

**CENTRO PAULA SOUZA
ETEC PREF. ALBERTO FERES**

Mecatrônica

João Augusto Saturnino Costa

Matheus Francisco Januário Alves

Miguel Henrique Lemos Marreiro

Mirela Da Silva Sousa

**Desenvolvimento e Implementação de um Sistema Inteligente de Coleta de
Lixo com Utilização de Robótica**

Araras

2024

João Augusto Saturnino Costa

Matheus Francisco Januário Alves

Miguel Henrique Lemos Marreiro

Mirela Da Silva Sousa

Desenvolvimento e Implementação de um Sistema Inteligente de Coleta de Lixo com Utilização de Robótica

Trabalho de Conclusão de Curso apresentado ao curso técnico em mecatrônica da Etec Prefeito Alberto Feres, orientado pelo professor Adalberto Zechin, como requisito parcial para obtenção do título de técnico em mecatrônica.

Araras

2024

DEDICATÓRIA

Dedicamos este trabalho á nossos amigos, que nos apoiaram igualmente nós os apoiamos com seus TCCs, agradecemos o nossos professores do curso que nos ajudaram para o concerto dos erros que aconteceram durante o projeto, igualmente agradecemos o bibliotecário da escola, que muitas vezes forneceu o lugar para o desenvolvimento do trabalho, e somos gratos a todos da família de cada integrante, que o tempo todo nos apoiaram.

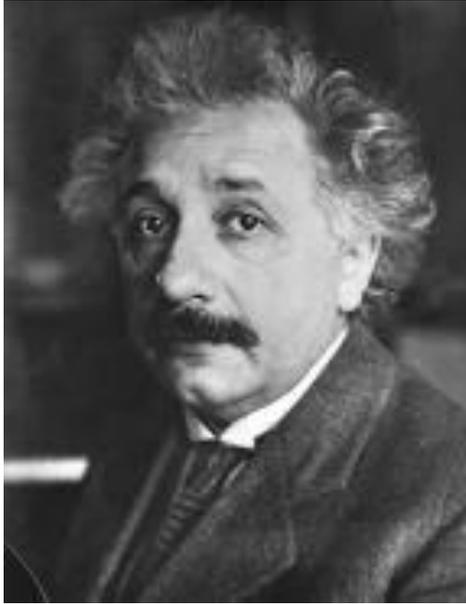
AGRADECIMENTOS

Primeiramente, obrigado aos meus professores, cujas aulas e ensinamentos foram fundamentais para o desenvolvimento deste TCC, especialmente aos professores Rochester e Guilherme, que, além de compartilharem seus vastos conhecimentos, forneceram ferramentas valiosas de pesquisa e orientação técnica, permitindo que nós pudéssemos explorar e aplicar os conceitos de mecatrônica de maneira prática. O incentivo e a dedicação deles foram essenciais em todo o projeto.

Agradecemos também ao meu orientador, Adalberto Zechin, pelo suporte contínuo, paciência e orientação específica ao longo de todo o percurso deste trabalho.

EPÍGRAFE

"Uma mente que se abre a uma nova ideia jamais voltará ao seu tamanho original"



- Albert Einstein

RESUMO

Este estudo aborda o desenvolvimento de um sistema robótico dedicado à coleta automatizada de resíduos. O objetivo principal é fornecer uma solução eficaz e eficiente para lidar com a gestão de entulho, reduzindo a poluição ambiental e promovendo práticas sustentáveis. O robô integra tecnologias de sensoriamento e controle autônomo para localizar, identificar e coletar resíduos de forma eficaz. Sua implementação representa benefícios tangíveis em termos de eficiência, redução de custos e minimização do impacto ambiental. Este estudo contribui para o avanço contínuo das soluções inovadoras para os desafios enfrentados na gestão de resíduos.

Palavras-chaves: robô; coleta; resíduos; sustentabilidade.

Abstract

This study addresses the development of a robotic system dedicated to automated waste collection. The main objective is to provide an effective and efficient solution to deal with waste management, reducing environmental pollution and promoting sustainable practices. The robot integrates sensing and autonomous control technologies to effectively locate, identify and collect waste. Its implementation represents tangible benefits in terms of efficiency, cost reduction and minimization of environmental impact. This study contributes to the continuous advancement of innovative solutions to the challenges faced in waste management.

Lista de Figuras

Figura 1: O que é o Arduino?.....	15
Figura 2: O que é servo motor? Funcionamento, tipos e como testar.....	16
Figura 3: Diferentes tipos de sensores e seu funcionamento.....	17
Figura 4 e 5: Kit Chassi 2WD Robô para Arduino - Bit Maker - Arduinos, Módulos, Sensores, 3D e robótica; Chassi Robô Móvel - 2WD.....	17
Figura 6: Chassi Tank com esteira.....	18
Figura 7: 1 -Comparação entre um braço humano e um robótico.....	19
Figura 8: Circuito genérico ponte H.....	20
Figura 9: Sensor De Obstáculos Reflexivo Infravermelho.....	21
Figura 10: Chassi Duplo 4WD Rodas Robótica Carro Robô.....	22
Figura 11: BlackBoard UNO R3.....	23
Figura 12: Kit motor DC + roda para Robô.....	24
Figura 13: Micro Servo 9g SG90.....	24
Figura 14: Braço Robótico em Acrílico com parafusos e Manual de Montagem para Arduino.....	25
Figura 15: Circuito genérico de funcionamento ponte H.....	26
Figura 16: Detalhando da Ponte H L298N.....	27
Figura 17: CARACTERÍSTICAS PRINCIPAIS DO ARDUINO UNO.....	28
Figura 18: Funcionamento do sensor de obstáculo reflexivo infravermelho IR em diferentes superfícies.....	30
Figura 19: Pinos e componentes do sensor de obstáculo reflexivo infravermelho IR.....	31
Figura 20: Montagem do chassi.....	32
Figura 21: Peças da garra.....	33
Figura 22: Montagem da garra.....	33
Figura 23: Line following robot.....	34

