**UNIVERSIDADE DO VALE DO PARAÍBA - UNIVAP**

**FACULDADE DE ENGENHARIAS ARQUITETURA E URBANISMO**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

MANUAL DE FABRICAÇÃO PARA ROBO SUMO

Thiago Bottrel Vieira Gutierrez

Gabriel Duarte Batista de Nazaré

Guilherme Palmanhani Valli

SÃO JOSÉ DOS CAMPOS - 2023

**SUMÁRIO**

1- INTRODUÇÃO.........................................................................3

2- OBJETIVOS.............................................................................6

3- JUSTIFICATIVA.......................................................................8

4- METODOLOGIA......................................................................10

6- CRONOGRAMA.......................................................................21

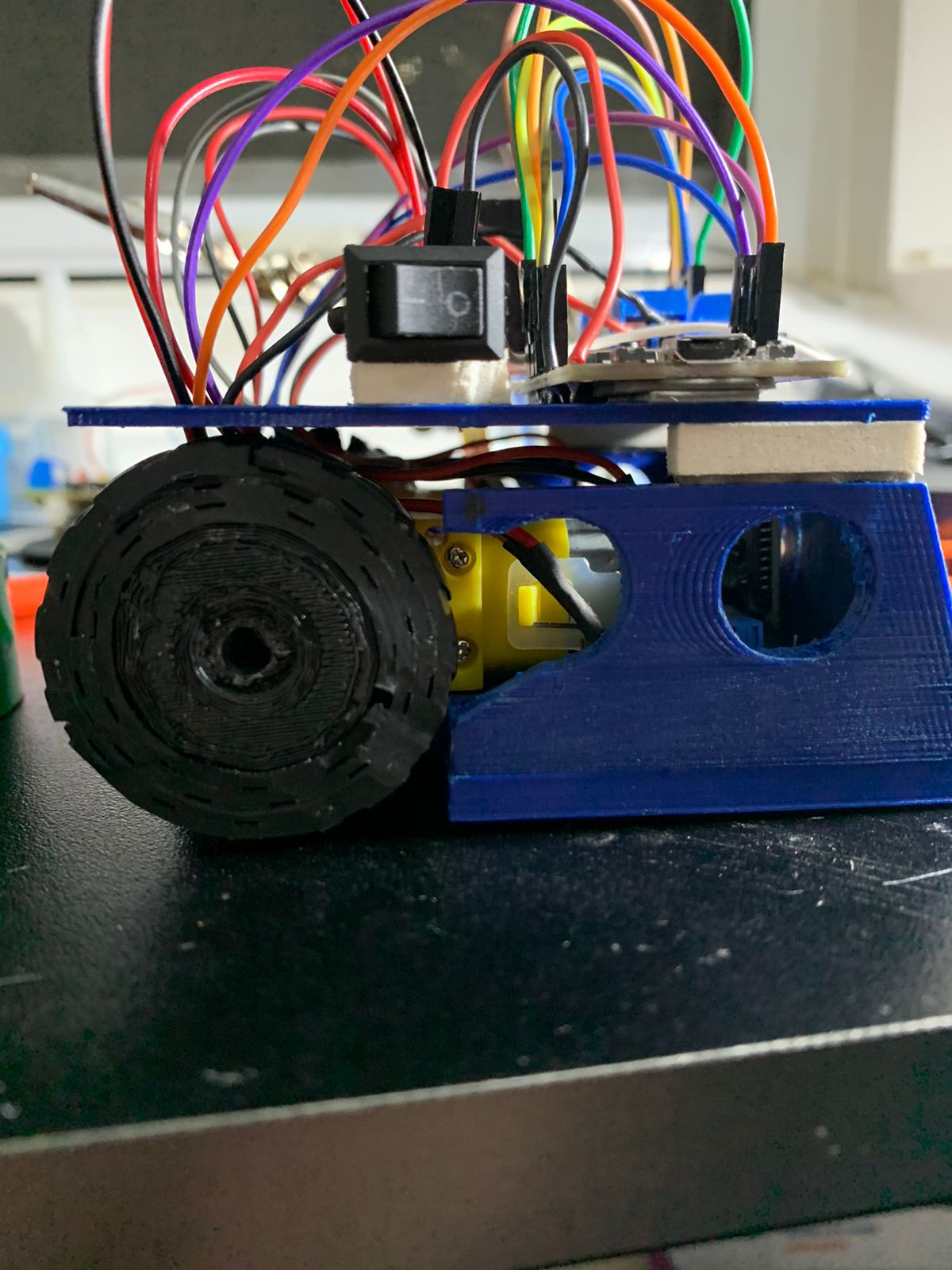
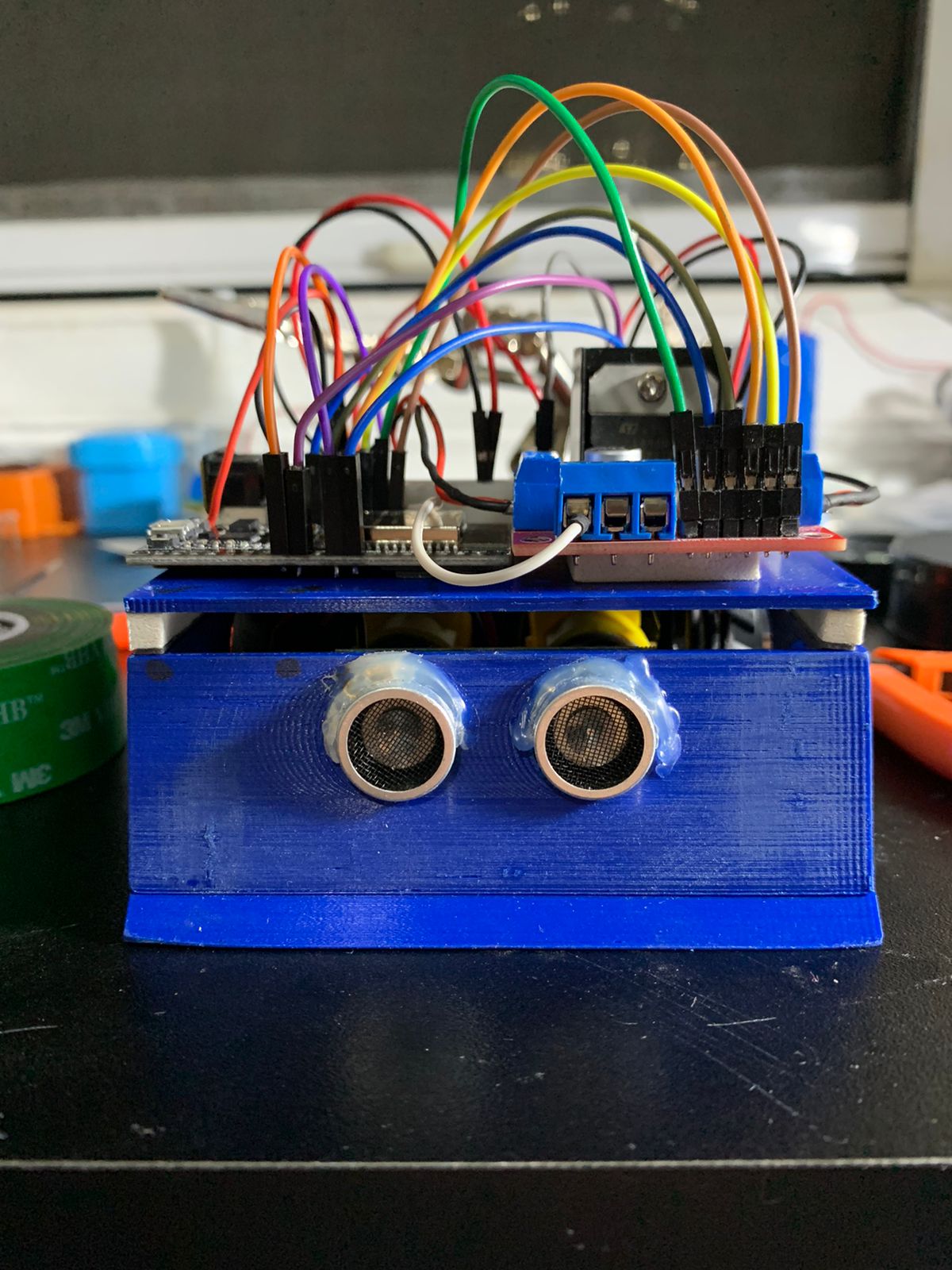
7- BIBLIOGRAFIA.........................................................................22

3

**1-INTRODUÇÃO**

A robótica, fascinante disciplina que amalgama eletrônica, programação e engenharia mecânica, desempenha um papel crucial na automatização e superação de desafios complexos em diversas áreas. Este manual visa adentrar um aspecto cativante da robótica: a concepção e fabricação de um Robô Sumô.

