

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC SYLVIO DE MATTOS CARVALHO

Curso Técnico em Mecatrônica M-tec

LAUANY NICOLLY WETTERICH

MARCOS VINICIUS ZAMBELLI

VITORIA CORREIA DE OLIVEIRA

VITORIA PEREIRA AURELIO DA SILVA

**DESENVOLVIMENTO E OTIMIZAÇÃO DE ESTRATEGIAS DO
CONTROLE PARA ROBOS DE SUMO**

Matão, SP

2024

SUMÁRIO

1. INTRODUÇÃO	3
2. OBJETIVOS (GERAL E ESPECÍFICOS)	4
3. DESCRIÇÃO DO PROJETO	5
3.1 Apresentação Geral do Projeto	5
3.2 Materiais Utilizados e suas Especificações Técnicas	6 a 15
4. CRONOGRAMA DE ATIVIDADES	16
4.1 Atividades e Metas 1º Semestre de 2024	16
4.2 Atividades e Metas 2º Semestre de 2024	17
5. DESENVOLVIMENTO DO PROJETO	18
5.1 Descrição do Funcionamento	18 a 19
5.2 Montagem Física do Projeto	20 a 22
5.3 Desenvolvimento do programa do robô	23
6. CONSIDERAÇÕES FINAIS	24
7. REFERENCIAS BIBLIOGRAFICAS	25
8. APÊNDICE	26 a 32

1 INTRODUÇÃO

Nos últimos anos, a robótica tem se tornado cada vez mais relevante, tanto na indústria quanto na educação. Segundo Asada e Uchibe (2008), "os robôs podem se tornar parceiros inteligentes e colaborativos em várias situações do cotidiano". Os robôs sumô, especificamente, têm se mostrado uma excelente plataforma para aprender e competir, permitindo explorar conceitos de engenharia, programação e estratégia. Akiyama et al. (2013) destacam que "os robôs sumô são ferramentas eficazes para ensinar mecânica, eletrônica e controle".

Este TCC se concentrará na criação e implementação de um robô sumô, abordando os aspectos mecânicos e os algoritmos de controle necessários para um desempenho competitivo. Como mencionado por Kajita et al. (2001). O robô será projetado para tomar decisões em tempo real com base em sensores que detectam a posição e movimento do oponente.

Os objetivos são dois: primeiro, projetar e construir um robô sumô capaz de competir com sucesso em torneios locais; e segundo, investigar e avaliar diferentes abordagens de controle e algoritmos de inteligência artificial para o robô. Siciliano e Khatib (2008) afirmam que "a robótica inteligente exige a integração de mecânica, eletrônica, computação e inteligência artificial para criar sistemas que percebam, raciocinem e ajam em ambientes complexos".

A metodologia incluirá pesquisa bibliográfica, projeto e simulação, construção do protótipo, testes e análise de resultados. O processo de desenvolvimento será iterativo, com a integração gradual de mecânica, eletrônica e software para otimizar o desempenho e a confiabilidade do robô.

Este projeto de TCC oferece uma oportunidade única para aplicar conhecimentos teóricos e práticos adquiridos no curso técnico em mecatrônica, desafiando os alunos a desenvolver um robô sumô que integra mecânica, eletrônica e programação. Além de permitir a exploração de conceitos fundamentais de controle e sensoramento, o projeto ajudará a desenvolver habilidades essenciais como resolução de problemas, trabalho em equipe e pensamento crítico.

2 OBJETIVOS (GERAL E ESPECÍFICOS)

2.1 Objetivo Geral

O objetivo geral do projeto é:

- Construir um robô de sumô

2.2 Objetivos Específicos

Os objetivos específicos do projeto são:

- Implementar estratégias de movimentação ágil para evitar ser empurrado para fora do ringue.
- Aprimorar a força de empurrão do robô por meio de um design otimizado de motores e engrenagens.
- Definir claramente as responsabilidades de cada membro do grupo para garantir uma distribuição equilibrada das tarefas.
- Estabelecer um cronograma de reuniões regulares para monitorar o progresso e discutir desafios.

3 DESCRIÇÃO DO PROJETO

3.1 Apresentação Geral do Projeto

O robô tem o formato de um veículo pequeno, contendo quatro rodas de borracha nas extremidades do chassi, acionadas cada uma através de um motor CC.

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. Em funcionamento, através dos comandos no Arduino Nano e Bluetooth na qual controlamos os comandos com o aplicativo Arduino Car.

