

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC SYLVIO DE MATTOS CARVALHO

Curso Técnico em Mecatrônica M-tec

Alan Antônio

Gustavo Henrique Viana Moreira

José Henrique Nascimento Bessa

Jucielton Alves Batista

Vinicius Gabriel David Razo

ROBÔ SHARP BLADE

Matão, SP

2024

SUMÁRIO

1	INTRODUÇÃO.....	4
2	OBJETIVOS (GERAL E ESPECÍFICOS).....	5
2.1	Objetivo Geral.....	5
2.2	Objetivos Específicos.....	5
3	DESCRIÇÃO DO PROJETO	6
3.1	Apresentação Geral do Projeto	6
3.2	Materiais Utilizados e suas Especificações Técnicas.....	6
4	CRONOGRAMA DE ATIVIDADES.....	13
4.1	Atividades e Metas 1º Semestre de 2024	13
4.2	Atividades e Metas 2º Semestre de 2024	13
5	DESENVOLVIMENTO DO PROJETO	15
5.1	Descrição do Funcionamento.....	15
5.2	Montagem Física do Projeto	16
5.3	Desenvolvimento do programa do robô	19
6	CONSIDERAÇÕES FINAIS.....	23
	REFERÊNCIAS BIBLIOGRÁFICAS:	24
	APÊNDICE	25

1 INTRODUÇÃO

Este documento apresenta o Projeto Técnico Robô Sharp Blade 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.

Isto posto, este documento encontra-se assim dividido:

No Capítulo 2, apresenta os objetivos geral e específicos desse documento.

No Capítulo 3, é feita uma apresentação geral do projeto e a lista de materiais utilizadas para a sua montagem física

O Capítulo 4 apresenta os cronogramas de atividades planejados e cumpridos no 1º e no 2º semestres de 2024 para a concretização do projeto.

O Capítulo 5 faz a descrição detalhada do projeto físico, apresentando o seu funcionamento, a sua montagem física e o programa aplicativo desenvolvido.

Na sequência são apresentadas as considerações finais sobre o projeto, apresentando as dificuldades encontradas e contornadas, e o resultado final obtido.

E, finalmente nas Referências Bibliográficas, são apresentadas toda a bibliografia utilizada para o desenvolvimento desse projeto.

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 boba 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, instalada 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

Componente	Valor unitário (R\$)	Quantidade	Valor total (R\$)
Arduino nano	45,00	1	45,00
Mini protoboard	15,00	1	15,00
Motor Shield L298N	13,00	1	13,00
Motor DC	15,00	2	30,00
Roda de borracha	9,00	2	18,00
Roda boba	16,77	1	10,36
Chassi	20,00	1	20,00
Módulo Bluetooth HC-06	37,90	1	14,20
Bateria 12V recarregável	20,90	1	63,00
kit Jumper macho – macho (20)	14,39	1	26,39
kit Jumper macho – fêmea (20)	12,16	1	24,16
Clip para bateria	16,65	1	28,65
Kit de parafusos	38,92	1	52,92
kit Abraçadeira	34,34	1	47,50
kit resistor 220R (10)	12,79	1	24,79
Pushbotton	9,99	1	21,99
kit resistor de 10kΩ (10)	27,61	1	39,61
Fita dupla face	13,00	1	13,00
		Total:	507,57

FONTE: Elaborado pelos autores (2024).

Os materiais exceto o chassi, listados na Tabela 1 foram custeados pelo Professor Rogério Varavallo.

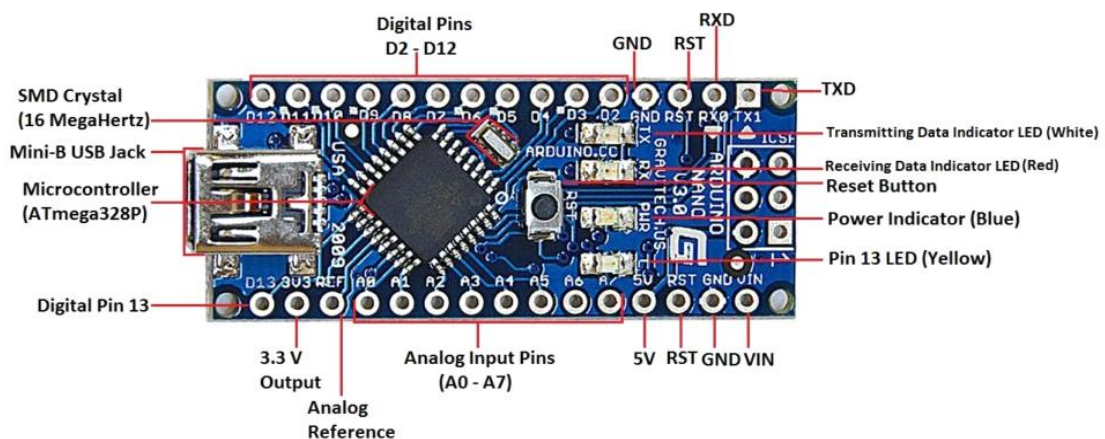
A seguir, há uma breve apresentação dos principais componentes utilizados no robô.