Figura 24: Sistema Elétrico Garra.....	35
Figura 25: Definição de pinos para o motor.....	36
Figura 26: Variáveis de velocidade.....	37
Figura 27: Configuração dos pinos no setup.....	37
Figura 28: Leitura dos sensores.....	37
Figura 29: Decisão de movimento.....	38
Figura 30: Motor esquerdo e direito giram para frente.....	38
Figura 31: Ambos os motores giram para trás.....	39
Figura 32: Motor esquerdo gira para frente e o direito para trás, fazendo o robô virar à direita.....	39
Figura 33: Motor direito gira para frente e o esquerdo para trás.....	39
Figura 34: Ambos os motores são desligados.....	40
Figura 35: Programação da Garra.....	41
Figura 36: Programação da Garra Continuação.....	42
Figura 37: Emenda feita errada.....	43
Figura 38: Emenda certa.....	43
Figura 39: Carro Andando.....	44
Figura 40: Robô com a garra.....	45

SUMÁRIO

1 INTRODUÇÃO	11
2 JUSTIFICATIVA	12
3 OBJETIVOS.....	13
3.1 Objetivo geral.....	13
3.2 Objetivos específicos.....	13
4 REFERENCIAL TEÓRICO.....	14
4.1 Arduino	14
4.2 Servo Motor	15
4.3 Sensor	16
4.4 Chassi	17
5 MATERIAIS E MÉTODOS	20
5.1 Materiais	21
5.1.2 Sensor de Obstáculos Reflexivo Infravermelho	21
5.1.2 Chassi	21
5.1.3 BlackBoard UNO.....	22
5.1.4 Servomotor	23
5.1.4.1 Motor DC 3-6v.....	23
5.1.4.2 Servo Sg90.....	24
5.1.5 Garra	25
5.1.6 Módulo L298N Ponte H.....	26
5.1.7 Especificações de componentes.....	26
5.1.7.1 Detalhando Componente Ponte H L298N	27
5.1.7.2 Detalhando Componente BlackBoard UNO.....	28
5.1.7.3 Detalhando Componente Sensor de Obstáculos Reflexivo Infravermelho.....	29
5.2 Métodos.....	31
5.2.1 Montagem do chassi	31
5.2.2 Montagem da garra.....	32
5.2.3 Sistema elétrico do robô segue linha	33
5.2.4 Sistema elétrico da Garra Robótica	35
5.2.5 Programação do Robô segue Linha	36
5.2.6 Programação da Garra Robótica	40
Resultados e Discussão	43
conclusão	46
REFERENCIAS.....	47

1 INTRODUÇÃO

A robótica é um campo em constante evolução que tem revolucionado diversos setores. O avanço tecnológico de robôs permite a automação de tarefas complexas, aumentando a eficiência e a precisão, ajudando em processos desde a exploração espacial até assistência pessoal.

Algumas tarefas em que a vida humana corre risco, podem contar com a ajuda de robôs para resolução de problemas, como acesso a áreas de incêndios ou gases perigosos. É essa uma das aplicações em que robôs devem ser utilizados, porém um trabalho que também pode ser atribuído as maquinas é a limpeza de alguns ambientes.

Com a utilização crescente da robótica em diversas áreas, ajudando os consumidores dos serviços oferecidos pelas máquinas, a automação no mercado cresceu cerca de 3% entre 2019 e 2020 de acordo com pesquisa da GS1. (Associação Brasileira de Automação,2020)

Os robôs comumente são controlados por humanos remotamente ou possuem inteligências que os tornam capazes de tomar as próprias decisões.

Para que o nosso robô consiga realizar ações, é necessário um controlador inteligente, que recebe informações, processa os dados e executa. Nesse projeto, foi utilizado o microcontrolador Arduino UNO. A execução do protótipo será realizada com a montagem e integração das demais peças do protótipo, e serem realizados os possíveis ajustes tanto na programação quanto no projeto físico.

2 JUSTIFICATIVA

De acordo com o relatório do Panorama dos Resíduos Sólidos no Brasil, publicado pela Associação Brasileira de Resíduos e Meio Ambiente (ABREMA,2023), estima-se que cada brasileiro gerou, em média, 1,04 kg de resíduos sólidos urbanos (RSU) por dia em 2022. Com base nos dados populacionais do Censo Demográfico 2022, calcula-se que o país produziu aproximadamente 77,1 milhões de toneladas de RSU ao longo do ano. Isso representa mais de 211 mil toneladas de resíduos gerados diariamente, ou cerca de 380 kg por habitante ao ano.

Estima-se que 93% dos resíduos gerados no Brasil em 2022 foram adequadamente coletados, o que corresponde a mais de 196 mil toneladas diárias de RSU recolhidos. Apesar de essa quantidade parecer significativa, os 7% de resíduos não coletados equivalem a mais de 5 milhões de toneladas que têm uma destinação final inadequada, o que representa uma ameaça ao meio ambiente e à saúde pública.

Além disso, na coleta de lixo, os trabalhadores são comumente expostos à altos níveis de exigência física, o que pode causar diversos agravos à saúde. São usuais também acidentes com materiais perfurocortantes, nos quais os trabalhadores podem se lesionar. Lesões essas que servem de porta de entrada para micro-organismos presentes no ambiente em que realizam a coleta.

Tendo em vista todos esses problemas, torna-se crucial refletir sobre como a tecnologia e inovação podem servir como solução para o aumento da eficiência da coleta de RSU, aumentando a quantidade e porcentagem de RSU devidamente coletados, e para a exoneração dos trabalhadores que se expõem diariamente à diversos riscos à saúde.

3 OBJETIVOS

3.1 Objetivo geral

A proposta do nosso TCC consiste no desenvolvimento de um robô que coleta resíduos das lixeiras, com o objetivo de facilitar e otimizar a coleta de lixo.