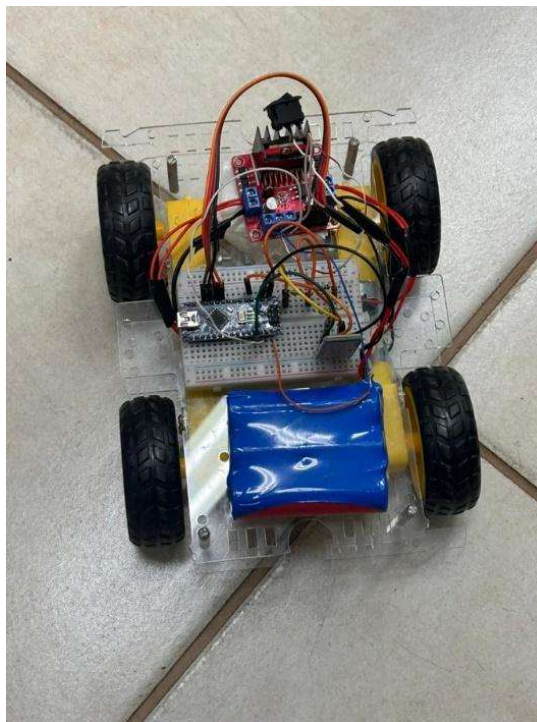


Figura 1 – Foto do projeto montado

FONTE: Elaborado pelos autores (2024).

3.2 Materiais Utilizados e suas Especificações Técnicas

Os materiais utilizados em nosso projeto estão listados na Tabela 1 e uma breve descrição é apresentada a seguir.

Tabela 1 – Lista de Materiais

Material	Quantidade	Preço
Arduino Nano	1	24,99
Módulo Ponte H L298n	1	24,90
Módulo Bluetooth HC-05	1	16,37
Motor com Redutor	4	16,00
Roda de borracha	4	17,16
Chassi	1	130,00
Protoboard	1	8,46
Kit Jumper	1	9,99
Bateria de íon-Lítio	1	28,99
	Total	276,86

FONTE: Elaborado pelos autores (2024).

Alguns dos materiais foram custeados pelo professor Rogério Varavallo.

O Arduino Nano é uma placa de prototipagem com o microcontrolador ATmega328p, um conversor USB-UART e componentes de alimentação e temporização. Sua aplicação inclui controlar os motores e o sistema de comunicação para a movimentação do robô durante a luta.



Figura 2 - Arduino Nano

FONTE: Elaborado pelos autores (2024).

A Ponte H tem como principal função controlar a velocidade e o sentido de motores DC escovados. Também pode ser usada para controlar a saída de um gerador DC ou como inversor monofásico. O termo "Ponte H" deriva de sua representação gráfica típica. Sua aplicação inclui receber sinais do Arduino e transmitir aos motores a direção e o sentido de giro do robô.

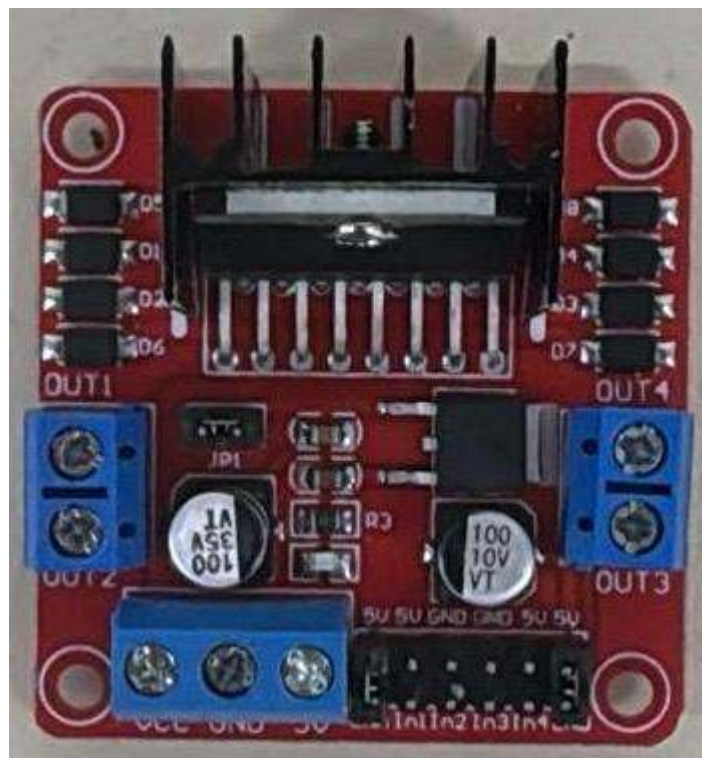


Figura 3 - Ponte H

FONTE: Elaborado pelos autores (2024).

