

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC SYLVIO DE MATTOS CARVALHO

CURSO TÉCNICO EM MECATRÔNICA M-TEC

LIVIA QUERUBIN

MARIANA CRISTINA PERRI

MATHEUS HENRIQUE BONINI BARROS

RENAN LINHARES PEREIRA

RIAN DE OLIVEIRA CASTRO

**DESENVOLVIMENTO E OTIMIZAÇÃO PARA ROBÔ SUMÔ:
ABORDAGEM PRÁTICA**

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	CRONOGRAMA DE ATIVIDADES	7
4.1	Atividades e Metas 1º Semestre de 2024	7
4.2	Atividades e Metas 2º Semestre de 2024.....	8
5	DESENVOLVIMENTO DO PROJETO.....	9
5.1	Descrição do Funcionamento.....	9
5.2	Montagem Física do Projeto	10
	Conforme apresentado no fluxograma, este diagrama descreve o processo de controle do robô de sumo, onde ele recebe comandos via Bluetooth e executa ações específicas com base nesses comandos. O controle é feito de forma sequencial, com verificações e respostas programadas para movimentar o robô e controlar seus LEDs. A seguir, explicamos o fluxo de cada etapa:.....	20
	• Início (A).....	21
6	CONSIDERAÇÕES FINAIS	22
	APÊNDICE	24

1 INTRODUÇÃO

Nos últimos anos, a popularidade dos esportes robóticos tem crescido exponencialmente, entre essas competições, a emocionante modalidade de luta de robôs de sumô. Inspirada no antigo esporte japonês de sumô, essa competição moderna coloca à prova não a força humana, mas sim a engenhosidade e habilidade dos projetistas e programadores de robôs.

Nesse cenário desafiador, robôs especialmente projetados, equipados com sensores, motores e algoritmos avançados, se enfrentam em um ringue circular, com o objetivo de empurrar o oponente para fora da arena ou de incapacitá-lo de alguma forma. Esses robôs, muitas vezes compactos e ágeis, são construídos com o propósito específico de maximizar sua eficiência na luta e, ao mesmo tempo, resistir aos ataques do oponente.

Neste trabalho, exploraremos a fascinante dinâmica da luta de robôs de sumô, investigando os princípios fundamentais por trás do projeto e funcionamento desses autômatos, bem como as estratégias utilizadas para maximizar suas chances de sucesso na arena. Além disso, examinaremos o impacto dessa modalidade de competição no avanço da tecnologia robótica, destacando sua importância como plataforma de aprendizado e inovação.

Por fim, este estudo busca não apenas oferecer uma visão abrangente da luta de robôs de sumô, mas também estudar e adquirir conhecimento sobre o universo da robótica não apenas dos “Robôs de sumô” mas sobre robótica.

2 OBJETIVOS (GERAL E ESPECÍFICOS)

2.1 Objetivo Geral

O objetivo geral do projeto é:

- Desenvolver um robô capaz de competir em uma arena de sumô, tentando derrubar ou desviar o oponente.

2.2 Objetivos Específicos

Os objetivos específicos do projeto são:

- Aprender a programar os componentes eletrônicos do robô, integrando sistemas que permitam a comunicação entre o controle e o robô, possibilitando um funcionamento ágil e responsivo;
- Desenvolver uma compreensão das táticas de combate, como atacar, desviar e se posicionar estrategicamente na arena;
- Colaboração entre os membros da equipe, cada um contribuindo com suas habilidades em áreas como eletrônica, programação e design;
- Realizar testes práticos para ajustar a operação do robô, melhorando sua agilidade, força e capacidade de resposta aos comandos do controle;
- Proporcionar uma experiência prática em robótica, programação e eletrônica, permitindo que os participantes apliquem conhecimentos teóricos de forma prática.

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.