3.2 Objetivos específicos

Os objetivos específicos deste trabalho são projetar, construir e programar um robô seguidor de linha utilizando componentes como o Arduino UNO, servomotores e sensores LDR. Os objetivos específicos incluem:

- 1- Estudar os componentes utilizados no projeto.
- 2- Compreender as linguagens de programação utilizadas no sistema Arduino.
- 3- Desenvolvimento da estrutura física do robô.
- 4- Selecionar e integrar os componentes eletrônicos necessários.
- 5- Desenvolver o algoritmo necessário para realizar o movimento do robô, bem como o de sua garra.
- 6- Testar e otimizar o desempenho do robô em diferentes ambientes e condições, identificando possíveis restrições.

4 REFERENCIAL TEÓRICO

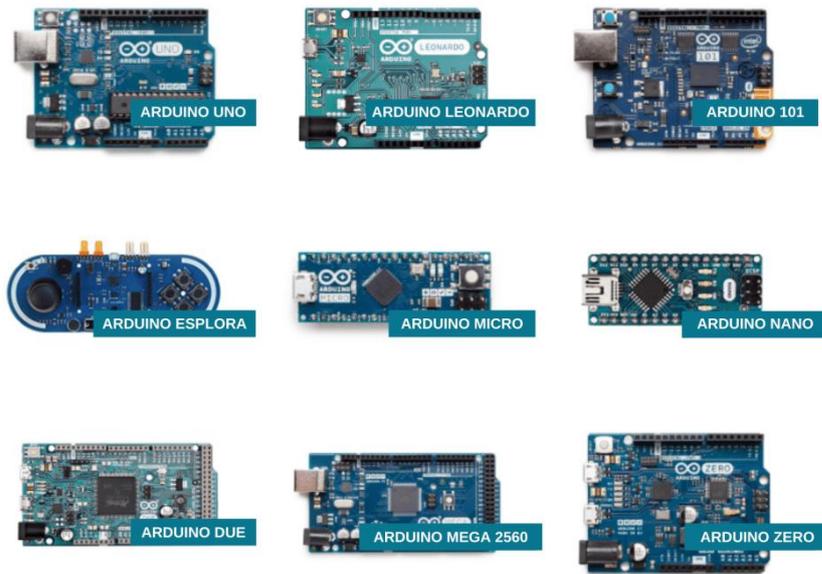
Para o desenvolvimento de um robô, alguns conhecimentos são necessários para uma melhor compreensão do projeto. Portanto, teremos a fundamentação teórica, na qual são especificadas algumas etapas do processo e os principais componentes necessários para atingir nosso objetivo.

4.1 Arduino

O Arduino é uma plataforma de desenvolvimento eletrônico de código aberto que se baseia em hardware e software de fácil utilização. As placas Arduino possuem a capacidade de ler diferentes tipos de entradas, como a luminosidade captada por um sensor, a ativação de um botão ou até mesmo uma mensagem proveniente de uma rede social, como o Twitter, e processá-las para gerar saídas, tais como o acionamento de um motor, o acendimento de um LED ou a publicação de informações online. Para programar as funções da placa, é necessário enviar um conjunto de instruções ao microcontrolador integrado. Esse processo é realizado por meio da Linguagem de Programação Arduino, derivada da plataforma Wiring, e utilizando o Software Arduino (IDE), baseado no ambiente de desenvolvimento Integrado. (Arduino.cc,2018)

A placa Arduino é composta por um microprocessador Atmel AVR, um cristal ou oscilador (um componente que gera pulsos de tempo em uma frequência específica para garantir o funcionamento adequado do sistema) e um regulador de voltagem de 5 volts. Vale destacar que algumas versões do Arduino utilizam reguladores de voltagem chaveados, enquanto outras, como o modelo Due, não são compatíveis com a voltagem de 5 volts. Dependendo do modelo de Arduino utilizado, a placa pode contar com uma saída USB, permitindo sua conexão a um computador, seja PC ou Mac, para a transferência de dados ou upload de programas. A placa também disponibiliza os pinos de entrada e saída do microcontrolador, facilitando sua conexão com outros circuitos, sensores e dispositivos. (McRoberts,2024)

Figura 1: O que é o Arduino?



Fonte: Blog da Robótica

blog da
ROBÓTICA

4.2 Servo Motor

O servo motor é um dispositivo eletromecânico empregado para movimentar com exatidão um objeto, permitindo sua rotação em ângulos ou distâncias específicas, assegurando o posicionamento e a velocidade.

Trata-se de um motor elétrico rotativo acoplado a um sensor que transmite a informação sobre sua posição, possibilitando o controle preciso da velocidade, aceleração e posição angular. Pode ser alimentado por corrente contínua ou alternada.

Recebe esse nome porque não realiza rotação livre e contínua como um motor convencional. Em vez disso, ele responde a um comando definido, ou seja, "serve" a um procedimento estabelecido.

É amplamente utilizado em sistemas de coordenadas e braços robóticos, drones, automação industrial, em diversas máquinas (tanto especiais quanto simples), aeromodelos de helicópteros e aviões, além de ser aplicado nas áreas aeroespacial, agrícola, de defesa, médica e em várias outras aplicações. (kalatec automação,2024)

Figura 2: O que é servo motor? Funcionamento, tipos e como testar.