O Bluetooth HC-05 Módulo para comunicação sem fio via Bluetooth, atuando como Master, em ambos os modos, controla a direção do robô, permitindo movimentos para frente, para trás, à direita e à esquerda.



Figura 4 - Bluetooth HC-05

FONTE: Elaborado pelos autores (2024).

O motor com redução é um dispositivo mecânico que reduz a rotação(velocidade) de um acionador, geralmente um motor elétrico. Sua aplicação inclui movimentar o robô para frente e para trás, aumentar a força para enfrentar inimigos durante lutas e permitir giros de 360°.



Figura 5 - Motor de corrente contínua com redutor
FONTE: Elaborado pelos autores (2024).

A roda de Tração recebem a potência do motor e impulsionam o movimento do veículo, são essenciais para a tração do robô, permitindo sua movimentação eficaz.



Figura 6- Roda de tração

FONTE: Elaborado pelos autores (2024).

Kit chassi utilizado na construção do robô de sumô



Figura 7- Kit chassi

FONTE: Elaborado pelos autores (2024).

O Protoboard é uma ferramenta essencial na eletrônica, permitindo a conexão de componentes como capacitores, resistores e circuitos integrados, facilitando a montagem precisa de circuitos. Utilizamos a protoboard para conectar o Arduino Nano montar o módulo Bluetooth e outros componentes do robô, permitindo testes e ajustes eficientes antes da montagem final.

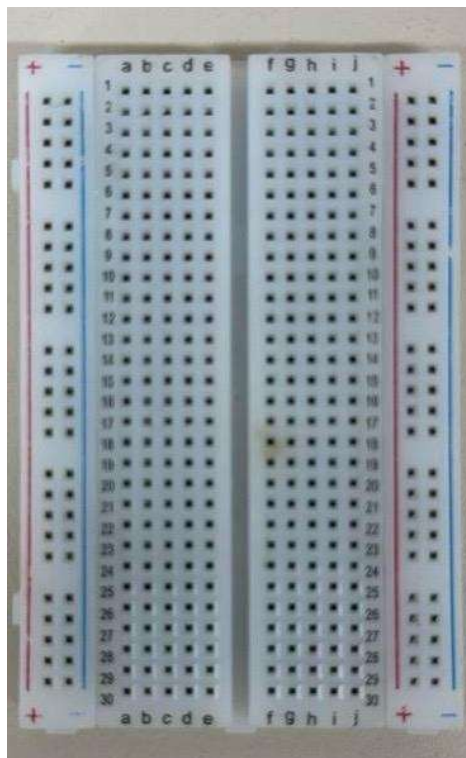


Figura 8 - Protoboard

FONTE: Elaborado pelos autores (2024).

O jumper é um produto 100% industrializado, tendo seus cabos grudados, mas podendo ser destacados para ser usados na quantidade desejada, além de poder ser reutilizado, visto que são totalmente compatíveis com os furos da protoboard

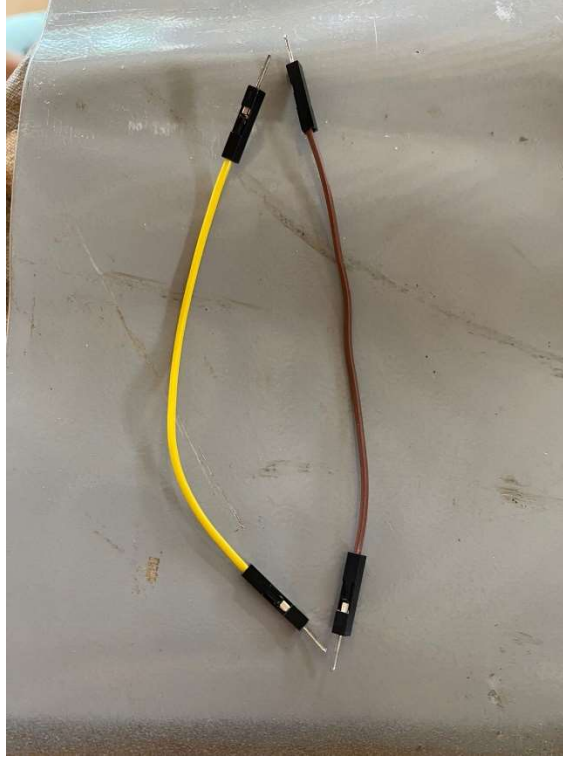


Figura 9 - Jumper

FONTE: Elaborado pelos autores (2024).