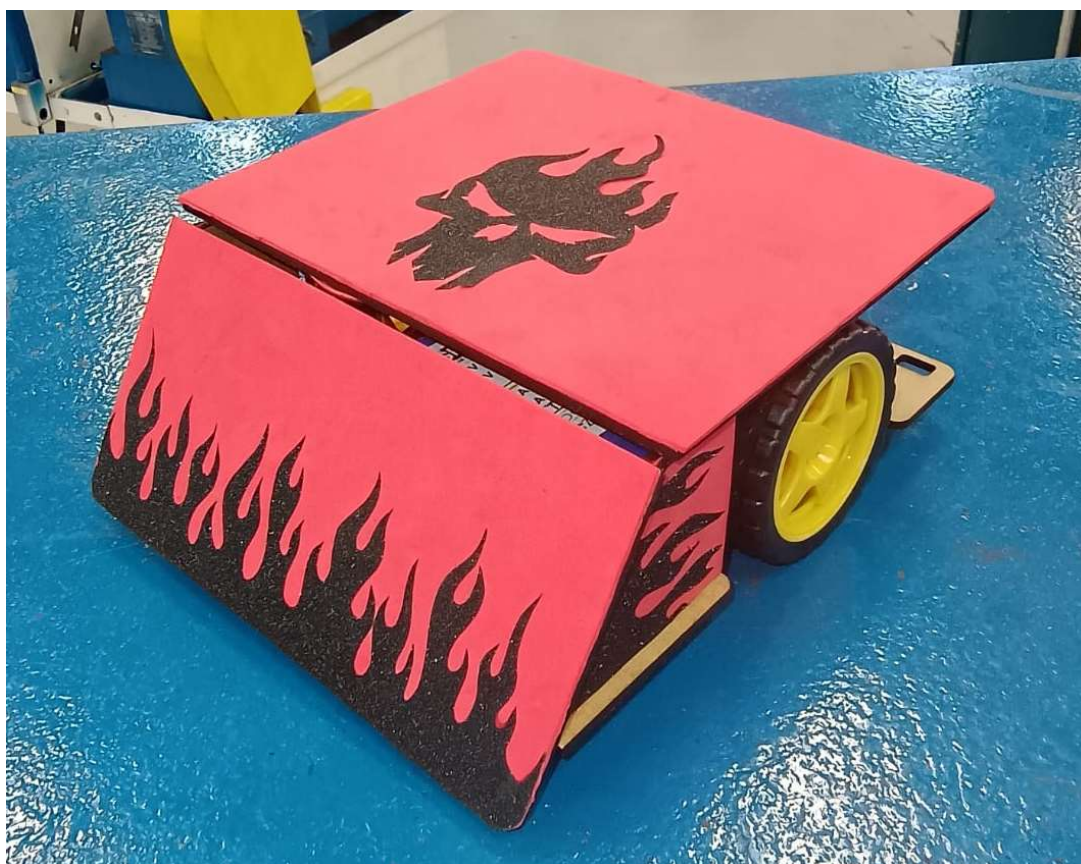


Figura 1 – Foto do projeto montado.
FONTE: Elaborado pelos autores (2024).

3.2 Materiais Utilizados e suas Especificações Técnicas

Tabela 1 – Lista de Materiais:

Material	Quantidade
Arduino nano	1
L298n (Ponte H)	1
Motor de passo com caixa de redução	2
Protoboard	1
Bateria de Lítio 12V	1
Chassis	1
Rodas emborrachadas	2
Botão com trava	1
Modulo bluetooth hc-05	1
Cabo jumper com quatro ligações	1
Roda esférica	1

FONTE: Autoria própria (2024).

Alguns dos materiais listados na Tabela 1 foram custeados pelo Professor Rogério Varavallo.

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2024

Planejamos a escolha dos integrantes, também dos itens que estaremos utilizando ao longo do ano, a separação de afazeres de cada pessoa.

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

Atividade	Fev.		Março		Abril		Maio		Junho	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
Montagem da equipe	x									
Definição do nome do robô		x								
Definição dos objetivos do projeto		x								
Definição da lista de materiais / orçamentos			x							
Construção do chassi				x						
Redação do relatório até o capítulo 3				x	x					
Redação do capítulo 3					x	x	x	x		
Montagem do robô					x	x	x	x		
Programação do robô							x	x		
Batalha de Robôs									x	x

FONTE: Elaborado pelos autores (2024).

4.2 Atividades e Metas 2º Semestre de 2024

Realizamos a compra dos itens do robô, fazendo testes e ajustes para sua melhoria, finalizando o boneco e concluindo o protótipo.