Fonte: makerhero

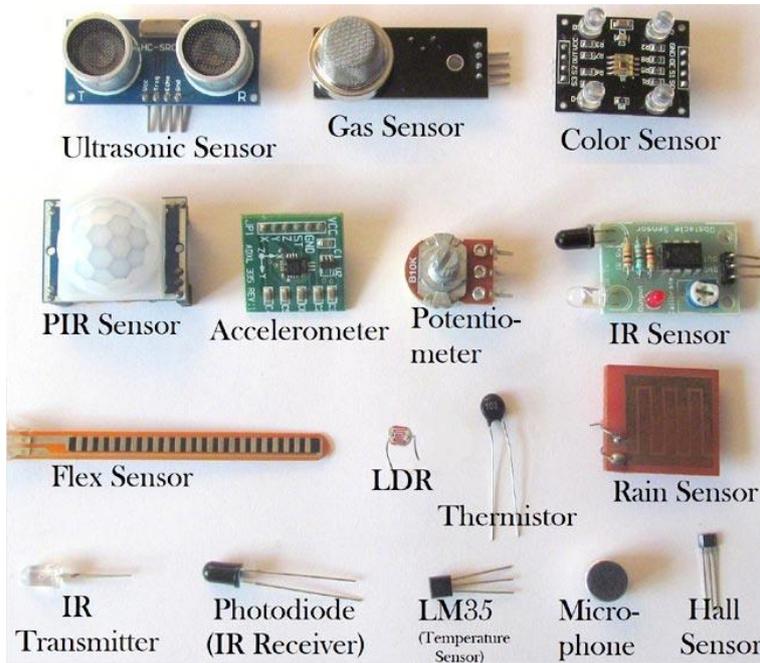
4.3 Sensor

Sensores servem para informar um circuito sobre um evento externo, evento sobre o qual ele deva atuar, ou a partir do qual ele comande uma determinada ação.

Esse termo refere-se a dispositivos que são sensíveis a diferentes formas de energia do ambiente, como luz, calor ou movimento, e que fornecem informações sobre uma grandeza física a ser medida, como temperatura, pressão, velocidade, corrente, aceleração, posição, entre outros.

Quando a saída de um sensor, ao ser ativado por uma energia externa, apresenta um nível de tensão muito baixo, é necessário amplificá-la. Para isso, utiliza-se um amplificador, que aumenta o nível do sinal para que possa ser utilizado.

Figura 3: Diferentes tipos de sensores e seu funcionamento.



Fonte: Cap Sistema

4.4 Chassi

O chassi é a estrutura física que compõe o robô. No mercado, há diversas opções de tamanhos, formas e modelos de chassi, que podem ser feitos de acrílico, plástico ou outros materiais. Geralmente, essa estrutura vem quase completa, necessitando apenas da montagem do kit. (Cardoso, 2016)

Figura 4 e 5: Kit Chassi 2WD Robô para Arduino - Bit Maker - Arduinos, Módulos, Sensores, 3D e robótica; Chassi Robô Móvel - 2WD



Fonte: Bit Maker; Loja Vida de Silício.

O chassi para robô móvel pode ser tradicional, com duas ou quatro rodas, ou do tipo esteira, que é ideal para terrenos irregulares. Os robôs com esteiras conseguem superar obstáculos maiores do que aqueles que um robô com rodas convencionais não conseguiria. No entanto, esse tipo de robô requer um consumo maior de energia durante seu deslocamento. (Cardoso, 2016)

Figura 6: Chassi Tank com esteira



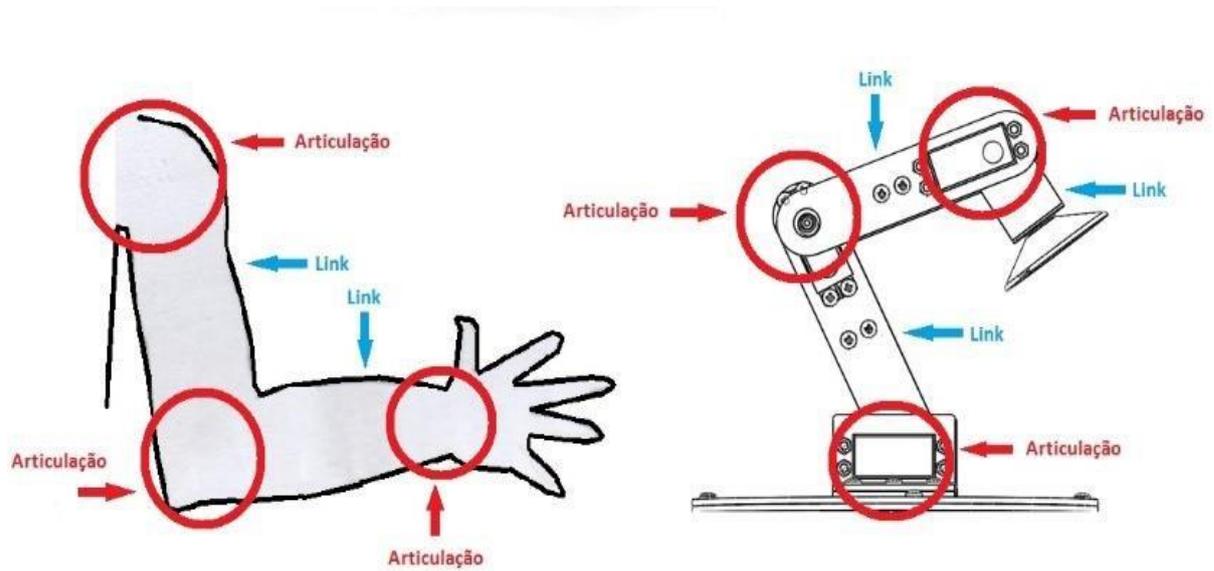
Fonte: WJ Componentes Eletrônicos

Com um chassi de esteira, não é necessário utilizar rodas direcionais para movimentos laterais ou curvas. Para realizar um deslocamento lateral, basta que uma esteira gire em uma direção e a outra no sentido oposto. Para fazer uma curva, é suficiente que uma das esteiras gire mais rápido que a outra. Isso simplifica a mobilidade e o controle do robô. A escolha do tipo de chassi dependerá das especificações e necessidades de cada projeto. (Cardoso,2016)

4.5 Braço Robótico

Os braços robóticos são compostos por ligamentos rígidos conectados por articulações, que permitem movimentos relativos entre eles. Os links de um braço robótico são as articulações que conectam os elos do braço e permitem que ele se mova em diversas posições. Essas articulações podem ser de rotação, que medem a posição angular, ou prismáticas, que seguem um movimento linear. O número de graus de liberdade de um robô manipulador corresponde ao número de variáveis de posição necessárias para controlar suas partes, geralmente igual ao número de articulações.