Arduino Nano

O Arduino Nano clássico é a menor placa da família Arduino, medindo apenas 45 mm de comprimento e 18 mm de largura, e pesando cerca de 7 gramas. Ele possui um conector USB Mini-B para programação e alimentação, e pinos de

cabeçalho que facilitam a conexão em uma placa de ensaio. A placa é baseada no microcontrolador Atmel ATmega328p nas versões mais recentes, ou ATmega168 nas versões anteriores, funcionando a uma frequência de 16 MHz. Dependendo da versão, a memória flash pode ser de 16 KB ou 32 KB (com 2 KB usados para o bootloader), 1 KB ou 2 KB de SRAM e 512 bytes ou 1 KB de EEPROM. Possui 14 pinos digitais (dos quais 6 podem ser usados como saídas PWM), 8 pinos analógicos, 2 pinos de reset e 6 pinos de alimentação (Vcc e GND). A placa opera com uma tensão de 5V, mas a tensão de entrada pode variar de 7 a 12V. Suporta comunicação I2C e SPI, permitindo a configuração de comunicação entre duas placas. Devido ao seu tamanho compacto e baixo consumo de energia, o Arduino Nano é ideal para projetos onde o espaço é limitado, como dispositivos vestíveis, sensores ambientais e projetos de automação residencial.

As especificações da placa do Arduino nano e seus componentes são apresentados na figura 2.



Arduino Nano V3.0 Pinout

Figura 2 – Imagem do Arduino com suas especificações
FONTE: Cap Sistema (2024).

Motor Shield L298N

Contém duas pontes H para controlar até dois motores DC ou um motor de passo. Foi projetado para controlar cargas indutivas como relés, solenóides, motores DC e motores de passo, permitindo o controle não só do sentido de rotação do motor, como também da sua velocidade, utilizando os pinos PWM do Arduino.

A figura 3 apresenta Módulo L298N e suas características.

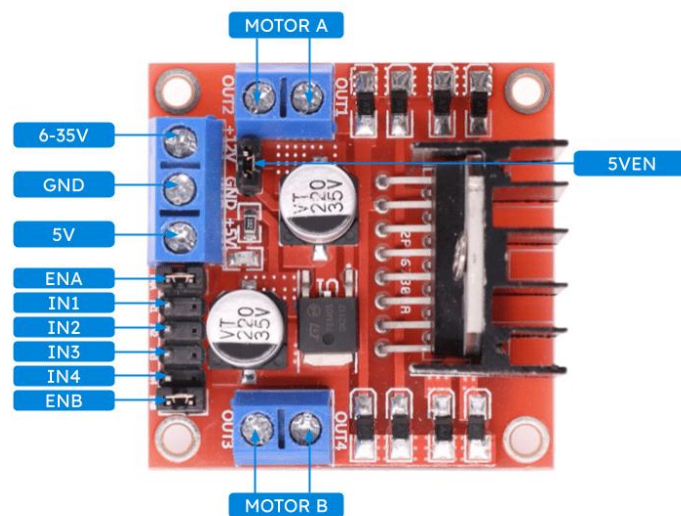


Figura 3 – Módulo L298N e suas características
FONTE: Maker Hero (2024).

- **(Motor A) e (Motor B)** – referem-se aos conectores para ligação de 2 motores DC ou 1 motor de passo.
- **(Ativa MA) e (Ativa MB)** – são os pinos responsáveis pelo controle PWM dos motores A e B. Se estiver com jumper, não haverá controle de velocidade, pois os pinos estarão ligados aos 5V. Esses pinos podem ser utilizados em conjunto com os pinos PWM do Arduino.
- **(Ativa 5V) e (5V)** – Este Driver Ponte H L298N possui um regulador de tensão integrado. Quando o driver está operando entre 6-35V, este regulador disponibiliza uma saída regulada de +5V no pino (5V) para um uso externo (com jumper), podendo alimentar por exemplo outro componente eletrônico. Portanto não alimente este pino (5V) com +5V do Arduino se estiver controlando um motor de 6-35V e jumper conectado, isto danificará a placa. O pino (5V) somente se tornará uma entrada caso esteja controlando um motor de 4-5,5V (sem jumper), assim poderá usar a saída +5V do Arduino.
- **(6-35V) e (GND)** – Aqui será conectado a fonte de alimentação externa quando o driver estiver controlando um motor que opere entre 6-35V. Por exemplo se estiver usando um motor DC 12V, basta conectar a fonte externa de 12V neste pino e (GND).
- **(Entrada)** – Este barramento é composto por IN1, IN2, IN3 e IN4. Sendo estes pinos responsáveis pela rotação do Motor A (IN1 e IN2) e Motor B (IN3 e IN4).

A tabela 2 mostra a ordem de ativação do Motor A, através dos pinos IN1 e IN2. O mesmo esquema pode ser aplicado aos pinos IN3 e IN4, que controlam o Motor B.

Tabela 2 –Ordem de ativação do motor A

MOTOR A	IN1	IN2
HORÁRIO	5V	GND
ANTI-HORÁRIO	GND	5V
PONTO MORTO	GND	GND
FREIO	5V	5V

FONTE: Maker Hero (2024).