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
Entrega final do boneco/diário de bordo										X
Revisão do diário de bordo/Boneco		X	X	X	X	X				
Compra dos itens do robô			X	X						
Teste do robô					X	X	X	X	X	
Finalização do diário de bordo/boneco							X	X	X	
Finalização do robô								X	X	X
Apresentação do TCC finalizado										X
Gravação do vídeo									X	
Montagem do chassi							X	X	X	

FONTE: Autoria própria (2024).

5 DESENVOLVIMENTO DO PROJETO

5.1 Descrição do Funcionamento

A Figura 2 apresenta o esquema elétrico de toda a parte eletrônica do projeto.

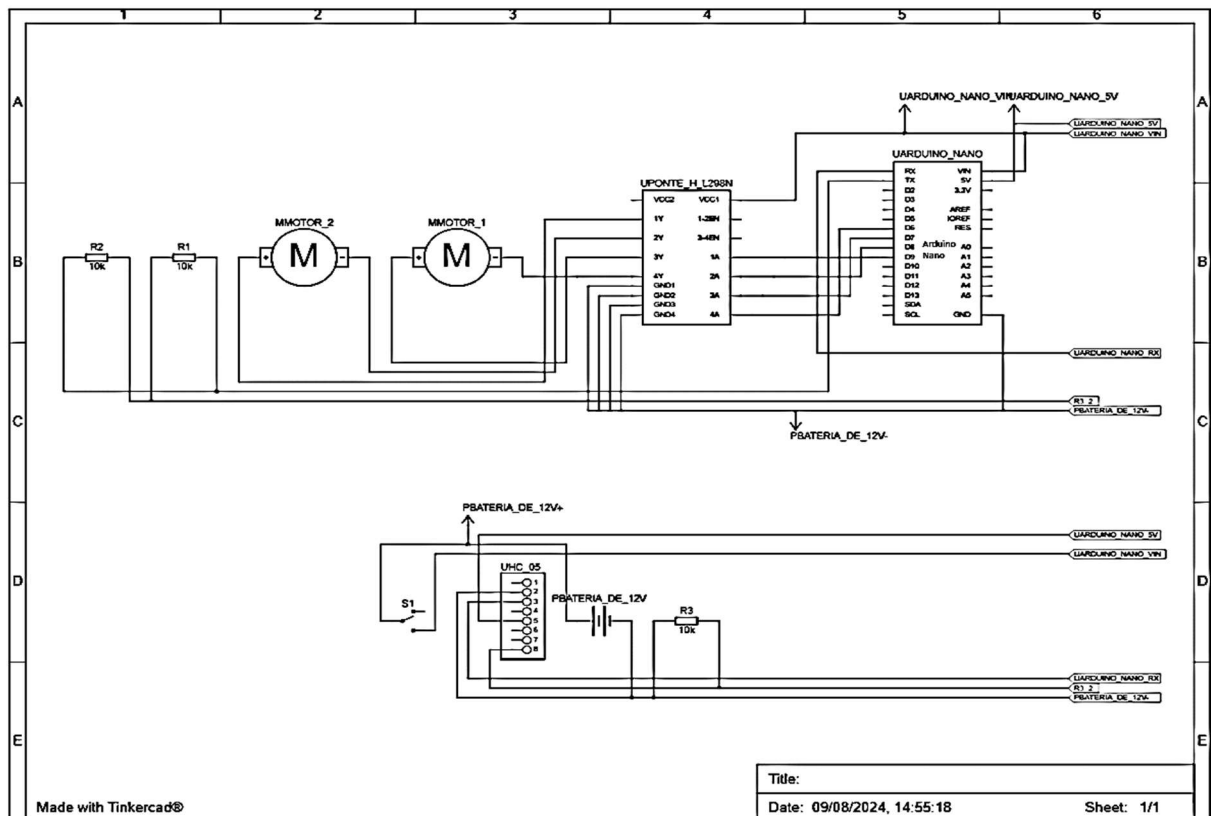


Figura 2: Esquema elétrico.
Fonte: Tinkercad (2024).

O funcionamento da parte eletrônica é descrito a seguir.

A bateria que é responsável por alimentar todo o projeto, energiza o Arduino que manda os sinais necessários para o módulo bluetooth e a ponte H que controla a velocidade e a direção em que os motores vão girar de acordo com os sinais recebidos do Arduino, permitindo que o robô ande em todas as direções. Todas as ligações foram feitas com cabos jumper, que são pequenos fios condutores próprios para esses tipos de projetos que utilizam protoboard.

