

**CENTRO ESTADUAL DE EDUCAÇÃO
TECNOLÓGICA PAULA SOUZA
ETEC SYLVIO DE MATTOS CARVALHO**

Curso Técnico em Mecatrônica M-tec

**JOSE ANTONIO BOIARO CAXA
MARIA CLARA RODRIGUÊS BELO
MARIA LUIZA LIMA CORRÊA
MIRIÃ YARA CUSTÓDIO PEREIRA
RAYSSA CRISTINA NEVES**

**DESENVOLVIMENTO DE ROBÔ SUMÔ E ANÁLISE DE
ESTRATÉGIAS DE COMBATE**

Matão, SP

2024

SUMÁRIO

1	32
	43
	43.1
	53.2
	64
104.1 Atividades e Metas 1º Semestre de 2024	11
4.2 Atividades e Metas 2º Semestre de 2024	12
5. DESENVOLVIMENTO DO PROJETO	13
5.1	135.2
	155.3
	186
217	Referencia Bibliográfica
	22
8	Apêndice
	23 a 28

1 INTRODUÇÃO

Este documento apresenta o Projeto Técnico “Aplicação de técnicas avançadas em robótica para o aprimoramento de desempenho em competições de 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 Mecnatrônica, contribuindo para sua formação e preparando-os para novas jornadas. O objetivo desse projeto é analisar e desenvolver um robô sumô com sucesso, utilizando abordagens de atividade escolares dos professores com o desafio de remover ou empurrar o adversário para fora da arena, buscando estratégias eficazes, engajando a criatividade e inovação do aluno ao trabalho em equipe. 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 nos 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

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 quatro rodas de borracha e dois 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 01 apresenta uma foto do nosso projeto final montado.

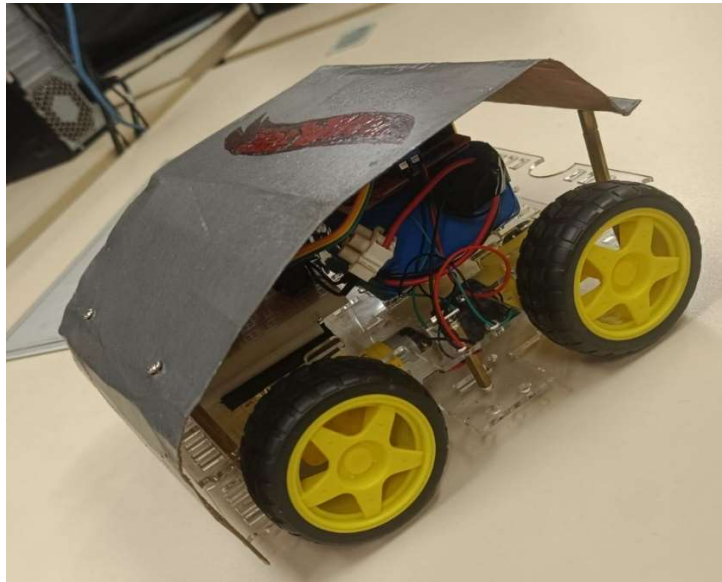


Figura 01 – Foto do projeto montado

FONTE: Elaborado pelos autores (2024).

3.2 Materiais Utilizados e suas Especificações Técnicas

Os materiais utilizados em nosso projeto estão apresentados na Tabela 1. Esses materiais, desde os componentes eletrônicos até os demais itens, foram escolhidos por sua praticidade e facilidade de uso, resistência e durabilidade, além, é claro, de seu custo e versatilidade.

Tabela 1 – Lista de Materiais

Material	Quantidade
Arduino Nano	1
Driver L298N	1
Módulo HC-05	1
Protoboard	1
Chassi acrílico	2
Chapa galvanizada	1
Interruptor ON/OFF	1
Jumpers	25
Motor CC/Redutor	4
Bateria 12V	1

FONTE: Elaborado pelos autores (2024).