Conectando Ponte H ao Arduino

Vamos mostrar dois esquemas de ligação deste módulo ao Arduino Nano que utilizarão o mesmo programa mostrado no final do post.

O primeiro circuito (figura 4) utiliza a alimentação do próprio Arduino, e deve ser feito sem o jumper em *Ativa 5V*. Utilizamos 2 motores DC 5V.

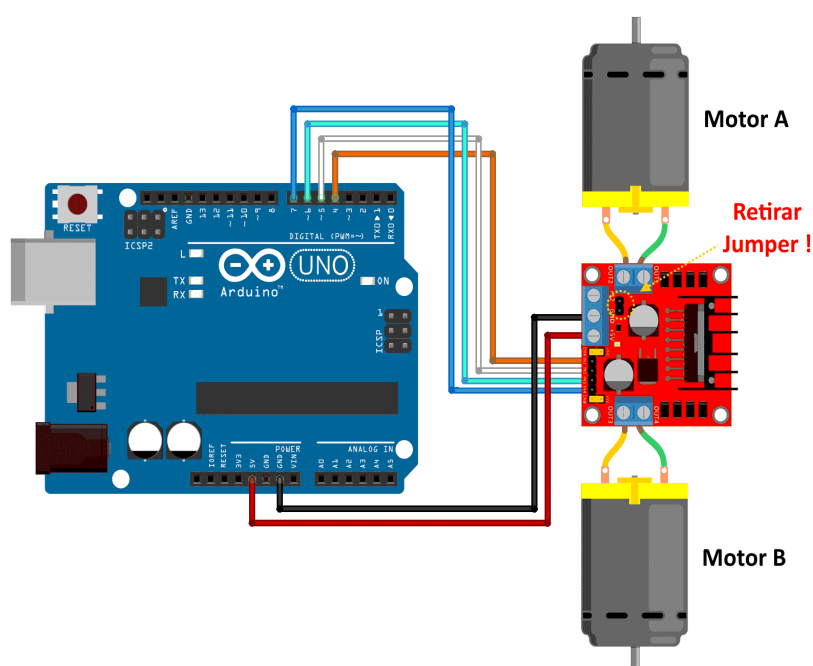


Figura 4 – Ligação entre o Arduino, motor DC e módulo L298N com jumper
FONTE: Eletronic Arduino (2024).

O segundo circuito utiliza alimentação externa e 2 motores DC de 12V. Nesse caso precisamos colocar o jumper em *Ativa 5V*, como mostra a figura 5.

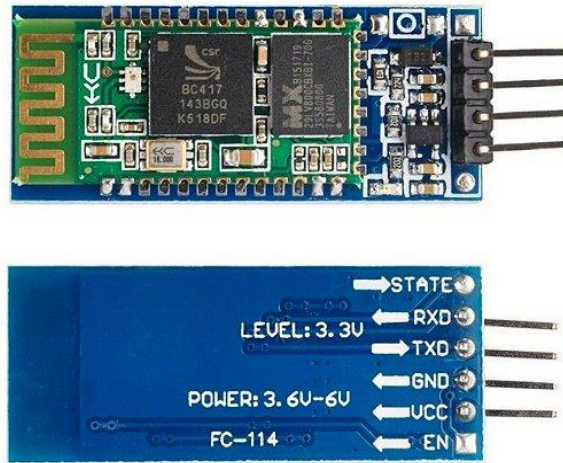


Figura 6 – Modulo Bluetooth HC-05
FONTE: Curto circuito (2024).

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2024

A tabela 3 representa o cronograma das atividades do grupo

Tabela 3 Cronograma de atividades e metas no 1º Semestre de 2024

Atividade	Fev.		Março		Abril		Maio		Junho	
	Quinzena 1ª	Quinzena 2ª	Quinzena 1ª	Quinzena 2ª	Quinzena 1ª	Quinzena 2ª	Quinzena 1ª	Quinzena 2ª	Quinzena 1ª	Quinzena 2ª
Integrantes da equipe	x									
Informações sobre a batalha de robôs		x								
Introdução ao diário de Bordo.		x								
Especificações do robô			x							
Pesquisa de preços dos materiais a serem utilizados			x							
Continuação da pesquisa dos materiais			x							
Redação do projeto boneco técnico até o capítulo 2				x	x	x	x	x		
Preparação dos slides de apresentação					x	x	x	x		
Programação do robô							x	x		
									x	x

FONTE: Elaborado pelos autores (2024).

4.2 Atividades e Metas 2º Semestre de 2024

As atividades avaliadas feitas foram apresentadas na Tabela 4.

Tabela 4: 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
Corte e furos do chassi	x	x								
Início da montagem do robô sumô			x							
Esquema Elétrico do Robô Sumô			x							
Atualização do Boneco Projeto Técnico				x						
Desenvolvimento do protótipo da cobertura robô sumo				x						
Preparação da cobertura do robô sumo					x	X				
Finalização do robô sumo						x	x	x	x	
Últimas atualizações do projeto técnico para entrega									x	x

FONTE: Elaborado pelos autores (2024).

5 DESENVOLVIMENTO DO PROJETO

5.1 Descrição do Funcionamento

A Figura 7 apresenta o esquema elétrico de toda a parte eletrônica do projeto.