5.2 Montagem Física do Projeto

Uma bateria, ou acumulador elétrico, é um dispositivo projetado para armazenar energia elétrica sob a forma de energia química e liberá-la novamente como energia elétrica quando necessário, com uma eficiência específica.

(Bateria de 12V 4.4 A)



Figura 3: Arduino nano.
Fonte: Autoria própria (2024).

O módulo Bluetooth HC-05 é uma ferramenta popular para a comunicação sem fio em projetos eletrônicos. Ele permite que dispositivos como microcontroladores, Arduino, e até mesmo computadores se comuniquem sem fio através do protocolo Bluetooth.



Figura 4: Arduino nano.
Fonte: Autoria própria (2024).

A ponte H L298N é um circuito integrado que permite controlar a direção e a velocidade de motores DC (corrente contínua). É uma ponte dupla H, ou seja, é capaz de controlar dois motores de corrente contínua (DC) bidirecionais ou um motor de passo bipolar em um único chip. Ela é amplamente utilizada em projetos de robótica e automação devido à sua confiabilidade e facilidade de uso.

Este componente é capaz de controlar a direção e a velocidade dos motores DC, permitindo que eles girem nos dois sentidos e em diferentes velocidades. Isso é possível graças aos quatro transistores de potência dentro do chip, que são acionados de forma a permitir a polarização correta do motor.

Além disso, a ponte H L298N possui proteção contra sobrecorrente e contra inversão de polaridade, garantindo a segurança do circuito e dos componentes conectados a ele.

Para controlar a ponte H L298N, é necessário um microcontrolador, como Arduino. O microcontrolador envia sinais PWM (modulação por largura de pulso) para os pinos de controle da ponte H, ajustando assim a velocidade dos motores.

A conexão da ponte H L298N é relativamente simples. Ela requer alimentação externa para os motores, geralmente entre 5V e 12V, e uma fonte de alimentação separada para a lógica do chip, que normalmente é de 12V.

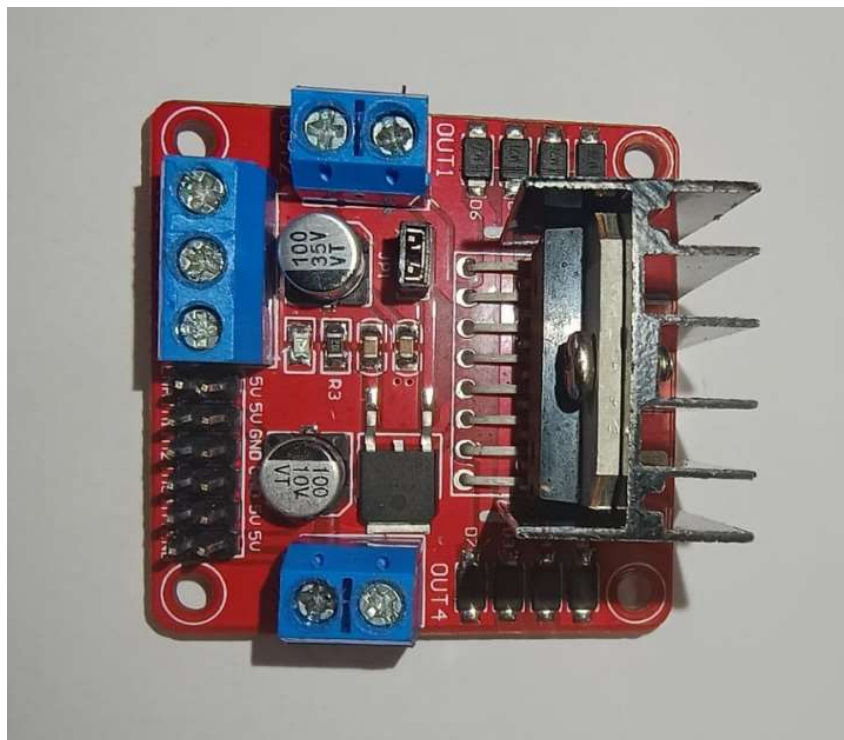


Figura 5: Arduino nano.
Fonte: Autoria própria (2024).