O cerne deste projeto reside na criação de um robô capaz de enfrentar desafios de combate sumô, um ambiente controlado onde estratégia, força e agilidade são postas à prova. Este desafio, embora à primeira vista simples, exige a integração harmoniosa de tecnologias e conhecimentos em eletrônica, mecânica e programação, resultando em um sistema autônomo de batalha.



FONTE: Autoria

4

No decorrer deste manual, dedicaremos nossa atenção à construção, configuração e programação de um Robô Sumô, destacando a aplicação de componentes como o ESP32, Ponte H, motores DC, sensores infravermelhos e uma bateria de 9V. Nosso foco será na edificação deste robusto autômato e na programação essencial para sua performance eficaz no ringue de combate sumô.

A robótica, como campo em constante evolução, abrange aplicações que transcendem desde a automação industrial até a exploração espacial. A criação

de robôs sumôs se insere de maneira singular nesse panorama, desempenhando um papel especial no currículo de Engenharia da Computação, onde estudantes têm a oportunidade ímpar de aplicar seus conhecimentos na empolgante competição de conceber e programar robôs sumôs.

A escolha de focar no desenvolvimento de um Robô Sumô como objeto deste manual foi motivada por razões multifacetadas. Em primeiro lugar, reconhecemos o valor educacional insubstituível dos projetos práticos de robótica, permitindo aos alunos a aplicação de conceitos teóricos em um cenário real. Além disso, a capacidade de criar um robô autônomo apto ao combate sumô não só tem aplicações práticas, mas também serve como uma excelente introdução ao controle de dispositivos e à automação.

5

Este manual explorará minuciosamente a construção e programação de um Robô Sumô, utilizando os componentes mencionados anteriormente para criar um autômato capaz de enfrentar desafios de combate sumô com destreza e autonomia. O enfoque estará na construção do hardware e na configuração do software para alcançar esse objetivo.

Com esta introdução, estabelecemos a base para a exploração do projeto do Robô Sumô neste manual, incluindo a importância da competição entre os alunos do curso de Engenharia da Computação

6

**2- OBJETIVOS**

Objetivo Geral:

Desenvolver um Robô Sumô funcional, capaz de competir eficazmente em um ambiente de combate, demonstrando estratégia, força e agilidade de forma autônoma.

Objetivos Específicos:

Identificar Componentes:

Listar e detalhar os materiais e componentes essenciais necessários para a construção do robô sumô, destacando o papel crucial de elementos como ESP32, Ponte H, motores DC, sensores infravermelhos e a bateria de 9V.

Montagem do Robô: Efetuar a montagem do chassi do robô, garantindo a correta fixação das rodas e motores DC, essenciais para a mobilidade e estabilidade do autômato no ringue de combate sumô.

Configuração Eletrônica:

Realizar a integração dos componentes eletrônicos, conectando o ESP32, a Ponte H e o sensor infravermelho de maneira eficiente para assegurar o funcionamento adequado e coordenado do robô.

7

Programação do Robô:

Desenvolver um código-fonte inteligente e eficaz que capacite o robô a realizar movimentos autônomos estratégicos, respondendo de forma rápida e precisa aos desafios do combate sumô.

Testes e Ajustes:

Submeter o robô a uma série de testes rigorosos, avaliando sua capacidade de seguir a linha e realizar movimentos estratégicos no ambiente controlado do combate sumô. Ajustar parâmetros conforme necessário para otimizar o desempenho.

Solucionar Problemas:

Identificar, diagnosticar e solucionar problemas comuns que possam surgir durante o processo de montagem e programação do robô, assegurando que o autômato esteja em plenas condições para enfrentar os desafios da competição sumô.