A Bateria de Lítio 12v 2200mah A bateria de lítio é um tipo de bateria recarregável que utiliza compostos de lítio como um dos eletrodos. É uma das melhores opções de bateria, pois oferece uma combinação de: Alta densidade energética, Longa vida útil, recarga rápida, baixa manutenção, sustentabilidade ambiental.



Figura 10 - Bateria de lítio 12V 2200mah
FONTE: Elaborado pelos autores (2024).

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2024

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

Atividade	Fevereiro		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 Qui nze na	1ª Qui nze na	2ª Qui nze na
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 Bixiga									x	x

FONTE: Elaborado pelos autores (2024).

4.2 Atividades e Metas 2º Semestre de 2024

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

Atividade	Julho		Agosto		Setembro		Outubro		Novembro		Dezembro
	1ª Qu inz en a	2ª Qu inz en a	1ª Qu inz en a	2ª Qu inz en a	1ª Qui nze na	2ª Qu inz en a	1ª Qu inz en a	2ª Qu inz en a	1ª Qui nze na	2ª Qui nze na	1ºquinzen a
Desenvolvimento do Boneco	x	x	X	x	X	x	x	x	x	x	
Diário de Bordo	x	x	X	x	X	x	x	x	x	x	
Escolha dos materiais para o chassi	x	x									
Análise dos custos		x									
Compra dos materiais para o chassi			X	x							
Montagem do chassi				x	X						
Programação do robô						x	x				
Testes no robô								x	x		
Batalha do robô sumô										x	
Entrega do relatório e do boneco											x

FONTE: Elaborado pelos autores (2024).

5 DESENVOLVIMENTO DO PROJETO

5.1 Descrição do Funcionamento

A Figura 11 e 12 apresentam o esquema elétrico de toda a parte eletrônica do projeto.

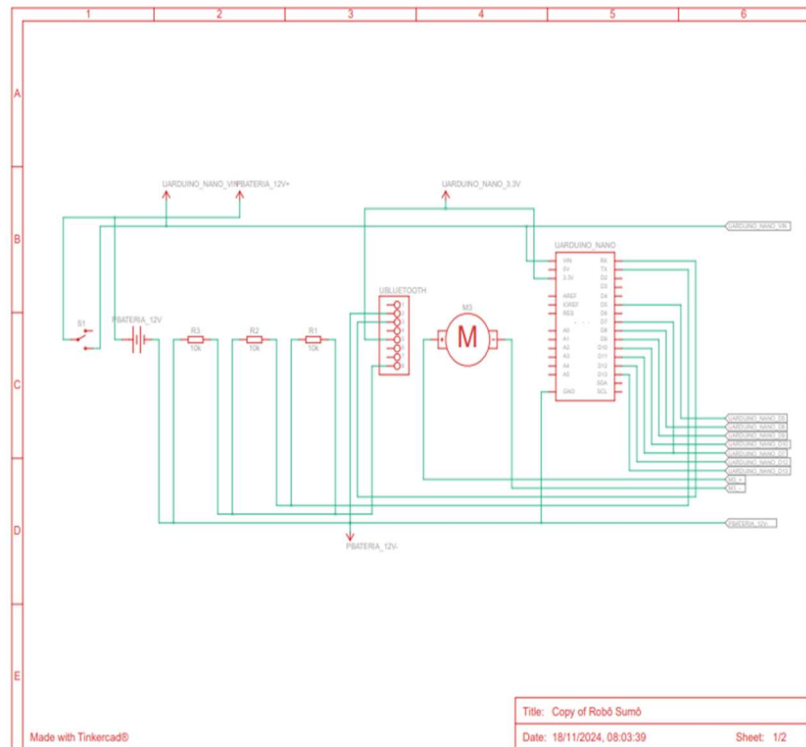


Figura 11 – Esquema elétrico do robô
FONTE: Elaborado pelos autores (2024).