Um motor com caixa de redução, também conhecido como motor redutor, é um dispositivo que combina um motor elétrico com um sistema de redução de velocidade. Esse conjunto é projetado para reduzir a velocidade de rotação do motor enquanto aumenta o torque (força de rotação) disponível na saída, tornando-o ideal para aplicações que exigem movimento lento e forte.

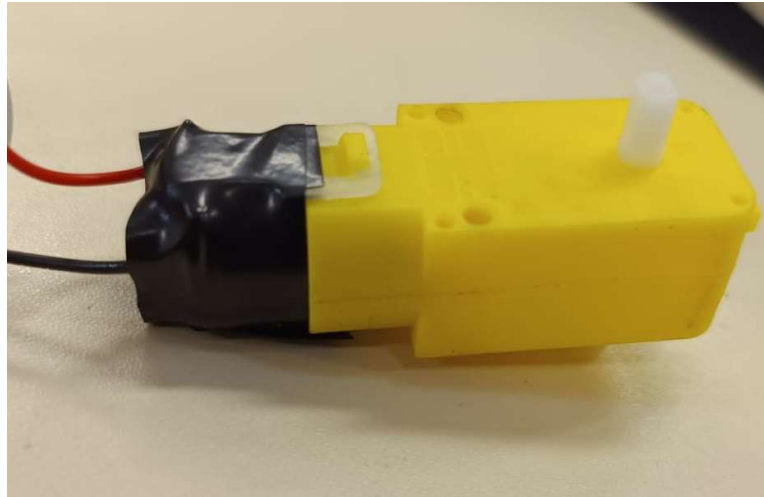


Figura 6: Arduino nano.
Fonte: Autoria própria (2024).

Roda emborrachada.



Figura 7: Arduino nano.
Fonte: Autoria própria (2024).

Resistores são componentes eletrônicos passivos utilizados para controlar a corrente elétrica em circuitos, limitando ou regulando a quantidade de fluxo de corrente. Eles são feitos de materiais que oferecem resistência à passagem de eletricidade.



Figura 8: Resistores.
Fonte: Autoria própria (2024).

O Arduino Nano é uma placa de desenvolvimento eletrônico compacta e versátil. Projetado para ocupar menos espaço e oferecer maior portabilidade, o Nano mantém muitos dos recursos e funcionalidades.

Uma das características mais distintivas do Arduino Nano é seu tamanho compacto. Com dimensões de aproximadamente 18 x 45 mm, o Nano é significativamente menor do que outras placas Arduino, tornando-o ideal para projetos onde o espaço é limitado. Apesar de seu tamanho reduzido, ele oferece uma ampla gama de recursos, incluindo 14 pinos digitais (6 dos quais podem ser usados como saídas PWM), 8 pinos analógicos e interfaces USB e UART para comunicação com outros dispositivos.



Figura 9: Arduino nano.
Fonte: Autoria própria (2024).



Figura 10 – Instalação das rodas e motores no chassi.
FONTE: Autoria própria (2024).