8

**3- JUSTIFICATIVA**

A concepção do Robô Sumô encontra sólidas justificativas que fundamentam o propósito deste projeto, as quais são cruciais para a compreensão de sua relevância:

Impacto Social da Automação e Robótica:

O cenário contemporâneo testemunha o impacto crescente da automação e robótica em diversos setores, desde a manufatura até a assistência médica. Projetos educacionais voltados para a robótica, como o Robô Sumô, não apenas inspiram, mas também preparam os alunos para carreiras futuras em campos tecnológicos em constante expansão. A habilidade de construir e programar um Robô Sumô não só oferece aprendizado prático, mas também fornece respostas para os desafios contemporâneos envolvendo automação e controle de dispositivos autônomos.

Aquisição de Habilidades Valiosas:

Ao participar ativamente da construção e programação de um Robô Sumô, os alunos adquirem habilidades valiosas em eletrônica, programação e engenharia mecânica. Essas competências são altamente relevantes em uma

9

sociedade cada vez mais tecnológica, preparando os estudantes para enfrentar os desafios e oportunidades proporcionados pelo avanço tecnológico.

Dinamismo e Avanços na Robótica:

O campo da robótica é dinâmico, com avanços constantes. Este projeto alinha-se com a necessidade de atualizar e expandir o conhecimento em robótica e automação, especialmente na aplicação prática de robôs sumôs. Além disso, proporciona uma oportunidade enriquecedora para explorar conceitos teóricos e práticos em um ambiente de aprendizado interdisciplinar.

Preparação para a Era Digital:

Em resumo, a justificativa deste projeto é impulsionada pela necessidade premente de preparar os alunos para as oportunidades e desafios da era digital. Ao fomentar a aquisição de habilidades práticas, estimular a criatividade e contribuir para o avanço contínuo da robótica e automação, este projeto busca ser um agente catalisador no processo educacional, capacitando os estudantes a enfrentarem os complexos cenários tecnológicos de nossa sociedade em constante evolução.

10

**4- METODOLOGIA**

Este projeto adota uma abordagem prática e aplicada, caracterizando-se como pesquisa experimental. Essa metodologia permite a construção e teste direto do Robô Seguidor de Faixa, utilizando um conjunto específico de componentes eletrônicos, aliado à programação para assegurar autonomia na navegação.

Os principais instrumentos e fontes de dados para este projeto incluem:

Componentes Eletrônicos:

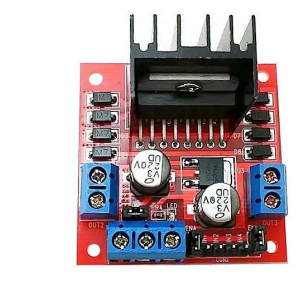
Esp32 x 01



FONTE: MakerHero

11

Ponte H x 01



FONTE: Robótica Educacional Brasil

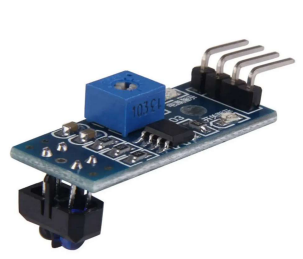
Motores DC x 02



FONTE: Eletrogate

12

Sensor infravermelho tcrt5000 x 02



FONTE: Casa da Robótica

Sensor de Distancia Ultrassônico x 01



FONTE: Eletrogate

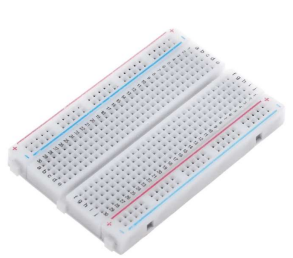
13

Cabo Jumper Macho X Fêmea x 20



FONTE: Casa da Robótica

Protoboard x 01



FONTE: Casa da Robótica

14

Código-Fonte: A programação do robô foi desenvolvida no ambiente Arduino IDE

usando a linguagem do ESP32. Segue o código abaixo:

// Robo sumo V2

/\* DECLARAÇÃO DE PINOS \*/

#define MotorDireito\_sentido1 27

#define MotorDireito\_sentido2 14

#define MotorEsquerdo\_sentido1 25

#define MotorEsquerdo\_sentido2 26

#define MotorDireito\_PWM 33

#define MotorEsquerdo\_PWM 32

#define Sensor\_direita 16

#define Sensor\_esquerda 17

#define Sensor\_ultrassomTrig 19

#define Sensor\_ultrassomEcho 18

/\* CONSTANTES \*/

#define DISTANCIA\_INIMIGO 30 // Distância para considerar um inimigo (ajuste conforme necessário)

#define TEMPO\_MAXIMO\_GIRO 50000 // Tempo máximo de giro em milissegundos (50 segundos)

unsigned long tempoInicioGiro = 0; // Variável para armazenar o início do giro

void moveFrente() {

digitalWrite(MotorDireito\_sentido1, HIGH);

digitalWrite(MotorDireito\_sentido2, LOW);

analogWrite(MotorDireito\_PWM, 150);

digitalWrite(MotorEsquerdo\_sentido1, HIGH);

digitalWrite(MotorEsquerdo\_sentido2, LOW);

15

analogWrite(MotorEsquerdo\_PWM, 150);

}

void moveGirar() {

digitalWrite(MotorDireito\_sentido1, HIGH);

digitalWrite(MotorDireito\_sentido2, LOW);

analogWrite(MotorDireito\_PWM, 73);

digitalWrite(MotorEsquerdo\_sentido1, LOW);

digitalWrite(MotorEsquerdo\_sentido2, HIGH);

analogWrite(MotorEsquerdo\_PWM, 73);

}

void parar() {

digitalWrite(MotorDireito\_sentido1, LOW);

digitalWrite(MotorDireito\_sentido2, LOW);

analogWrite(MotorDireito\_PWM, 0);

digitalWrite(MotorEsquerdo\_sentido1, LOW);

digitalWrite(MotorEsquerdo\_sentido2, LOW);

analogWrite(MotorEsquerdo\_PWM, 0);

}

int medirDistancia() {

digitalWrite(Sensor\_ultrassomTrig, LOW);

delayMicroseconds(2);

digitalWrite(Sensor\_ultrassomTrig, HIGH);

delayMicroseconds(10);

digitalWrite(Sensor\_ultrassomTrig, LOW);

return pulseIn(Sensor\_ultrassomEcho, HIGH) / 58.0; // Converte o tempo de retorno em centímetros

}

16

// Função para verificar se a linha branca foi detectada pelos dois sensores

bool linhaBrancaDetectada() {

// Lógica de detecção da linha branca com os dois sensores

// Substitua a lógica abaixo com a lógica específica para seus sensores

return (digitalRead(Sensor\_esquerda) == HIGH && digitalRead(Sensor\_direita) == HIGH);

}

void setup() {

Serial.begin(115200); // Inicia a comunicação serial

Serial.println("Controle do Robô com Detecção de Inimigos e Bordas!");

pinMode(MotorDireito\_sentido1, OUTPUT);

pinMode(MotorDireito\_sentido2, OUTPUT);

pinMode(MotorEsquerdo\_sentido1, OUTPUT);

pinMode(MotorEsquerdo\_sentido2, OUTPUT);

pinMode(MotorDireito\_PWM, OUTPUT);

pinMode(MotorEsquerdo\_PWM, OUTPUT);

pinMode(Sensor\_direita, INPUT);

pinMode(Sensor\_esquerda, INPUT);

pinMode(Sensor\_ultrassomTrig, OUTPUT);

pinMode(Sensor\_ultrassomEcho, INPUT);

}

void avancarPorCentimetros(int centimetros) {

moveFrente(); // Inicia o movimento para frente

delay(centimetros \* 10); // Delay proporcional ao número de centímetros a percorrer

parar(); // Para o movimento

}

void loop() {

int sensorDireita = digitalRead(Sensor\_direita);

int sensorEsquerda = digitalRead(Sensor\_esquerda);

17

if (sensorDireita == HIGH && sensorEsquerda == HIGH) {

// Etapa inicial: avançar alguns centímetros

avancarPorCentimetros(1); // Ajuste o número de centímetros conforme necessário

// Lógica para detecção de inimigos após avançar

while (true) {

int distancia = medirDistancia(); // Mede a distância em centímetros

Serial.print("Distância: ");