Figura 7: 1 -Comparação entre um braço humano e um robótico



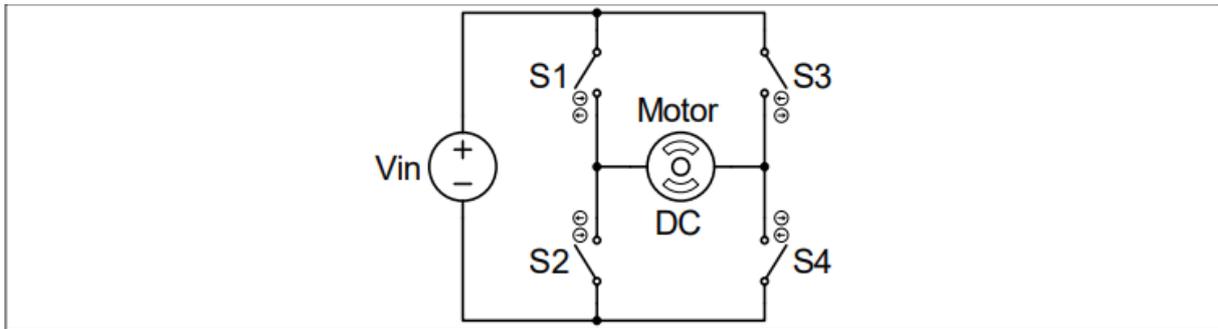
Fonte: Scientific Diagram

Além disso, mini-braços robóticos são usados para realizar tarefas pequenas e precisas, aproveitando esses conceitos, o que os torna ideais para trabalhos delicados e detalhados.

4.6 Ponte H

O circuito em questão, de acordo com Felipe, é projetado para converter uma fonte de corrente contínua fixa em tensão contínua ajustável, funcionando por abertura e fechamento cíclicos. Em outras palavras, é por meio desse circuito que a direção da corrente, sua polaridade e a amplitude da tensão que é aplicada a um sistema ou componente específico é definida. Funcionando por meio do chaveamento de componentes eletrônicos suportados pela técnica de modulação por largura de pulso, tal instrumento possibilita o controle preciso da polaridade e intensidade da tensão em um ponto específico do circuito, cuja ilustração é disposta na Figura(?). O circuito é comumente usado para controlar a velocidade e direção de motores de corrente contínua com escovas, bem como regular a saída de geradores CC ou utilizado como inversor monofásico. O nome de “Ponte H” deve-se à forma do circuito, visualmente apresentada na Figura (?). (Maimon,2004)

Figura 8: Circuito genérico ponte H.



Fonte: PROJETO DE UM BRAÇO ROBÓTICO UTILIZANDO ATUADORES PNEUMÁTICOS E ELÉTRICOS CONTROLADOS PELO SISTEMA EMBARCADO ARDUINO

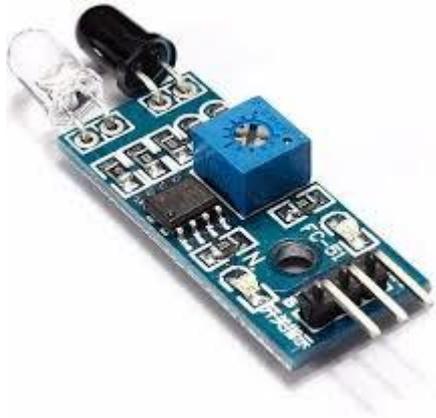
5 MATERIAIS E MÉTODOS

5.1 Materiais

5.1.2 Sensor de Obstáculos Reflexivo Infravermelho

O Sensor de Obstáculo Reflexivo Infravermelho utiliza um LED que emite luz infravermelha e um fotodiodo. Quando um objeto ou obstáculo passa dentro do ângulo de reflexão e da distância configurada, o sensor detecta essa presença e coloca a saída em nível lógico BAIXO. O módulo conta com um trimpot (É um tipo de potenciômetro miniatura ajustável, que é montado diretamente na placa de circuito impresso e ajustado por uma pequena chave.) para ajustar e calibrar a distância de detecção, que pode variar de 2 cm a 30 cm, dependendo do nível de reflexão do objeto. Mesmo com o ajuste feito, a distância de detecção pode sofrer variações conforme o tamanho e a cor do obstáculo. A tensão de operação do sensor vai de 3,3V a 5V, o que permite a conexão direta a uma porta de microcontrolador (como Arduino, PIC e outros). (Eletrogate,2024)

Figura 9: Sensor De Obstáculos Reflexivo Infravermelho



Fonte: Fermarc

5.1.2 Chassi

Após considerarmos alguns modelos de chassi, que fossem simples de se utilizar, e com um preço de custo não elevado, optou-se por utilizar o Chassi 4WD para formulação do projeto.

O chassi utilizado foi disponibilizado pela BR eletrônica, com dimensões de 25,6 x 14,9 cm, sendo de acrílico (polímero sintético, de baixo custo e fácil processamento com potencial para diversas aplicações, é leve e possui boa resistência) e contendo dois andares perfurados, permitindo um espaço extra tanto em cima quanto no meio para acomodar uma variedade de sensores e componentes eletrônicos, tais como os motores DC 3-6V com Redução, e as rodas. (huinfinito. 2024)

Figura 10: Chassi Duplo 4WD Rodas Robótica Carro Robô



Fonte: BR ELETRÔNICA,2024

5.1.3 BlackBoard UNO

Para o desenvolvimento deste trabalho, foi selecionada a placa BlackBoard UNO R3, sendo ela um Arduino UNO compatível fabricada pela RoboCore, projetada unindo o melhor das placas Arduino básicas lançadas até hoje, além de implementações e melhorias feitas pelos engenheiros da robocore.