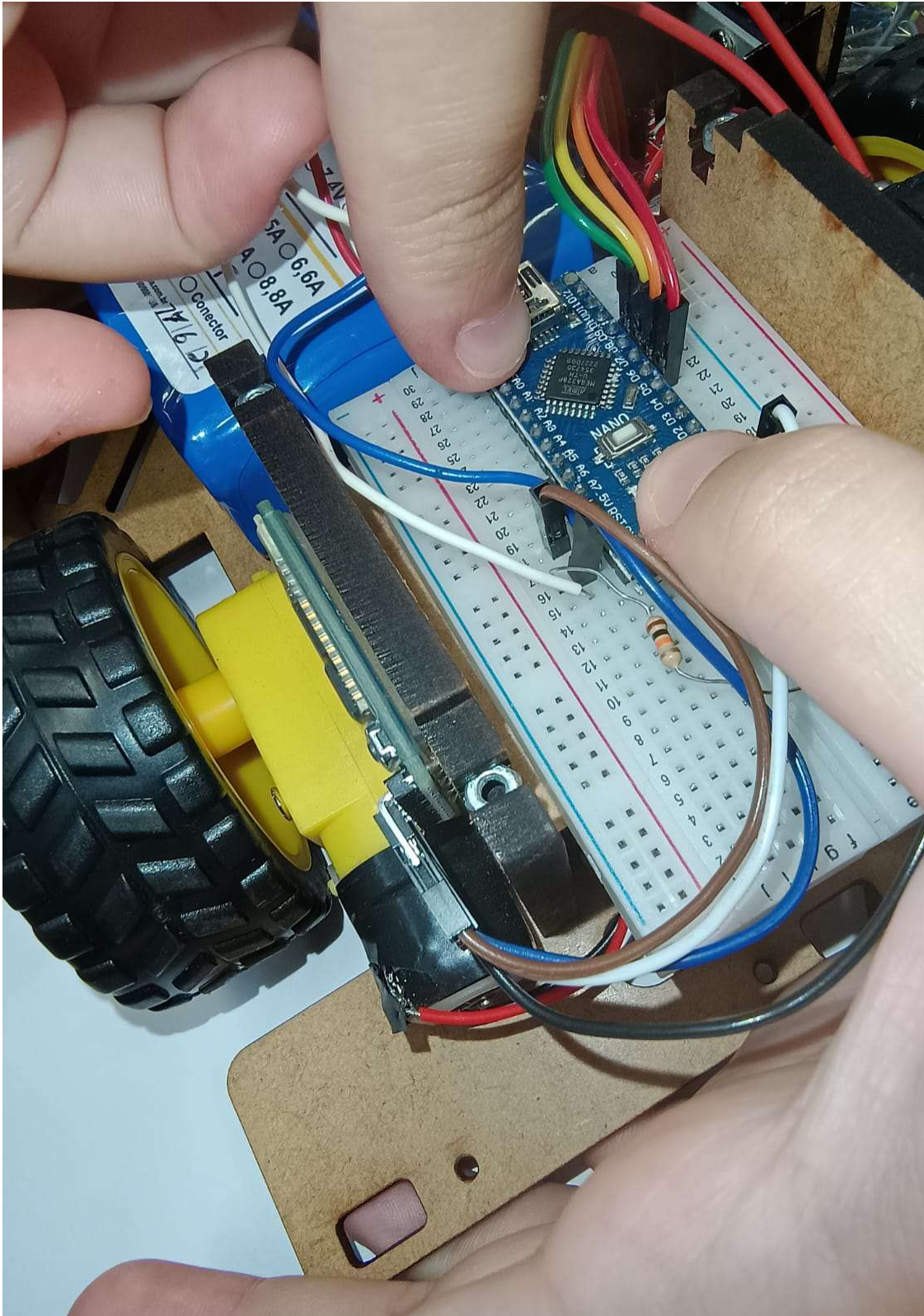


Figura 11 – Instalação do Arduino e módulos no chassi.
FONTE: Autoria própria (2024).