3.2.1 - Arduino nano

O Arduino Nano como observado na figura 1 é uma placa de prototipagem que inclui um hardware (placas que contêm microcontroladores e pinos de entrada e saída, digital e analógico) e um software IDE (um software que permite a programação da placa Arduino). Sua função é ser um componente versátil que permita a criação de vários tipos de protótipos e sistemas autônomos, ele é microcontrolada pequena e de fácil uso baseada no microcontrolador Atmel, ZATmega328p que funciona numa frequência de 16 Mhz, tem uma tensão de alimentação de 5v, 14 pinos de entradas/saídas digitais, 8 entradas analógicas pinos que utiliza um miniUSB padrão para programação.

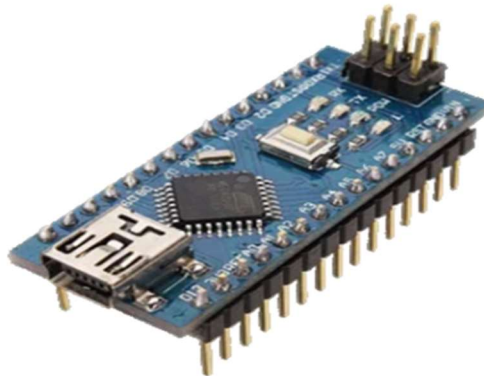


Figura 2: Arduino Nano

FONTE: <https://www.blogdarobotica.com/2021/08/10/programando-o-arduino-nano-com-old-bootloader/> (2024).

3.2.2 - Driver L298N

A ponte H como observado na figura 2 é uma placa em formato de “H” que permite a conexão entre o Arduino e outros componentes, além da função de reverter a polaridade de uma carga sem a necessidade como motores DC, utilizando os pinos PWM do Arduino. A ponte H é um circuito de eletrônica de potência do tipo *chopper* de classe E (um chopper de classe E converte uma fonte fixa de corrente contínua fixa em uma tensão de corrente contínua variável abrindo e fechando diversas vezes), e , portanto, pode determinar o sentido da corrente, a polaridade da tensão e a tensão em um dado sistema ou componente. (Wikipédia 2022)

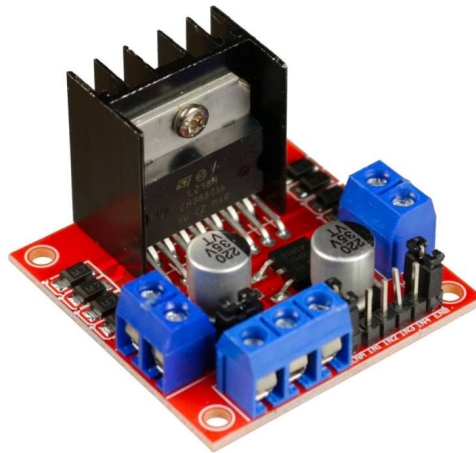


Figura 3: Driver L298N

Fonte: <https://www.blogdarobotica.com/2023/05/15/como-controlar-motor-dc-utilizando-o-driver-ponte-h-l298n/> (2024).

3.2.3 Bateria 12V

A bateria apresentada na figura 3 é uma fonte de alimentação interna que possui capacidade de 12V de tensão elétrica, além de uma duração de 4400 mah (miliampere/hora). Este modelo é Li-ion 18650.



Figura 4: Bateria 12V

FONTE: Autoria própria (2024).

3.2.4 - Módulo HC-05

O “HC-05” como o da figura 4 é um módulo que permite a conexão entre dispositivos bluetooth, como controles ou até mesmo o celular. Ele conecta a dispositivos através de conexão sem fio bluetooth, usado como microcontroladores por facilitar a troca de informação entre dispositivos em uma distância de 10 metros, pode ser configurado de forma mestre ou escravo, recebendo ou enviando informação. Possui uma alimentação de 3,6v a 6v com uma frequência de 2,4 GHz, com um consumo de 30 mA em modo de espera, 10va em modo de espera.