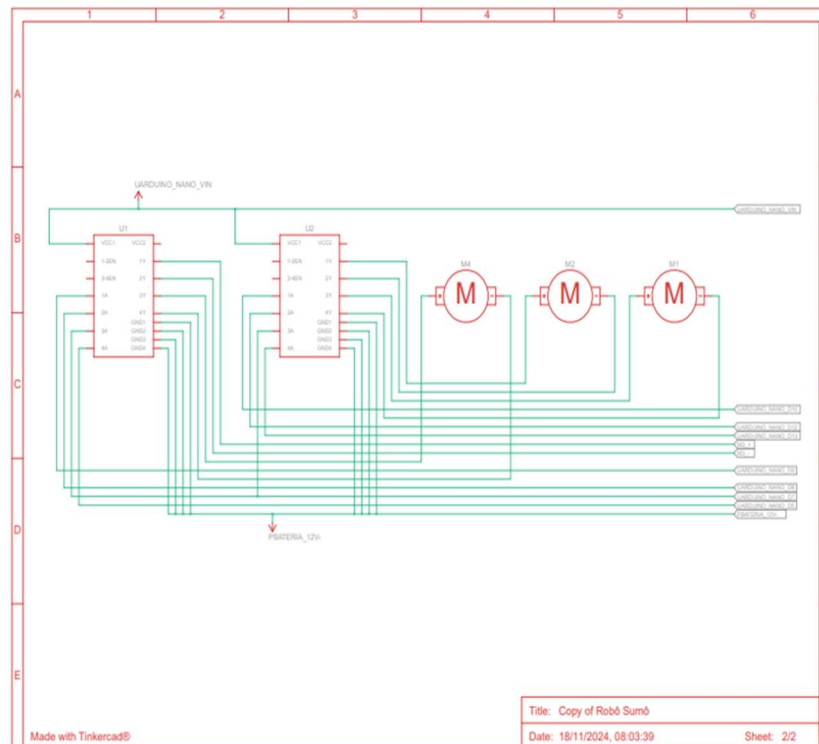


Figura 12 – Esquema elétrico do robô

FONTE: Elaborado pelos autores (2024).

O funcionamento da parte eletrônica é descrito a seguir.

Funcionamento da parte eletrônica:

Quando ligar o botão a bateria vai fornecer energia para todos os componentes principalmente para alimentar o “Arduino” que é o cérebro do projeto, também tem uma programação interna específica pro nosso projeto que vai se comunicando com a ponte H que já está sendo alimentada, o Arduino vai mandar um sinal para ponte H que faz o controle dos motores, esse controle é dado pelo aplicativo que irá fazer o robô se movimentar de acordo com as letras da programação.

5.2 Montagem Física do Projeto

As Figuras 12 a 15 apresentam a sequência da montagem física do nosso projeto.



Figura 12 – Peças utilizadas no robô

FONTE: Elaborado pelos autores (2024).

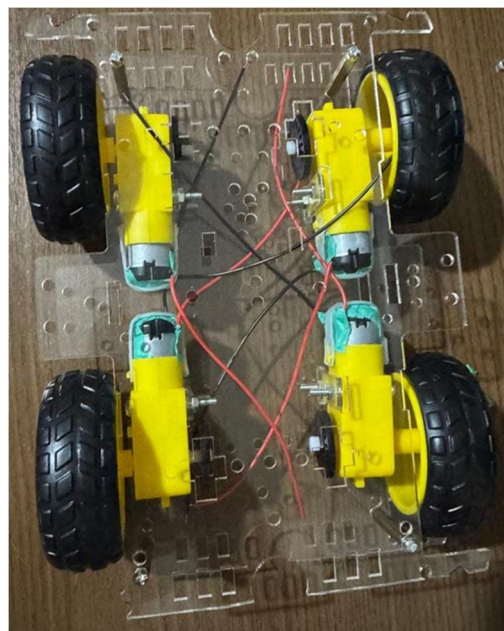


Figura 13 – Instalação das rodas e motores no chassi

FONTE: Elaborado pelos autores

(2024).

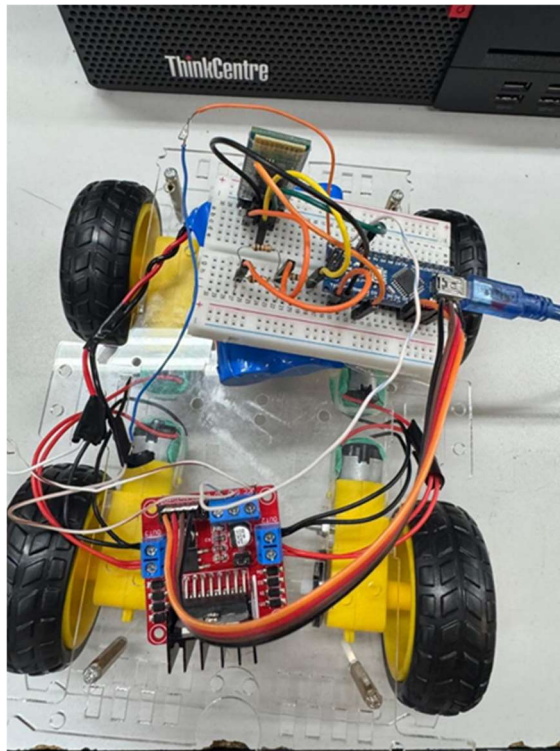


Figura 14 – Instalação do Arduino e módulos no chassi
FONTE: Elaborado pelos autores (2024).