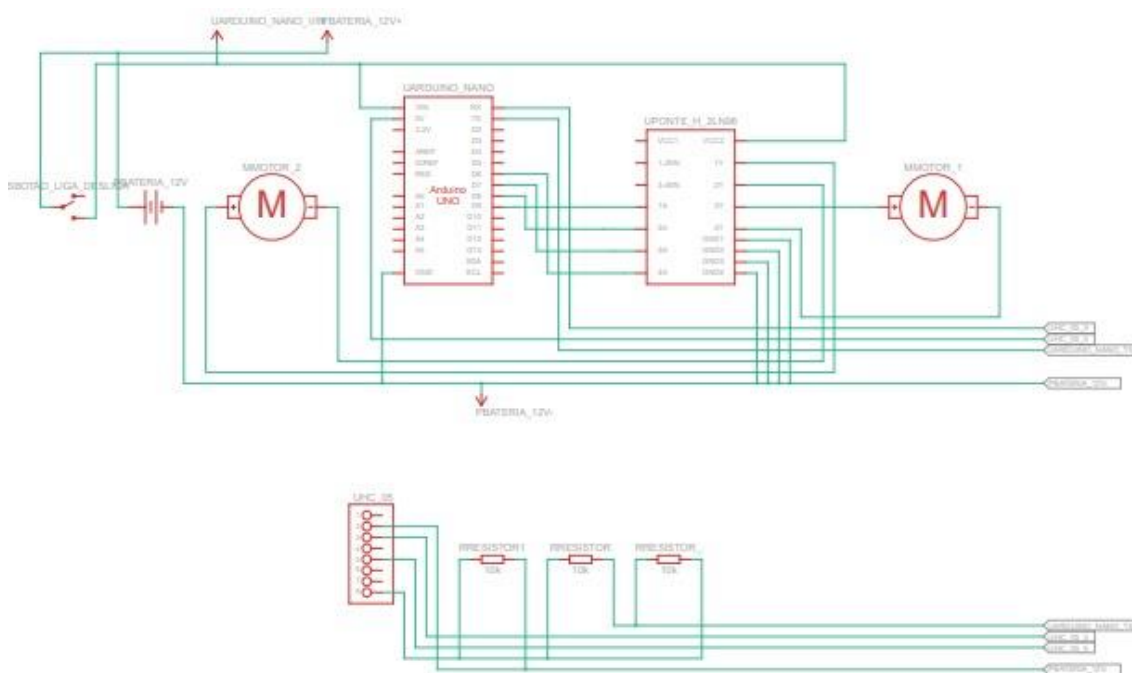


Figura 7 – Esquema elétrico do robô
FONTE: Elaborado pelos autores (2024).

O esquema elétrico da parte eletrônica é descrito a seguir.

O robô é alimentado por uma bateria recarregável 12V com a corrente de 4,4 AH, que é passado o fio positivo para um botão N/O liga-desliga para controlar a passagem de energia ao carrinho, e o fio negativo que é passado diretamente para o GND do Arduino, ao Módulo L298N e ao Módulo bluetooth. A partir do botão, a passagem do positivo é feita por um fio para a alimentação do Arduino, Módulo bluetooth e Ponte-H L298N. O GND do Arduino é interligado com o do Módulo L298N. O RX do módulo bluetooth é conectado a 3 resistores em associação mista que, a partir deles é efetuada a conexão ao TX do Arduino, e o TX do módulo bluetooth é conectado ao RX do Arduino. Posteriormente, o Módulo L 298N é interligado ao Arduino nos seus 4 pinos nas saídas PWN.

As entradas identificadas por IN do Módulo L 298N são conectadas do seguinte modo: IN1 conectado ao pino 6 do Arduino; IN2 conectado ao pino 7 do

Arduino; IN3 conectado ao pino 8 do Arduino; e IN4 conectado ao pino 9 do Arduino.

E para finalizar a conexão dos 2 motores são feitas nos diferentes lugares destinados ao motor para o Módulo L 298N que direcionará o sentido de rotação, velocidade e frenagem do carrinho.

A descrição do funcionamento é apresentada a seguir.

O robô tem como componente principal o Arduino Nano, que, uma vez programado, aciona os motores através do módulo L298N conforme os comandos enviados por um celular via aplicativo Arduino Car e recebidos e enviados ao Nano através do módulo bluetooth HC06. A figura 14 apresenta o fluxograma do programa gravado no Arduino Nano.