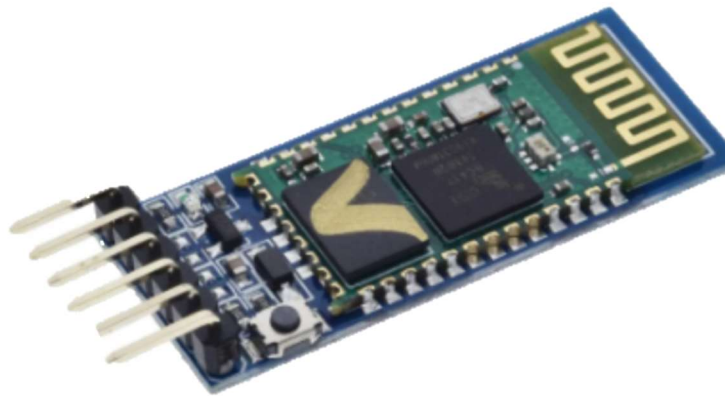


Figura 5: Ponte HC-05

Fonte: <https://www.blogdarobotica.com/2023/02/13/como-utilizar-o-modulo-bluetooth-hc-05-com-arduino/> (2024).

3.2.5 - Motor CC/Redutor

O motor CC/ Redutor como visto abaixo, é um tipo de motor que funciona com corrente contínua, que serve para diminuir a velocidade de rotação do eixo do motor enquanto o torque é aumentado. Seu funcionamento converte a energia elétrica para energia mecânica.



Figura 6: Motor CC

Fonte: [Como controlar um motor usando o Arduino - Blog da Robótica](#) (2024).

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2024

No primeiro semestre, nosso grupo planejou iniciar as montagens da estrutura do robô e organizar os componentes que nele serão inseridos. Também utilizaremos esse período para realizar a programação do nosso robô.

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

atividades	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 Qui nze na	2ª Qu inz en a	1ª Qui nze na	2ª Qui nze na
Férias escolar	x									
Escolha da equipe		x								
Escolha do tema padrão		x								
Pesquisa sobre o tema			x	x	x	x	x	x	x	x
Pesquisa sobre os materiais					x	x	x	x	x	x
Desenvolvimento do diário de bordo			x	x	x	x	x	x	x	x
Desenvolvimento do boneco					x	x	x	x	x	x
Escolha do nome do nosso protótipo					x					
Preparação para prévia/ revisão dos documentos						x	x	x		
Prévia da banca								x		
Correção sugerido pela banca									x	
Avaliação do professor orientador										x

FONTE: Elaborado pelos autores (2024).

4.2 - Atividades e Metas 2º Semestre de 2024

No segundo semestre, planejamos realizar testes com o protótipo do robô, e de acordo com os resultados desses testes, modificar, consertar e melhorar diversos elementos do robô, que podem ser desde sua movimentação e comunicação até sua resistência e seus componentes.

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

atividades	Julho		Agosto		Setembro		Outubro		Novembro	
	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 Qui nze na	2ª Qui nze na	1ª Qui nze na	2ª Qui nze na
Férias escolar	x	x								
Desenvolvimento do fluxograma							x	x	x	
Compra dos matérias				x	x	X	X			
Desenvolvimento do esquema elétrico			x							
Desenvolvimento do diário de bordo			x	x	x	x	x	x	x	x
Desenvolvimento do boneco			x	x	x	x	x	x	x	x
Desenvolvimento da programação								x		
Correção dos erros encontrados							x	x		
Montagem física do robô sumô							x	x	x	
Testes do protótipo e treino de estratégias de combate								x	x	x
Apresentação do trabalho de conclusão de curso										x
Batalha robô sumô										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.