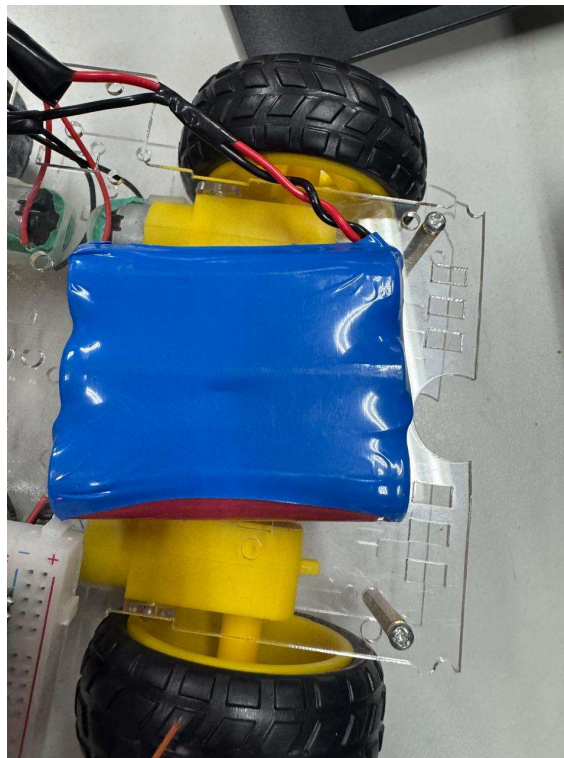


Figura 15 – Instalação da bateria no chassi

FONTE: Elaborado pelos autores (2024).

As Figuras 16 e 17 apresentam os testes finais efetuados em nosso projeto

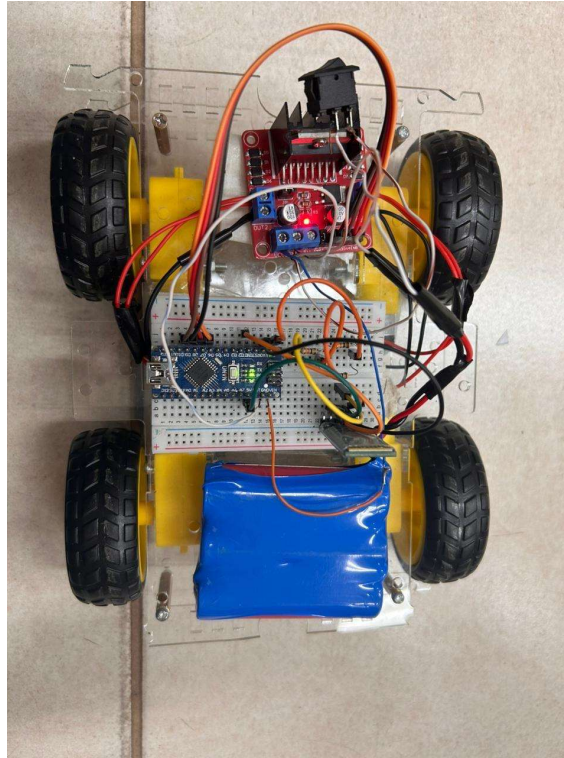


Figura 16 – Teste do hardware

FONTE: Elaborado pelos autores (2024).

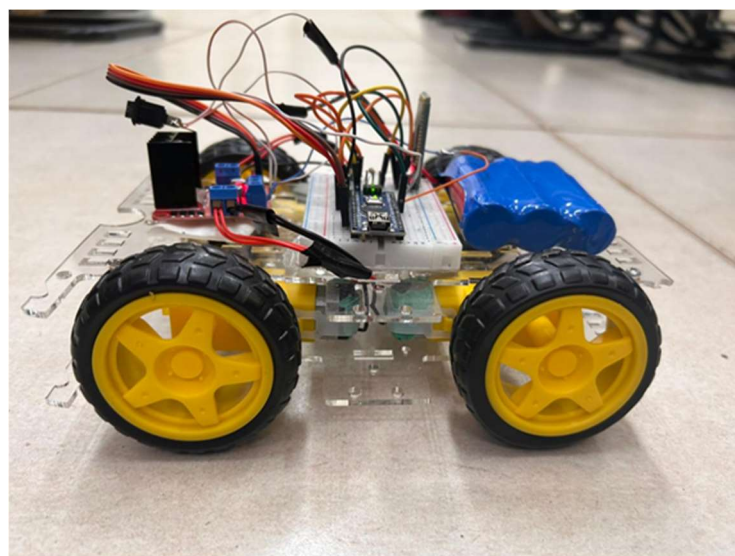


Figura 17 – Teste geral do robô

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 18 apresenta o fluxograma do programa.

