         //envia ao pino ENA o valor da velocidade
  do motor 1
```

```
    analogWrite (ENB, velocidade);        //envia ao pino ENB o valor da velocidade
  do motor 2
```

```
    digitalWrite(IN1, 1);                 //envia nível lógico alto para o pino IN1 //liga o
  motor 1
```

```
    digitalWrite(IN2, 0);                 //envia nível lógico baixo para o pino IN2 //para
  frente
```

```
    digitalWrite(IN3, 1);                 //envia nível lógico alto para o pino IN3 **liga o
  motor 2
```

```
    digitalWrite(IN4, 0);                 //envia nível lógico baixo para o pino IN4 **para
  frente
```

```
    digitalWrite(13,1);                   //liga o led L ligado ao pino 13
```

```
  }
```

```

else if (Texto_Recebido == 'I') {    //"MAS SE" a letra recebida for "I", faça...
    Serial.println("Para Frente e Esquerda"); //imprime o texto "Para Frente e
Esquerda" no monitor serial
    analogWrite (ENA, velocidade);        //envia ao pino ENA o valor da velocidade
do motor 1
    analogWrite (ENB, velocidade-150);    //envia ao pino ENB o valor da velocidade-
150 do motor 2
    digitalWrite(IN1, 1);                 //envia nível lógico alto para o pino IN1    //liga o
motor 1
    digitalWrite(IN2, 0);                 //envia nível lógico baixo para o pino IN2    //para
frente
    digitalWrite(IN3, 1);                 //envia nível lógico alto para o pino IN3    **liga o
motor 2
    digitalWrite(IN4, 0);                 //envia nível lógico baixo para o pino IN4    **para
frente
}

```

```

else if (Texto_Recebido == 'G') {    //"SE" a letra recebida for igual a 'G', o carro
se movimenta para Frente Direita.
    Serial.println("Para Frente e Direita"); //imprime o texto "Para Frente e Direita" no
monitor serial
    analogWrite (ENA, velocidade-150);    //envia ao pino ENA o valor da velocidade-
150 do motor 1
    analogWrite (ENB, velocidade);        //envia ao pino ENB o valor da velocidade
do motor 2
    digitalWrite(IN1, 1);                 //envia nível lógico alto para o pino IN1    //liga o
motor 1
    digitalWrite(IN2, 0);                 //envia nível lógico baixo para o pino IN2    //para
frente
    digitalWrite(IN3, 1);                 //envia nível lógico alto para o pino IN3    **liga o
motor 2
    digitalWrite(IN4, 0);                 //envia nível lógico baixo para o pino IN4    **para
frente

```

```
}
```

```
else if (Texto_Recebido == 'B') {      //"SE" a letra recebida for igual a 'B', o carro
se movimenta para Trás.
```

```
    Serial.println("Para Trás");        //imprime o texto "Para Trás" no monitor serial
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da velocidade
do motor 1
```

```
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da velocidade
do motor 2
```

```
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino IN1  //liga o
motor 1
```

```
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino IN2  //para
trás
```

```
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino IN3  **liga o
motor 2
```

```
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino IN4  **para trás
}
```

```
else if (Texto_Recebido == 'H') {      //"SE" a letra recebida for igual a 'H', o carro
se movimenta para Trás e esquerda.
```

```
    Serial.println("Para Trás e Esquerda"); //imprime o texto "Para Trás e Esquerda"
no monitor serial
```

```
    analogWrite (ENA, velocidade-150);  //envia ao pino ENA o valor da velocidade-
150 do motor 1
```

```
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da velocidade
do motor 2
```

```
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino IN1  //liga o
motor 1
```

```
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino IN2  //para
trás
```

```
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino IN3  **liga o
motor 2
```

```
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino IN4  **para trás
```

```
}
```

```
else if (Texto_Recebido == 'J') {      //"SE" a letra recebida for igual a 'J', o carro se
movimenta para Trás e direita.
```

```
    Serial.println("Para Trás e Direita"); //imprime o texto "Para Trás e Direita" no
monitor serial
```

```
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da velocidade
do motor 1
```

```
    analogWrite (ENB, velocidade-150);  //envia ao pino ENB o valor da velocidade-
150 do motor 2
```

```
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino IN1  //liga o
motor 1
```

```
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino IN2  //para
trás
```

```
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino IN3  **liga o
motor 2
```

```
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino IN4  **para trás
}
```

```
else if (Texto_Recebido == 'L') {      //"SE" a letra recebida for igual a 'L', o carro
se movimenta para a esquerda.
```

```
    Serial.println("Para Esquerda");    //imprime o texto "Para Esquerda" no monitor
serial
```

```
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da velocidade
do motor 1
```

```
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da velocidade
do motor 2
```

```
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino IN1  //liga o
motor 1
```

```
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino IN2  //para
trás
```

```
    digitalWrite(IN3, 1);                //envia nível lógico alto para o pino IN3  **liga o
motor 2
```

```

    digitalWrite(IN4, 0);          //envia nível lógico baixo para o pino IN4 **para
frente
}

else if (Texto_Recebido == 'R') {    //"SE" a letra recebida for igual a 'R', o carro
se movimenta para a direita.
    Serial.println("Para Direita");    //imprime o texto "Para Direita" no monitor serial
    analogWrite (ENA, velocidade);    //envia ao pino ENA o valor da velocidade
do motor 1
    analogWrite (ENB, velocidade);    //envia ao pino ENB o valor da velocidade
do motor 2
    digitalWrite(IN1, 1);            //envia nível lógico alto para o pino IN1    //liga o
motor 1
    digitalWrite(IN2, 0);            //envia nível lógico baixo para o pino IN2    //para
frente
    digitalWrite(IN3, 0);            //envia nível lógico baixo para o pino IN3    **liga o
motor 2
    digitalWrite(IN4, 1);            //envia nível lógico alto para o pino IN4    **para trás
}

else if (Texto_Recebido == 'S') {    //"SE" a letra recebida for igual a 'S', o carro
deve parar.
    Serial.println("Parado");        //imprime o texto "Parado" no monitor serial
    digitalWrite(IN1, 0);            //envia nível lógico baixo para o pino IN1
//desliga
    digitalWrite(IN2, 0);            //envia nível lógico baixo para o pino IN2    //o
motor 1
    digitalWrite(IN3, 0);            //envia nível lógico baixo para o pino IN3    **desliga
    digitalWrite(IN4, 0);            //envia nível lógico baixo para o pino IN4    **o
motor 2
    digitalWrite(13,0);              //desliga o led L ligado ao pino 13
}
}

```