5.2 Montagem Física do Projeto

As Figuras 8 a 11 apresentam a sequência da montagem física do nosso robô

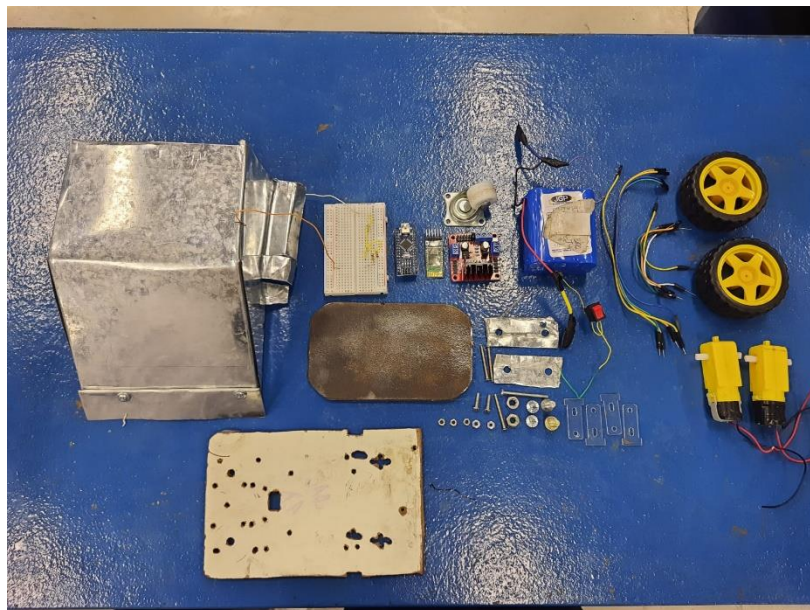


Figura 8 – Peças utilizadas no robô
FONTE: Elaborado pelos autores (2024).

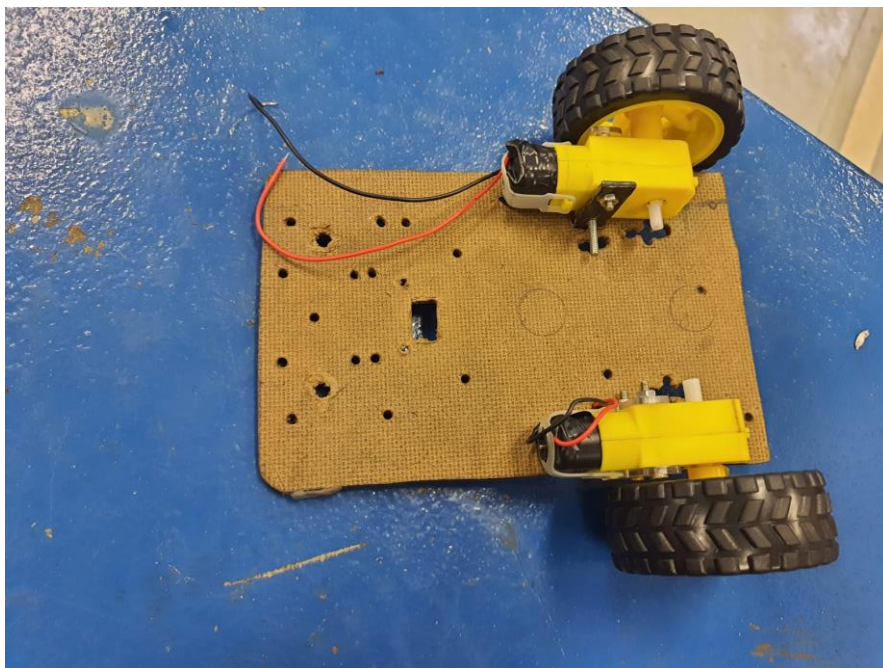


Figura 9 – Instalação das rodas e motores no chassi
FONTE: Elaborado pelos autores (2024).

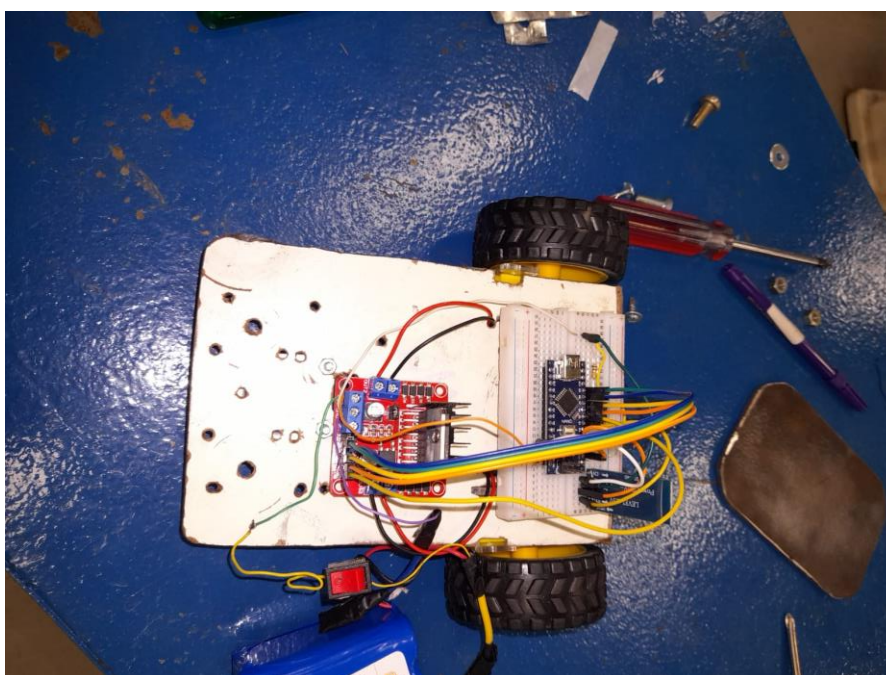


Figura 10 – Instalação do Arduino e módulos no chassi
FONTE: Elaborado pelos autores (2024).