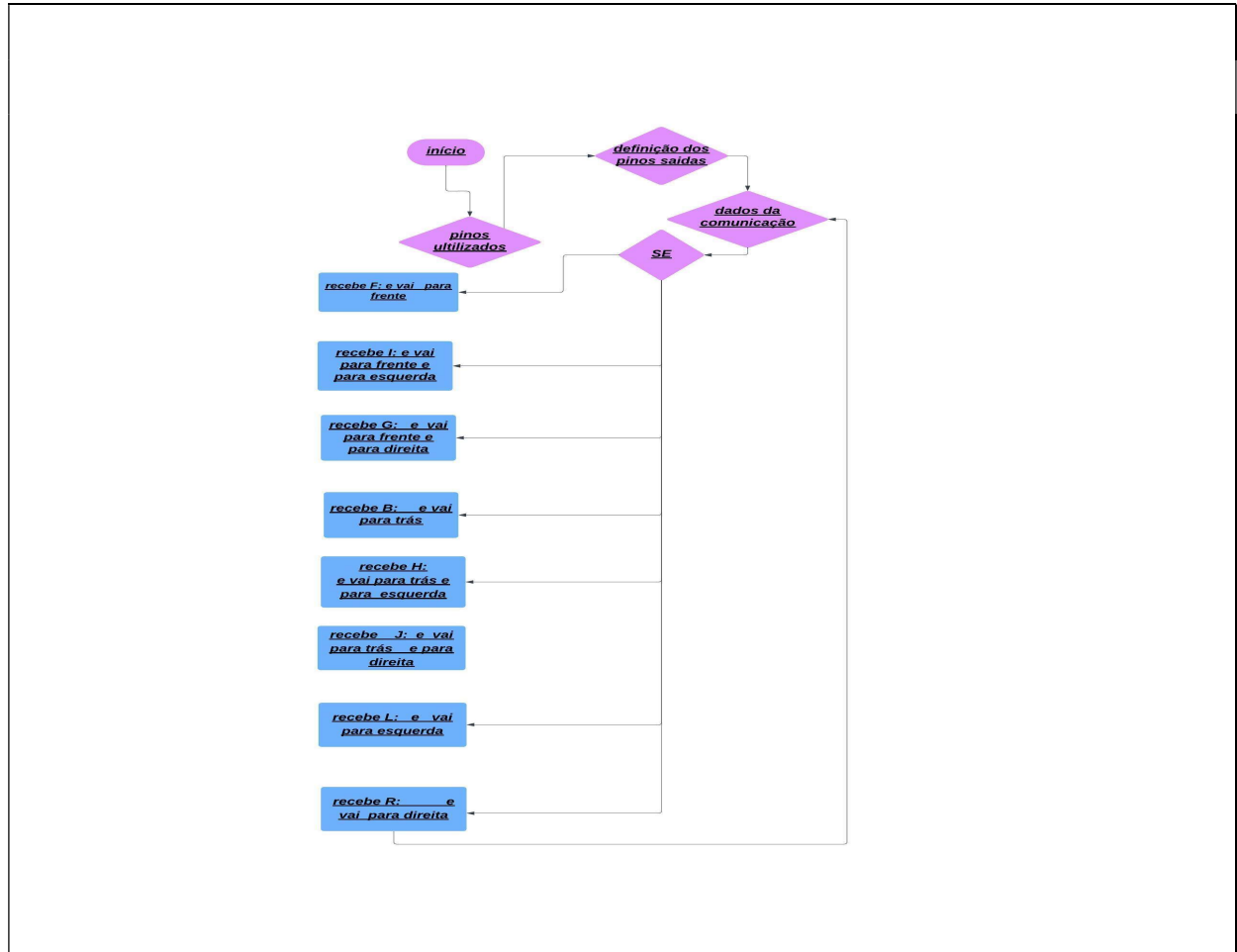


Figura 18 – Fluxograma do programa do robô

FONTE: Elaborado pelos autores (2024).