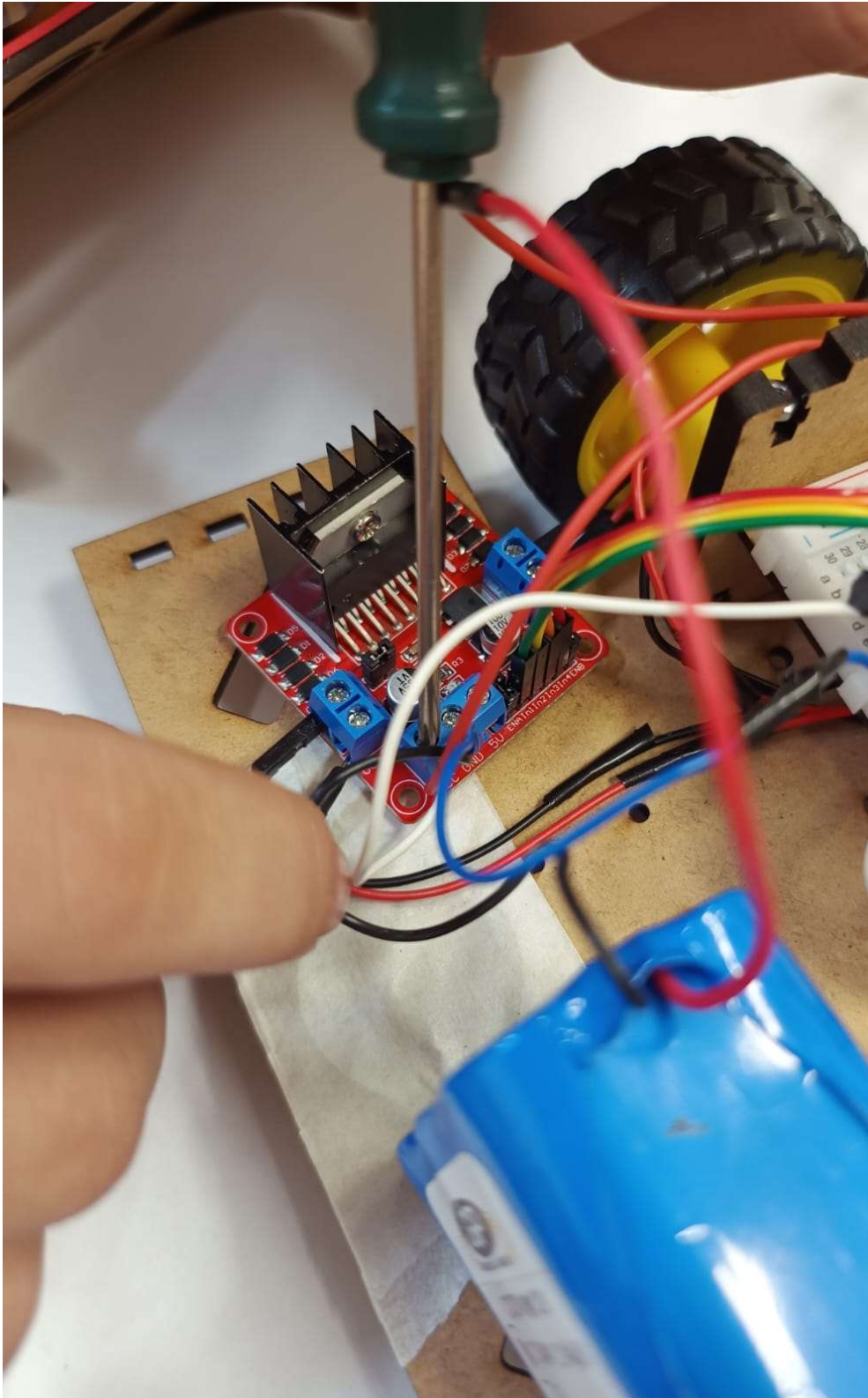


Figura 12 – Instalação da bateria no chassi.
FONTE: Autoria própria (2024).

As Figuras apresentam os testes finais efetuados em nosso projeto.



Figura 13 – Teste do hardware.
FONTE: Elaborado pelos autores (2024).



Figura 14 – 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.

A Figura 15 apresenta o fluxograma do programa.

Z



Figura 15 – Fluxograma do programa do robô.
FONTE: Elaborado pelos autores (2024).

Conforme apresentado no fluxograma, este diagrama descreve o processo de controle do robô de sumô, onde ele recebe comandos via Bluetooth e executa ações específicas com base nesses comandos. O controle é feito de forma sequencial, com verificações e respostas programadas para movimentar o robô e controlar seus LEDs. A seguir, explicamos o fluxo de cada etapa:

- Início (A)

O sistema inicia o processo e começa a esperar por comandos, entrando no estado "A - Aguardar Dados".

- Recebimento de Dados (B)

O robô então aguarda a recepção de dados via Bluetooth (**B - Receber dados via Bluetooth**). Após o recebimento, o sistema verifica se os dados foram recebidos com sucesso.

- Verificação de Dados (C)

Após a recepção dos dados, o sistema verifica se há dados válidos. Se não houver dados, ele retorna ao estado de **aguardar dados (A)**. Caso os dados tenham sido recebidos corretamente, o processo segue para a próxima etapa de **leitura dos dados seriais (D)**.