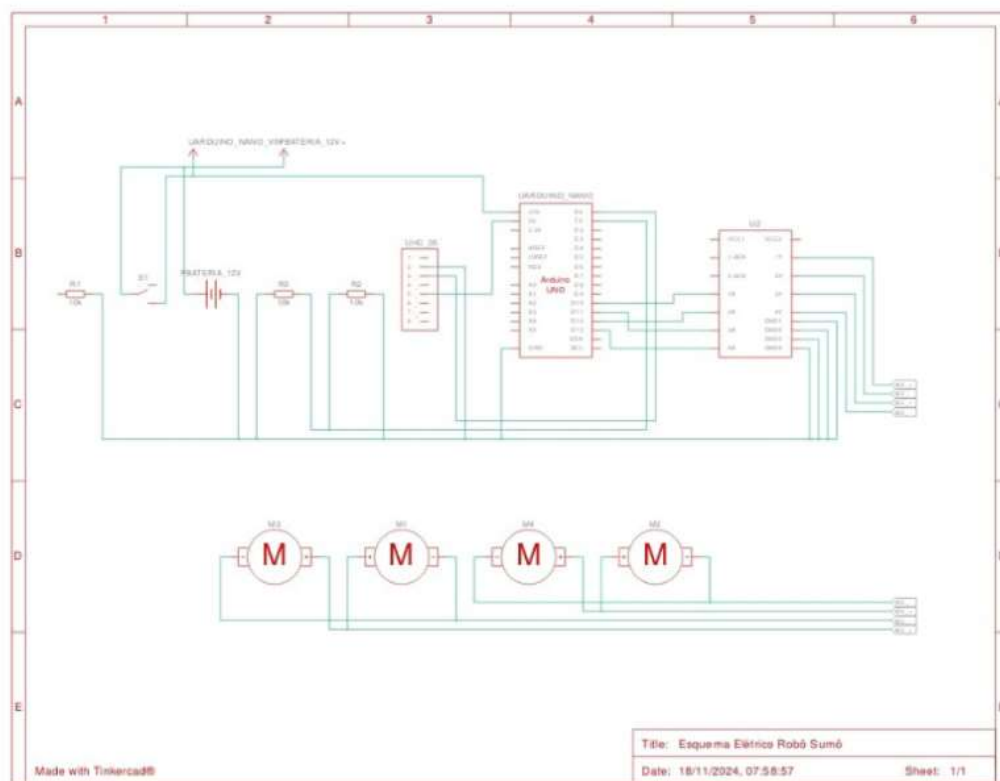


Figura 7 – Esquema elétrico do robô
FONTE: Elaborado pelos autores (2024).

O funcionamento da parte eletrônica é descrito a seguir:

. Este diagrama de circuito mostra um sistema robótico controlado por um Arduino Nano.

Vejamos os componentes principais e sua interconexão:

Arduino Uno: Atua como unidade de controle central, recebendo entradas de sensores e processando as informações para controlar os motores.

L293D: Este chip é essenciais porque amplifica a corrente do sinal do Arduino, fornecendo a potência necessária para operar os motores. O L293D controla os quatro motores, dois motores de cada lado do próprio.

Motores DC (quatro): São motores que podem ter até uma alta potência, Eles são usados em diversas aplicações, como eletrodomésticos, veículos elétricos, brinquedos e ferramentas elétricas, onde é necessário controlar a velocidade e o torque do motor.

Módulo Bluetooth HC05: Um módulo Wi-Fi que permite que o robô se comunique sem fio com um dispositivo externo, que será nosso smartphone. Isso permite controle do robô.

Protoboard: É uma placa de plástico com furos condutores organizados em linhas horizontais e colunas que também permite a montagem de circuitos sem a necessidade de solda.

Funcionamento: O Arduino envia sinais PWM (Pulse Width Modulation) para os pinos digitais dos L293D. Esses sinais PWM controlam a velocidade e a direção dos motores. O módulo Wi-Fi permite que um dispositivo externo envie comandos para o Arduino, alterando os sinais PWM e, conseqüentemente, o movimento do robô.