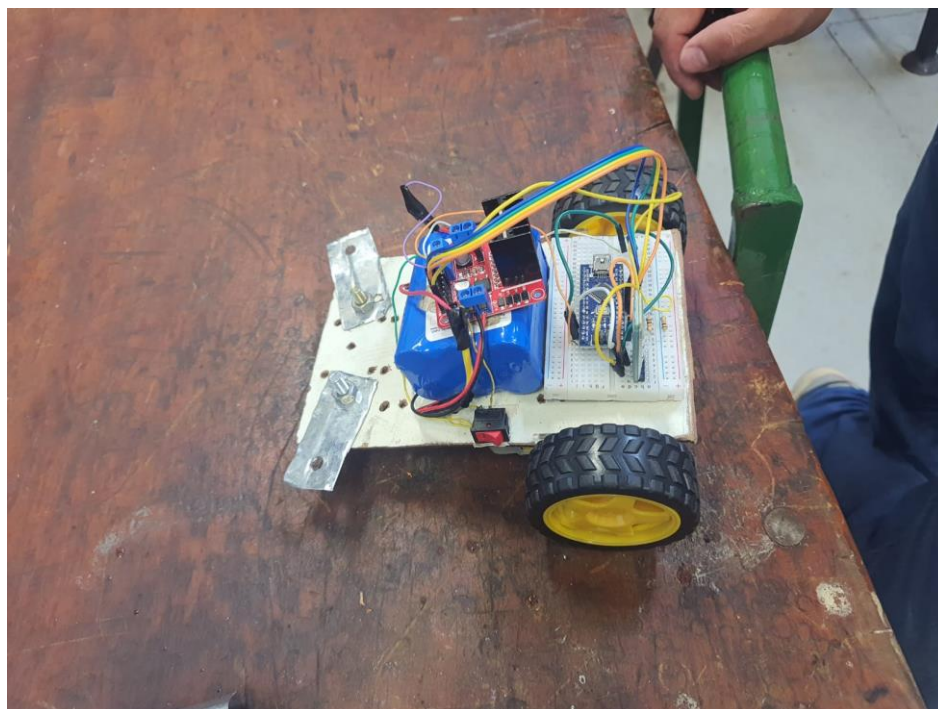


Figura 11 – Instalação da bateria no chassi
FONTE: Elaborado pelos autores (2024).

As Figuras 12 e 13 apresentam os testes finais efetuados em nosso projeto.

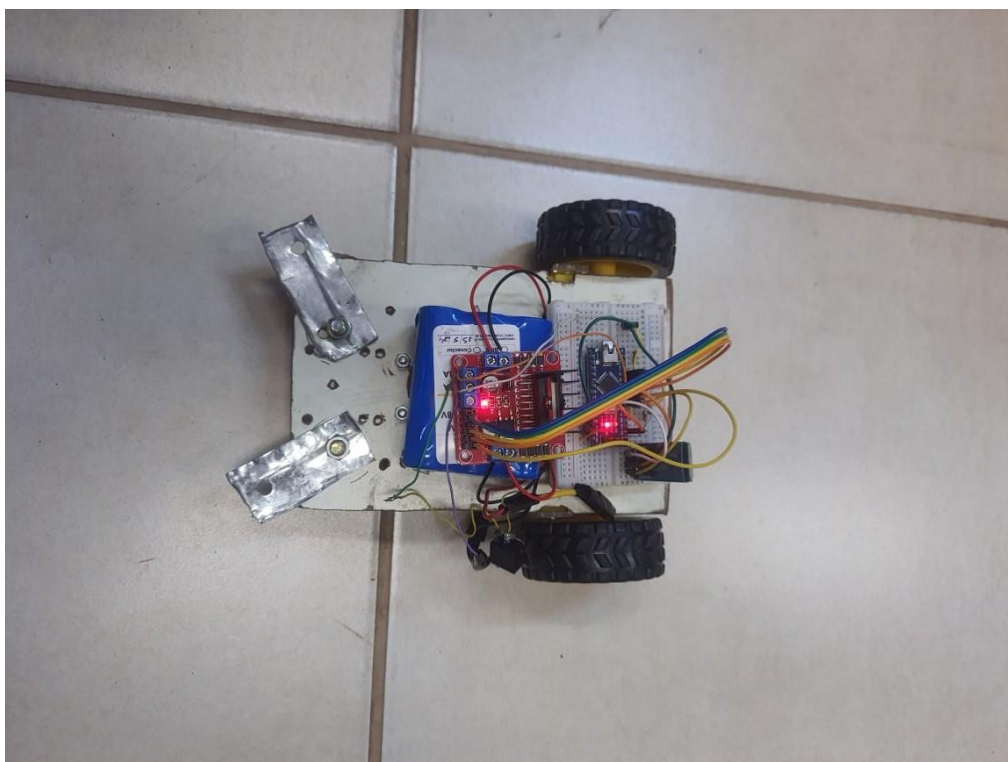


Figura 12 – Teste do hardware
FONTE: Elaborado pelos autores (2024).