Conforme apresentado no fluxograma, o início é ligado aos pinos utilizados, passando pela definição dos pinos de saída, conectados aos dados da comunicação que se conectam ao SE. Os pinos são (F,I,G,B,H,J,L e R) nas quais recebe F; e vai para frente, recebe I; e vai para frente e para a esquerda, recebe G; e vai para a frente e para a direita, recebe B; e vai para trás, recebe H; e vai para trás e para a esquerda, recebe J; e vai para trás e para a direita, recebe L; e vai para a esquerda, recebe R; e vai para a direita ao qual é ligado aos dados de comunicação. O código fonte desenvolvido no Arduino IDE é apresentado no Apêndice A.

6 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo o desenvolvimento de um robô de sumô, que por meio de sistemas de controle, sensores e atuadores, foi projetado para competir em um ambiente controlado, seguindo as regras e dinâmicas da modalidade de robô sumô. A proposta envolveu não apenas a construção física do robô, mas também a implementação de algoritmos de controle, a integração de sensores para detecção e evasão de obstáculos, bem como a otimização da estratégia de combate para empurrar o oponente para fora da arena.

Durante o desenvolvimento do projeto, foi possível explorar diversos conceitos da robótica, como cinemática, dinâmica, controle de motores e sistemas de sensores. O projeto também envolveu um aprendizado prático sobre como a interação entre hardware e software é essencial para o desempenho eficiente do robô em um ambiente competitivo. A escolha de componentes como [Ponte H, Arduino Nano, motores de corrente contínua, rodas, etc.] e o desenvolvimento de um algoritmo de controle eficiente permitiram que o robô executasse movimentos precisos, além de responder de forma dinâmica às mudanças no comportamento do oponente durante as lutas.

A principal contribuição deste trabalho foi a implementação de uma estratégia de combate que não só focou na potência de empurrão, mas também no comportamento inteligente, com o uso de sensores para detectar o oponente e otimizar a movimentação do robô. O protótipo desenvolvido foi capaz de competir de maneira satisfatória, apresentando bom desempenho nas arenas de sumô, evidenciando a eficácia das escolhas tecnológicas feitas durante o desenvolvimento.

Contudo, algumas limitações ainda foram observadas durante os testes realizados. Entre os desafios enfrentados, destaca-se a necessidade de uma maior precisão no controle de movimento em ambientes mais dinâmicos, além da otimização do consumo energético, que poderia permitir uma maior autonomia de operação do robô. Outro ponto que pode ser melhorado diz respeito à capacidade de adaptação do robô às diferentes estratégias adotadas pelos oponentes, o que demanda avanços nos algoritmos de inteligência artificial utilizados para o controle do comportamento do robô durante a competição.

REFERÊNCIAS BIBLIOGRÁFICAS

1. MINAKAWA, Riccieli; TIOSSO, Luiz Eduardo; SANTOS, Wellington Oliveira dos; SILVA, Lucas Tsutsui da. **Ambiente de Desenvolvimento Arduino**. Fundação Universal Federal de Mato Grosso do Sul. Faculdade de Computação – FACOM.
2. MINAKAWA, Riccieli; TIOSSO, Luiz Eduardo; SANTOS, Wellington Oliveira dos; SILVA, Lucas Tsutsui da. **Descrição do Funcionamento de uma Protoboard**. Fundação Universal Federal de Mato Grosso do Sul. Faculdade de Computação – FACOM.
3. CONCEIÇÃO, Patrick Silva da; LADIVEZ, Paulo Sebastião; BRUNO, Daniel Otavio Tambasco; OLIVEIRA JUNIOR, Vicente Gomes de. **Comunicação Bluetooth**. *Revista Brasileira de Mecatrônica*, São Caetano do Sul, v. 4, n. 1, p. 72-89, jul./set. 2021.
4. ASADA, Hirose; UCHIBE, Eiichi. **Robótica: uma introdução**. In: *Robotics: Modelling, Planning and Control*. 1. ed. Springer, 2008.
5. AKIYAMA, Masayuki; KUBO, Kenji; OKADA, Masahiro. **Educational Robotics: A Framework for Designing and Building Robots for Teaching Control**. In: *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*. 2013.
6. KAJITA, Satoshi; KANEKO, Kazuhiro; YOKOI, Katsushi; HIRUKAWA, Hiroshi. **The Real-time Control of a Sumô Robot**. In: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2001.
- 7 SICILIANO, Bruno; KHATIB, Oussama. **Springer Handbook of Robotics**. 2. ed. Springer, 2008.

APÊNDICE

```
// CARRO CONTROLADO POR BLUETOOTH

// === 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
    }
}

```