Em resumo: Este é um circuito básico de um robô com quatro motores DC, controlado por um Arduino Uno e com capacidade de controle remoto via Wi-Fi. A utilização de um protoboard indica que se trata de um projeto de prototipagem.

5.2 Montagem Física do Projeto

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



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)

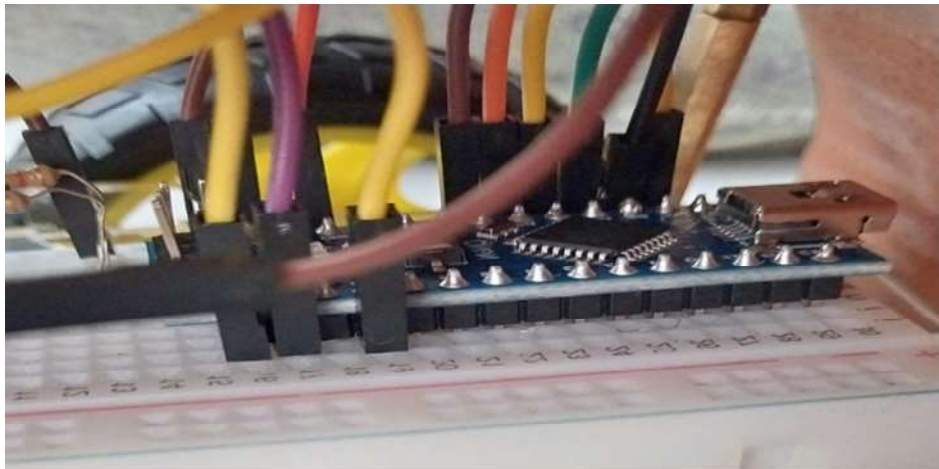


Figura 10 – Instalação do Arduino e módulos no chassi

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.

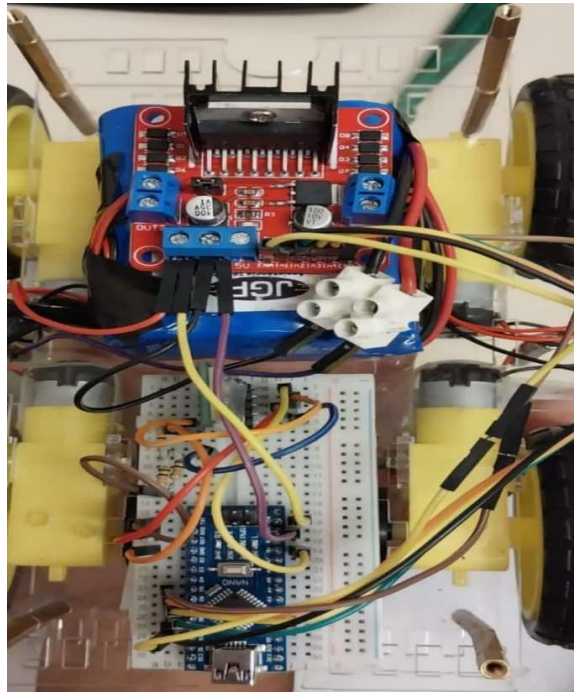


Figura 12 – Teste do hardware
FONTE: Elaborado pelos autores (2024).