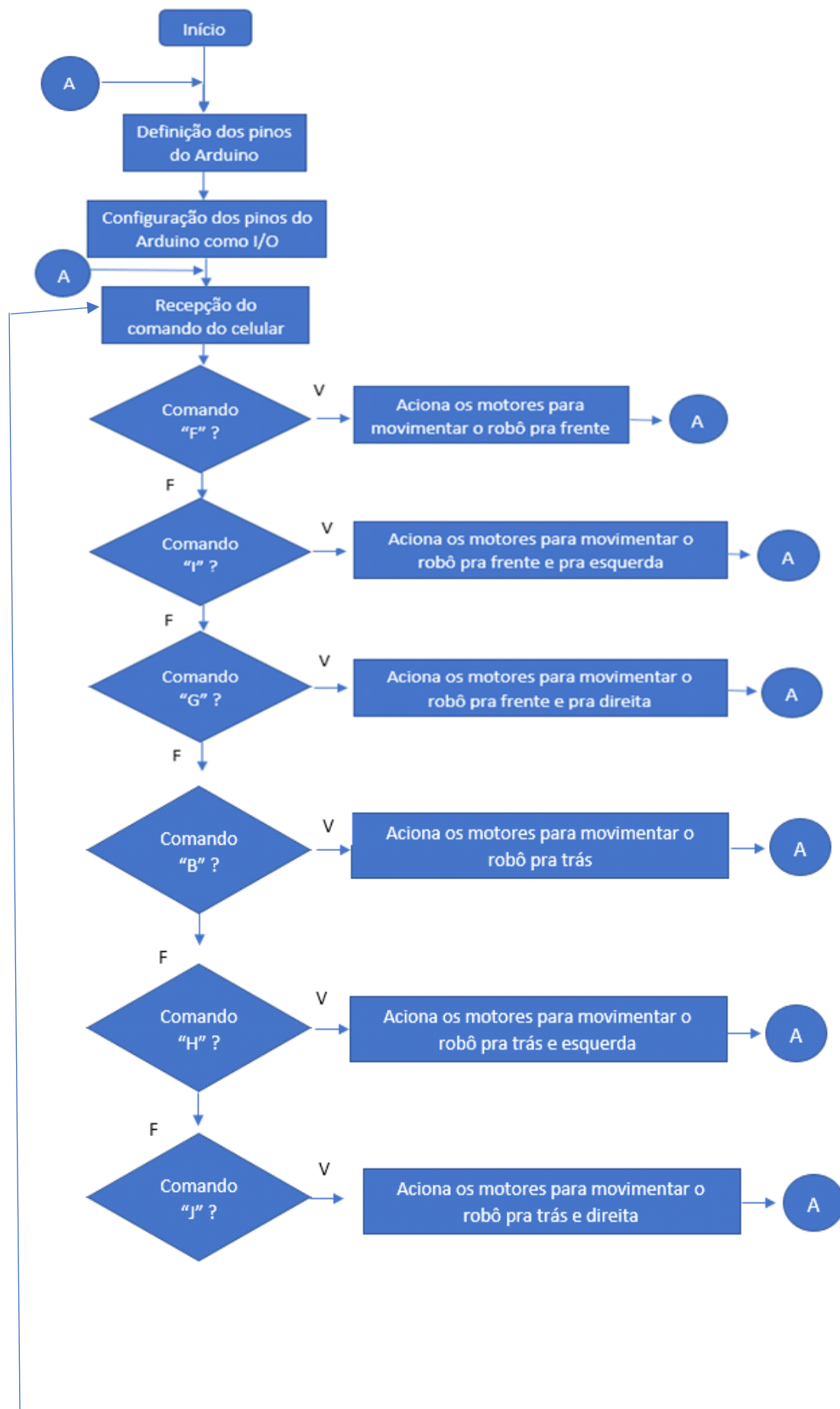


Figura 13 – Teste geral do robô
FONTE: Elaborado pelos autores (2024).

5.3 Desenvolvimento do programa do robô

A programação do robô foi desenvolvida no aplicativo Arduino IDE versão 1.8.5.

A Figura 14 apresenta o fluxograma da programação do robô.