Serial.print(distancia);

Serial.println(" cm");

// Lógica para detecção de inimigos

if (distancia <= DISTANCIA\_INIMIGO) {

Serial.println("Inimigo detectado! Movendo imediatamente em direção a ele.");

parar(); // Para o movimento antes de seguir em direção ao inimigo

moveFrente();

delay(1000); // Move em direção ao inimigo por 1 segundo

parar(); // Para o movimento após avançar

break; // Sai do loop após detectar e seguir em direção ao inimigo

} else {

// Lógica para girar até encontrar um inimigo

Serial.println("Girando até encontrar um inimigo.");

// Se não começou a girar, registra o início do giro

if (tempoInicioGiro == 0) {

tempoInicioGiro = millis();

}

// Se o tempo de giro atingir o limite, avança mais 10 centímetros

if (millis() - tempoInicioGiro >= TEMPO\_MAXIMO\_GIRO) {

Serial.println("Tempo máximo de giro atingido. Avançando mais 10 centímetros.");

18

parar(); // Para o movimento antes de avançar

avancarPorCentimetros(5);

break; // Sai do loop após avançar

}

moveGirar();

}

}

} else {

Serial.println("Sensores não detectam linha branca. Interrompendo ação.");

parar(); // Para qualquer ação caso os sensores não estejam detectando uma linha branca