De acordo com informações do fabricante (ROBOCORE), ela conta com um clock de 16 MHz, memória flash de 32 KB, além de 22 pinos de entrada e saída digitais, sendo que 6 deles podem ser utilizados como saídas PWM. A modulação por largura de pulso (PWM) permite o controle da potência ou da velocidade dos dados transmitidos através de uma onda quadrada. No Arduino, essas portas são identificadas pelo símbolo "~". Além disso, a placa dispõe de 6 entradas analógicas, suporte para conexão via cabo USB ou alimentação por fonte externa, e um botão de reset.

Sendo ela praticamente um Arduino Uno melhorado, é considerado igual a o Arduino Uno um dos modelos mais básicos da família Arduino, sendo ideal para iniciantes em projetos eletrônicos. No entanto, também se destaca como uma plataforma poderosa para prototipagem, oferecendo uma ampla variedade de funcionalidades. (Cardoso,2016)

Figura 11: BlackBoard UNO R3



Fonte: Robocore

5.1.4 Servomotor

5.1.4.1 Motor DC 3-6v

As características exigidas de um servomotor entre outras são: dinâmica, controle de rotação, torque constante e precisão de posicionamento. (INDRAMAT, 1997).

O Motor DC 3-6V com Caixa de Redução e Eixo Duplo é bastante versátil e ideal para projetos com Arduino e construção de Robôs, como por exemplo um Robô seguidor de linha. Seguir linha é uma das funções do nosso robô, e é para essa aplicação que o servomotor será utilizado. Algumas de suas características são a rotação de saída adequada para projetos de carros e robôs, bom torque graças à caixa de redução e tensão de operação flexível, e acompanha uma roda integrada ao motor.

Figura 12: Kit motor DC + roda para Robô.



Fonte: Baú da Eletrônica - Baú da Eletrônica - Componentes Eletrônicos

5.1.4.2 Servo Sg90

O Micro Servo SG90 é um servo motor compacto e leve, pesando apenas 9g, ideal para projetos de aeromodelismo, robótica e automação. Ele possui um torque de 1,8 kgf/cm a 4,8V e um ângulo de rotação de 180 graus, oferecendo flexibilidade e precisão. Funciona com tensão entre 4,8V e 6V e é controlado por sinal PWM, o que facilita a integração com Arduino e a Raspberry Pi (um conjunto de mini-computadores de placa única e outros microcontroladores). Vem com três tipos de braços e parafusos para montagem. Seu uso é comum em modelos RC, braços robóticos e outros sistemas que exigem movimento controlado. (Eletrogate,2024)

Figura 13: Micro Servo 9g SG90



Fonte: Eletrogate

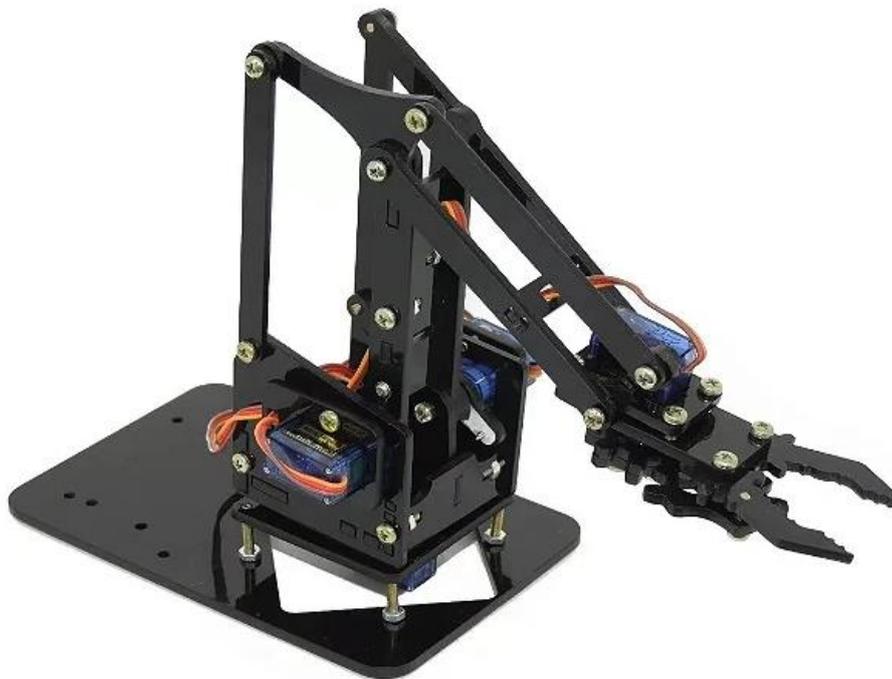
5.1.5 Garra

O Braço Robótico Acrílico foi projetado para ser operado com 4 servos motores TowerPro SG90, permitindo o controle preciso dos movimentos de suas articulações. O módulo responde de forma proporcional aos comandos, ajustando o giro e a posição do braço de acordo com as instruções enviadas.

O kit é enviado desmontado, mas contém todas as peças necessárias para a montagem, que é simples e intuitiva. Suas articulações oferecem uma ampla faixa de movimento de até 180°, o que proporciona flexibilidade para realizar diversos tipos de tarefas. (R.E.B,2024)

Além disso, o braço conta com uma garra que pode ser aberta até 55mm, permitindo pegar e manipular objetos de diferentes tamanhos. A garra também contribui para aumentar o grau de liberdade do braço robótico, possibilitando movimentos de apreensão, liberação e ajuste de itens com precisão. Essa combinação de articulações e garra proporciona um grau de liberdade eficaz, ideal para realizar tarefas mais complexas de manipulação e movimentação em projetos de robótica.