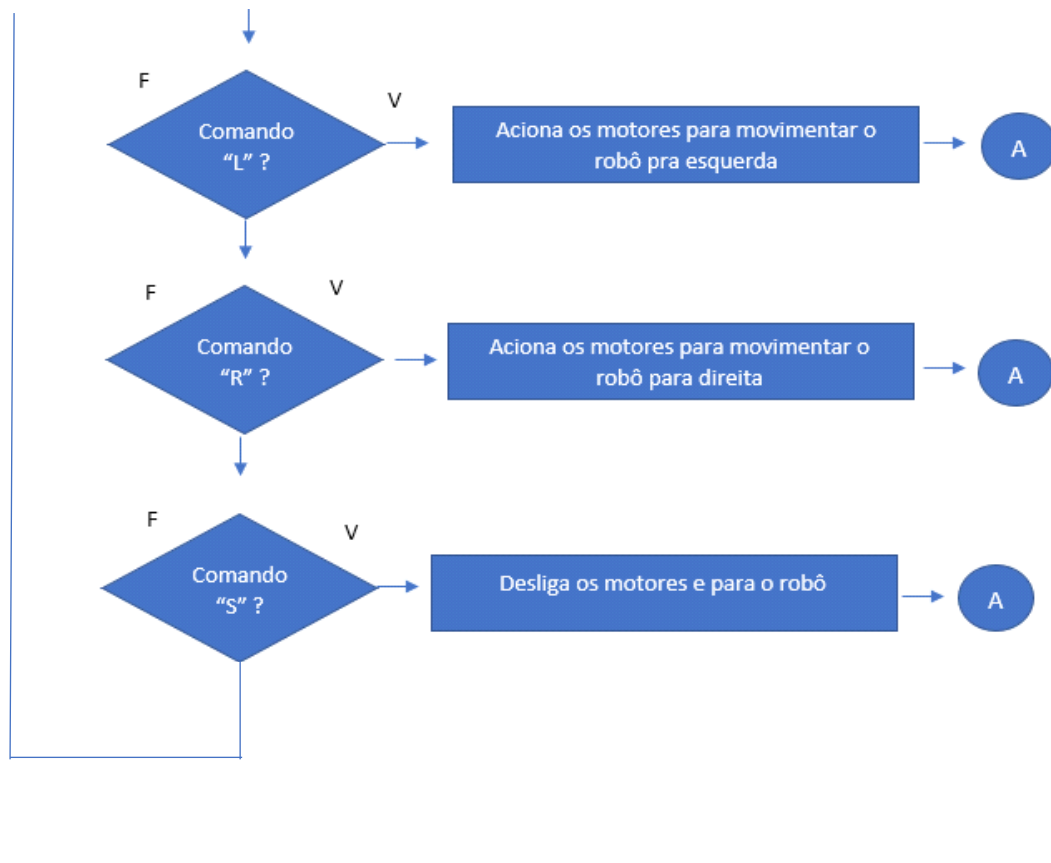


Figura 14 – Fluxograma do programa do robô
 FONTE: Elaborado pelos autores (2024).

Conforme apresentado no fluxograma, o funcionamento dele começa após fazer o upload do programa no Arduino, fazer as definições dos pinos do Arduino, depois fazer as configurações dos pinos no Arduino, fazer a recepção do comando no celular.

- Quando o comando for “F” aciona os motores para o robô ir pra frente.
- Quando o comando for “I” aciona os motores para o robô ir para frente e para esquerda.
- Quando o comando for “G” aciona os motores para o robô ir para frente e para direita.
- Quando o comando for “B” aciona os motores para o robô ir para trás.
- Quando o comando for “H” aciona os motores para o robô ir para trás e esquerda.
- Quando o comando for “J” aciona os motores para o robô ir para trás e direita.

- Quando o comando for “L” aciona os motores para o robô ir para esquerda.
- Quando o comando for “R” aciona os motores para o robô ir para direita.
- Quando o comando for “S” desligará os motores.

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

6 CONSIDERAÇÕES FINAIS

Esperamos ter concluído nosso projeto com excelência, atendendo plenamente às expectativas de nossos professores e orientador. Este trabalho reflete o conhecimento e as habilidades que adquirimos ao longo do curso, demonstrando nosso comprometimento e dedicação. Acreditamos que o resultado final está à altura dos padrões acadêmicos exigidos e que conseguimos integrar de maneira eficaz a teoria e a prática. Nosso objetivo foi não apenas cumprir os requisitos do projeto, mas também superar as expectativas, proporcionando uma contribuição significativa para a área de estudo. Estamos confiantes de que nosso esforço e empenho serão reconhecidos e valorizados.

REFERÊNCIAS BIBLIOGRÁFICAS:

Cap Sistema. Guia do Arduino Nano – Pinagem e Esquemas. Capsistema, 06/01/2021. Disponível em: <https://capsistema.com.br/index.php/2021/01/06/guia-do-arduino-nano-pinagem-e-esquemas/>. Acesso em: 26/04/2024.

Maker Hero. Motor DC, 14/03/2013. Disponível em: <https://www.makerhero.com/blog/motor-dc-arduino-ponte-h-l298n/>. Acesso em: 19/04/2024.

Wikipédia. Pont H, 02/10/2022. Disponível em: https://pt.wikipedia.org/wiki/Ponte_H. Acesso em: 26/04/2024.

Curto circuito. Módulo Bluetooth Slave - HC05, 2016-2022. Disponível em: <https://curtocircuito.com.br/modulo-bluetooth-master-hc05.html>. Acesso em: 26/04/2024.

Eletronic Arduino. Conectando Pont H ao Arduino, 17/06/2016. Disponível em: <https://eletronicarduino.wordpress.com/2016/06/17/motor-dc-com-driver-ponte-h-l298n/> Acesso em: 26/04/2024

APÊNDICE

Programa do Robô

// CARRO CONTROLADO POR BLUETOOTH

```
// === Pinos utilizados no driver L298 === //
#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

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

void setup() {
    // === Definição dos pinos de 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(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" 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
        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
        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
        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
    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
    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
    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
    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
    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
}
}

```