Interpretação e Ação com Base nos Dados (D, E, H, K, M, O)

O sistema então interpreta os dados recebidos e decide qual ação tomar com base no comando:

- **Comando 'F'**: Se o comando for 'F', o robô se move para a frente (**F Mover o robô para frente**) e acende o LED no pino 13 (**G - Ligar LED no pino 13**).
- **Comando 'B'**: Se o comando for 'B', o robô se move para trás (**I - Mover o robô para trás**) sem mudar o estado do LED (**J - Nenhuma mudança de estado do LED**).
- **Comando 'L'**: Se o comando for 'L', o robô se move para a esquerda (**L – Mover o robô para a esquerda**), novamente sem alteração no LED.
- **Comando 'R'**: Se o comando for 'R', o robô se move para a direita (**N – Mover o robô para a direita**) e o controle do LED segue o mesmo comportamento de não mudança.
- **Comando 'S'**: Se o comando for 'S', o robô **para imediatamente (P – Parar o robô)** e desliga o LED (**Q - Desligar LED no pino 13**).
- **Comando desconhecido**: Se os dados recebidos não correspondem a nenhum dos comandos esperados, o robô **ignora o comando (R - Ignorar comando)**.

- Controle dos Motores e LED (S)

Independentemente do movimento executado, o sistema continua a controlar os motores do robô e o LED. O controle dos motores é feito via **PWM** (modulação por largura de pulso) nos pinos **ENA e ENB**, ajustando a intensidade e direção dos motores para o movimento desejado. Esse controle é indicado no estado **S - Controlar motores via PWM em ENA e ENB**.

O ciclo de controle dos motores e LED garante que o robô responda de maneira precisa a cada comando, mantendo o comportamento desejado enquanto o sistema aguarda os próximos dados via Bluetooth.

Controle dos Motores Esquerdo e Direito (T1 e T2)

Dentro do controle do movimento, o sistema gerencia individualmente os motores do robô, com o motor esquerdo sendo controlado através dos pinos **IN1 e IN2** (T1) e o motor direito sendo controlado através dos pinos **IN3 e IN4** (T2). O controle específico dos motores assegura que os movimentos do robô sejam feitos com precisão, seja para frente, para trás, ou em qualquer direção.

Ciclo de Reinicialização (A)

Após executar a ação correspondente ao comando, o sistema reinicia o ciclo e volta para o estado de **aguardar novos dados (A)**. Isso garante que o robô esteja sempre pronto para responder a novos comandos a qualquer momento.

6 CONSIDERAÇÕES FINAIS

Ao longo do desenvolvimento deste projeto, enfrentamos diversos desafios, tanto técnica quanto relacionadas a questões de hardware. Alguns componentes utilizados apresentaram falhas, exigindo substituições durante o processo. Além disso, a parte teórica do projeto também representou um ponto crítico, demandando uma longa pesquisa e aprofundamento. Outro aspecto que se mostrou desafiador foi o trabalho em equipe, que exigiu uma boa organização e comunicação para superar obstáculos e alinhar as diferentes fases do projeto. Apesar dos desafios encontrados, o trabalho conseguiu alcançar seus objetivos.

REFERÊNCIAS BIBLIOGRÁFICAS

BETRYBE. **Pesquisa do Arduino**. Disponível em: <https://blog.betrybe.com> acesso em: 22/08/2024.

FABRICA DE BOLSO. **Pesquisa sobre a ponte H-L298N**. Disponível em: <https://fabricadebolso.com.br>. Acesso em: 22/08/2024.

MOKO BLUE. **Pesquisa sobre o modulo bluetooth HC-05**. Disponível em: <https://www.mokoblue.com>. Acesso em 22/08/2024.

GOLD ENERGY. **Pesquisa sobre a bateria**. Disponível em: <https://goldenergy.pt>. Acesso em 29/08/2024.

FABRICA DE BOLSO. **Pesquisa sobre protoboard**. Disponível em: <https://fabricadebolso.com.br>. Acesso em: 29/08/2024.

ZANINI RANK. **Pesquisa sobre motores com caixa de redução**. Disponível em: <https://zaninirenk.com.br>. Acesso em 29/08/2024.

TINKERCAD. **Aplicativo utilizado esquema elétrico**. Disponível em: <https://www.tinkercad.com/dashboard>. Acessado em 25/11/2024.

MIRO. **Site utilizado para fazer o fluxograma**. Disponível em: <https://miro.com>. Acessado em 26/11/2024.

ARDUINO CAR. **Aplicativo utilizado para controlar o protótipo**.

APÊNDICE

Programa do Robô

// CARRO CONTROLADO POR BLUETOOTH

// ABRAÃO LACERDA - ACESSE O LINK

https://www.youtube.com/channel/UCn_SRCloMrZ5y9XOHSIUUFUQ

// === 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 = 70; //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
}
}

```