Figura 14: Braço Robótico em Acrílico com parafusos e Manual de Montagem para Arduino



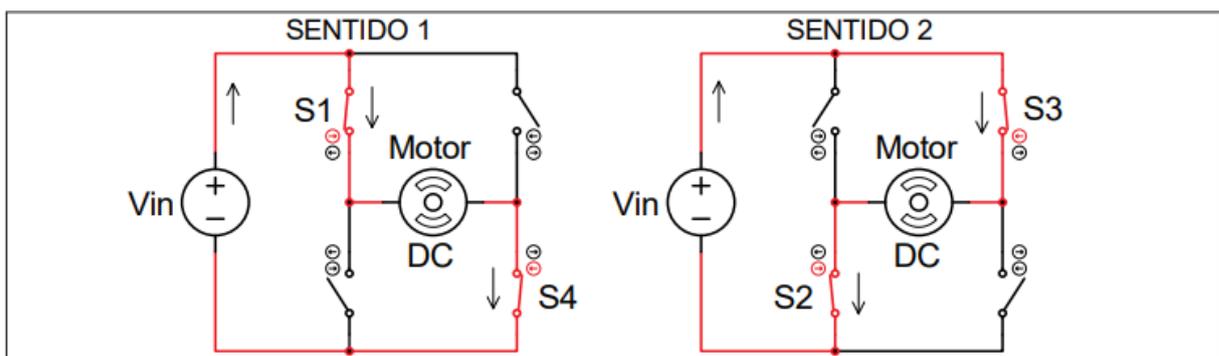
5.1.6 Módulo L298N Ponte H

Optamos pelo uso da ponte H L298N devido à sua facilidade de utilização e versatilidade em aplicações de controle de motores DC e motores de passo. O módulo L298N é um driver que implementa uma ponte H, sendo composto por dois canais que permitem controlar a direção e a velocidade de dois motores DC independentes ou um motor de passo, tudo através de um único chip.

O circuito de ponte H é utilizado para definir a direção da corrente e a tensão aplicadas no controle de um motor DC. A Figura (?) apresenta um diagrama genérico que ilustra seu funcionamento. Ao ativar simultaneamente as chaves S1 e S4, o terminal esquerdo do motor recebe uma tensão mais elevada em relação ao terminal direito, o que faz a corrente fluir da esquerda para a direita. Dessa forma, o motor passa a girar em uma direção conhecida como Sentido horário.

Por outro lado, quando as chaves S3 e S2 são ativadas juntas, o terminal direito do motor apresenta uma tensão mais alta do que o terminal esquerdo, invertendo o fluxo de corrente para a direção da direita para a esquerda. Com isso, o motor adquire um sentido de rotação inverso ao Sentido 1, chamado de Sentido anti-horário.

Figura 15: Circuito genérico de funcionamento ponte H



Fonte: PROJETO DE UM BRAÇO ROBÓTICO UTILIZANDO ATUADORES PNEUMÁTICOS E ELÉTRICOS CONTROLADOS PELO SISTEMA EMBARCADO ARDUINO

5.1.7 Especificações de componentes

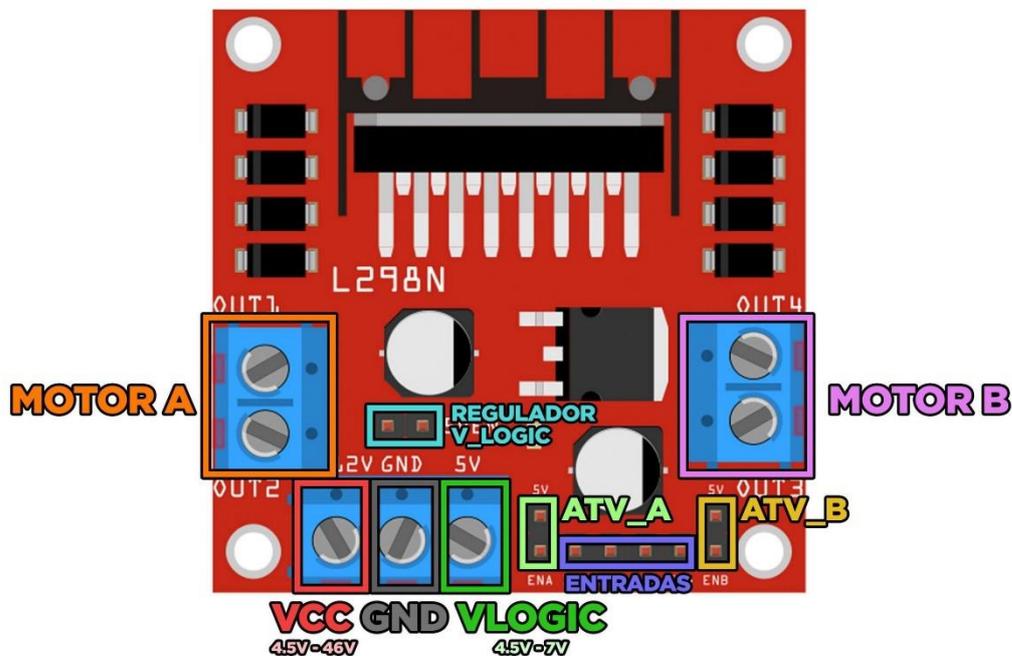
Neste tópico, iremos detalhar as especificações dos principais componentes utilizados: a ponte H L298N, o Arduino BlackBoard UNO e o Sensor de Obstáculos Reflexivo Infravermelho. Cada um desses elementos desempenha um papel essencial no funcionamento do projeto e será

descrito cuidadosamente para proporcionar uma visão clara de suas características e funcionalidades.

5.1.7.1 Detalhando Componente Ponte H L298N

O uso da ponte H L298N pode ser bastante confuso, devido à grande quantidade de informações pouco claras disponíveis na web, o que pode levar a erros de funcionamento. Abaixo, explicaremos em detalhes o funcionamento da placa e o modo correto de utilizá-la de modo geral.