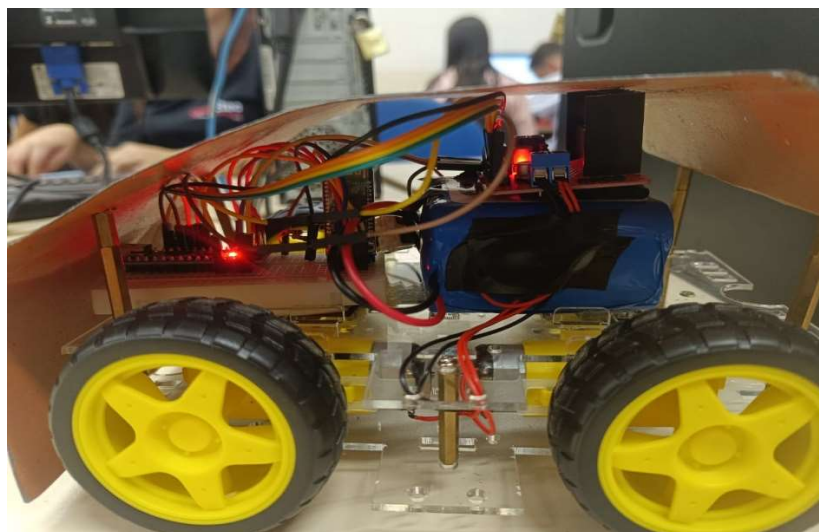


Figura 13 – Teste geral do robô
FONTE: Elaborado pelos autores (2024).

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 do programa.



Figura 14 – Fluxograma do programa do robô

FONTE: Elaborado pelos autores (2024).

O sistema inicia com a execução do programa que controlará o robô.

Definição dos pinos utilizados:

Configura os pinos do microcontrolador que serão empregados no controle do robô, incluindo aqueles conectados aos motores e sensores.

Definição dos pinos como entrada/saída:

Define a função de cada pino configurado, especificando quais atuarão como entradas (para sensores) e saídas (para controle de motores).

Programação em Loop:

A partir desse ponto, o programa entra em um laço contínuo, executando as ações necessárias para o movimento e o controle do robô de acordo com os estímulos recebidos.

Movimento do robô:

Baseando-se nos sinais recebidos dos sensores, diferentes comandos de movimento são enviados aos motores do robô, conforme a lógica abaixo:

- **H:** Movimento **para trás e esquerda**.
- **B:** Movimento **para trás**.
- **G:** Movimento **para frente e direita**.
- **I:** Movimento **para frente e esquerda**.
- **F:** Movimento **para frente**.
- **J:** Movimento **para trás e direita**.
- **L:** Movimento **para esquerda**.
- **R:** Movimento **para direita**.
- **S:** Estado **parado** (sem movimentação dos motores).

Fim:

Finaliza a execução do programa, interrompendo os comandos enviados aos motores do robô.

Aplicação no Robô Sumô

Este fluxograma reflete o comportamento esperado em uma competição de robôs sumô. A lógica de movimentação é baseada nos sinais dos sensores, como os que detectam a borda da arena ou a posição do adversário, priorizando estratégias para empurrar o oponente para fora da arena e evitar a saída do próprio robô.

6 CONSIDERAÇÕES FINAIS

Realizamos o projeto com o objetivo de demonstrar os nossos conhecimentos com de correr do ensino médio integrado com curso técnico em mecatrônica, o nosso trabalho de conclusão de curso robô sumô, durante este período, pudemos trabalhar nossas habilidades de robótica, mecânica, elétrica e programação. Desenvolvemos experiências de trabalhos práticos, trabalhos em grupo, e raciocínio lógico, tivemos oportunidade de conhecer diversos tipos de materiais e componentes, além de ser algo que futuramente podemos colocar em prática na área da indústria e da nossa vida acadêmica.

REFERÊNCIAS BIBLIOGRÁFICAS

VIANA, CAROL CORREIA, CHRISTIAN, RAPHAEL. Blog da Robótica. 2014. colegas de faculdade com propósito de disseminar conhecimento e discutir sobre os projetos realizados pelo Grupo de Estudo em Robótica e Automação e pelo Grupo de Iniciação Científica. – Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), Olinda, 2023.

WALES, JIMMY, SANGER LARRY. Wikipédia. 2001. uma enciclopédia online colaborativa e livre, cujo objetivo principal é fornecer informações acessíveis e confiáveis sobre uma vasta gama de tópicos. – Estados Unidos, 2024/2023.

APÊNDICE

```
// === 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/comunicação serial
```

```
byte velocidade = 90; //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" 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

    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
```