}

tempoInicioGiro = 0; // Reseta o tempo de início do giro para a próxima iteração

delay(1000); // Aguarda 1 segundo antes de começar a próxima iteração

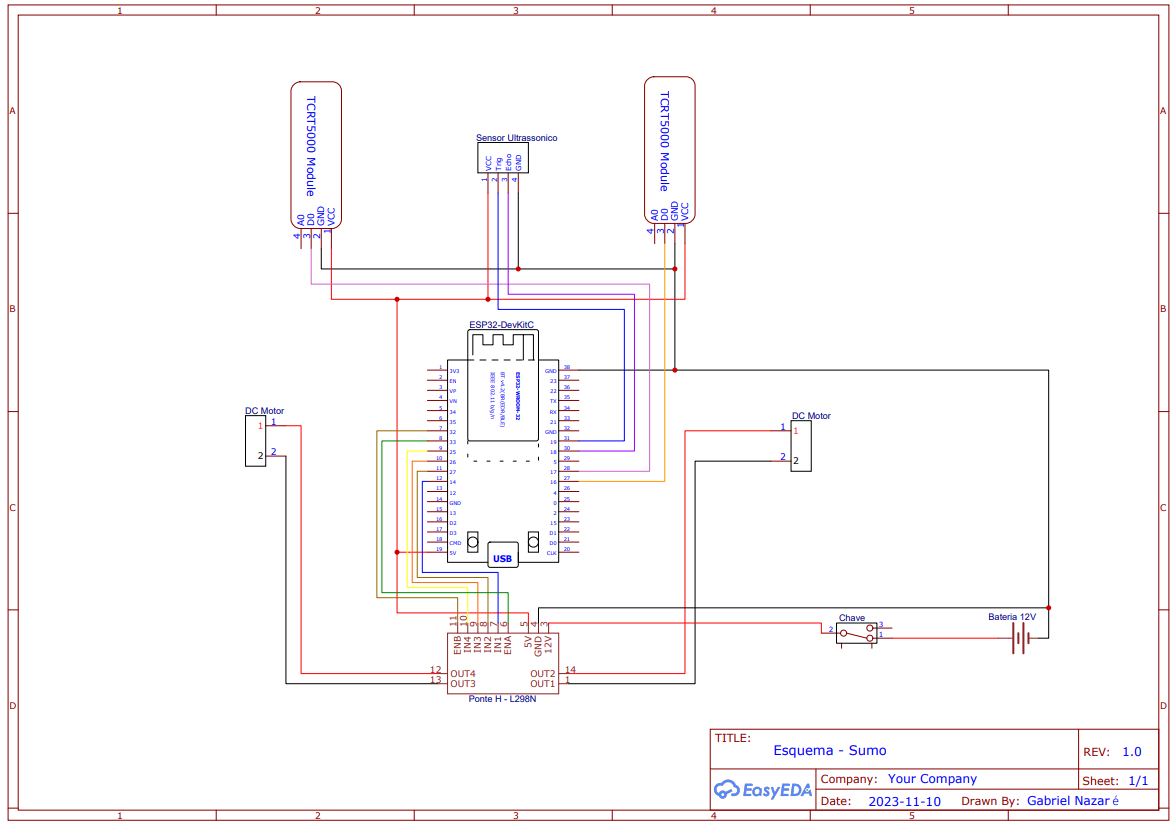
}

FONTE: Autoria

Manuais Técnicos: Manuais técnicos e Datasheets dos componentes eletrônicos foram consultados para garantir a configuração correta.

Montagem do Robô: O chassi do robô foi montado a partir de peças impressas em 3D, criadas com uma impressora 3D. As conexões elétricas foram feitas com base nos diagramas elétricos a seguir.

19



FONTE: Autoria

Testes e Ajustes: O robô foi testado em um ambiente controlado, onde deveria seguir uma linha preta de forma autônoma. Foram feitos ajustes no código e na montagem conforme necessário para garantir o funcionamento correto.

Durante o desenvolvimento deste projeto, foram utilizados os seguintes recursos de

informação:

Bibliotecas da Instituição: A biblioteca da instituição de ensino foi consultada para obter materiais de referência, manuais técnicos e literatura relacionada à robótica.

20

Recursos Online: A internet foi uma fonte de informações, incluindo tutoriais, fóruns de discussão e documentação técnica disponível online sobre os componentes e a

programação utilizada no ambiente Arduino.

Lojas e Fornecedores: Os materiais e componentes básicos na concepção do projeto foram adquiridos para a Fornecedores de componentes21

**6- CRONOGRAMA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ETAPAS** | **SEMANA**  **01-02** | **SEMANA**  **03-04** | **SEMANA**  **05-06** | **SEMANA**  **07-08** | **SEMANA**  **09-10** | **SEMANA**  **11-12** |
| **Definição de Requisitos e Objetivos** | **X** |  |  |  |  |  |
| **Design Mecânico** | **X** | **X** |  |  |  |  |
| **Design Eletrônico** |  | **X** |  |  |  |  |
| **Montagem Mecânica e Eletrônica** |  | **X** | **X** |  |  |  |
| **Teste Eletrônico** |  |  | **X** |  |  |  |
| **Programação** |  |  | **X** | **X** |  |  |
| **Testes e Depuração** |  |  |  | **X** | **X** |  |
| **Aprimoramento e Otimização** |  |  |  |  | **X** | **X** |
| **Documentação e Relatório** |  |  |  |  |  | **X** |
| **Competição** |  |  |  |  |  | **X** |

22

**7-BIBLIOGRAFIA**

Material da Internet

Simulador de projetos IoT de navegador WOKWI

Disponível em: <https://wokwi.com/projects/new/esp32>. Acesso em: 18 out. 2023

Figuras ilustrativas dos componentes eletrônicos do robô. Disponível em:

<https://www.casadarobotica.com/prototipagem-e ferramentas/prototipagem/protoboard/protoboard-400-furos-pontos>

<https://www.eletrogate.com/kit-motor-dc-3-6v-roda 68mm?utm\_content=&utm\_term=&gad=1>

Data Sheets  
https://www.alldatasheet.com/datasheet-pdf/pdf/26406/VISHAY/TCRT5000.html

https://www.alldatasheet.com/view.jsp?Searchword=ESP32

Guia Definitivo para Ponte H  
https://blog.eletrogate.com/guia-definitivo-de-uso-da-ponte-h-l298n/

A biblioteca da instituição de ensino foi consultada para obter materiais de referência,

manuais técnicos e literatura relacionada à robótica, conteúdo disponibilizado por HELIO

Disponível em:

<https://avea.univap.br/moodle/pluginfile.php/62140/mod\_resource/content/0/01x01.pdf

>

<https://avea.univap.br/moodle/pluginfile.php/64189/mod\_resource/content/0/01x02.pdf

>

<https://avea.univap.br/moodle/pluginfile.php/64971/mod\_resource/content/0/01x03.pdf

>