Figura 16: Detalhando da Ponte H L298N



Fonte: eletrogate

Vcc: Este borne alimenta os motores com a tensão aplicada nele. Certifique-se de que essa tensão não exceda a especificação dos motores (ex: máximo de 12V para motores de 12V). A placa possui um regulador de tensão 78m05 com limite seguro de até 20V; caso a alimentação Vcc esteja acima de 12V, não use a porta Vlogic para alimentar outros circuitos. Para tensões acima de 20V, desative o regulador de tensão e alimente a Vlogic diretamente.

GND: Conecte este borne ao GND do Arduino ou do microcontrolador, garantindo que todos os GNDs estejam conectados entre si.

Vlogic: Este borne fornece a alimentação lógica do chip L298N, necessária para receber comandos do Arduino. Se o regulador de tensão estiver ativo, ele regula Vlogic a 5V. É possível alimentar o Vlogic manualmente (desativando o regulador), mas se o regulador estiver ativo, nunca conecte alimentação externa para evitar danos.

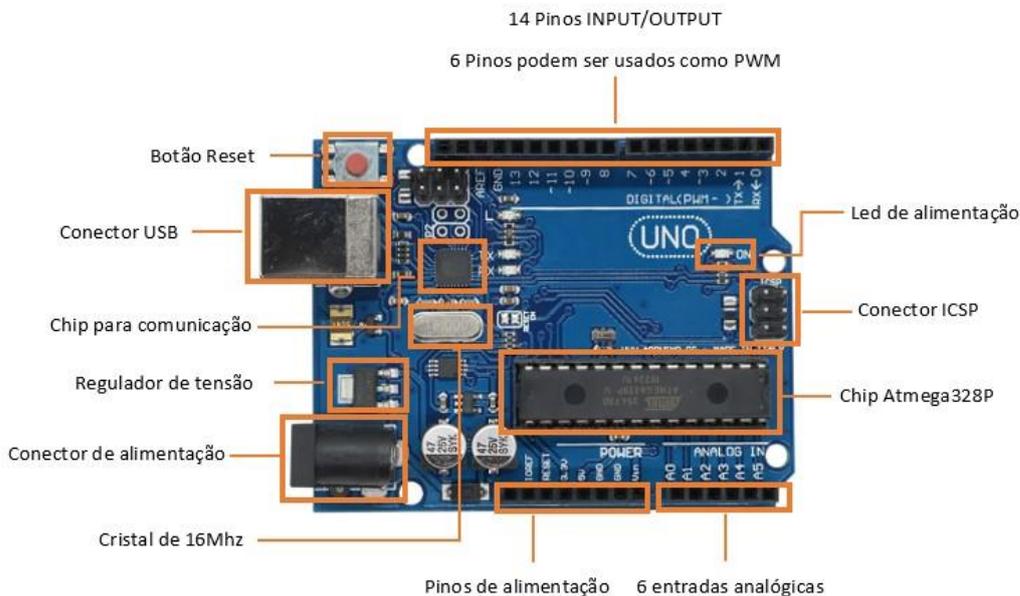
Regulador Vlogic (Jumper): Este jumper ativa ou desativa o regulador de tensão de Vlogic. Com ele ativado, Vlogic é regulada em 5V a partir da tensão de Vcc.

Atv_A e Atv_B: Estes jumpers controlam a ativação dos motores A e B, respectivamente. Quando conectados, permitem que o Arduino ou microcontrolador controle o motor; quando desconectados, o motor fica desativado.

5.1.7.2 Detalhando Componente BlackBoard UNO

Como o Arduino BlackBoard UNO compartilha o mesmo design do Arduino UNO R3, utilizaremos uma imagem do Arduino UNO como referência para ilustrar todas as funcionalidades do BlackBoard UNO. Dessa forma, será mais fácil compreender cada uma de suas utilidades, já que ambos possuem estrutura e configuração de pinos praticamente idênticas.

Figura 17: CARACTERÍSTICAS PRINCIPAIS DO ARDUINO UNO



Fonte: sta-eletronica

Fonte de Alimentação: Pode ser alimentado via USB ou uma fonte externa (7-12V) inserida no conector de alimentação, ou por meio dos pinos Vin e GND.

Entradas e Saídas Digitais: Possui 14 pinos configuráveis como entrada ou saída, usando funções como `pinMode()`, `digitalWrite()`, e `digitalRead()`.

Entradas Analógicas: Conta com 6 canais analógicos (A0 a A5) com resolução de 10 bits, mapeando tensões entre 0 e 5V para valores de 0 a 1023, lidos com a função `analogRead()`.

Pinos Serial (TX e RX): Pinos 0 (RX) e 1 (TX) para transmissão de dados seriais TTL, conectados ao chip USB-Serial ATmega8U2.

Pinos de Interrupção Externa: Pinos 2 e 3 funcionam como entradas para interrupções externas, designadas como INT0 e INT1.

Pinos PWM: Possui 6 pinos PWM (3, 5, 6, 9, 10, 11) para controle de intensidade com função de duty cycle.

Pinos SPI: Inclui pinos SS (10), MOSI (11), MISO (12), e SCK (13) para comunicação serial SPI full duplex.

LED Interno (Pino 13): Um LED embutido no pino 13 acende quando o pino está em nível ALTO.

Pinos I2C (SDA e SCL): Pinos 4 (SDA) e 5 (SCL) para comunicação I2C usando a biblioteca `Wire`.

AREF (Tensão de Referência): Permite configurar a tensão de referência para entradas analógicas, importante para leituras de precisão.

Pino de Reset: Usado para resetar o microcontrolador.

Memória: O Atmega328 possui 32 KB de flash (para código), 2 KB de SRAM, e 1 KB de EEPROM.

Comunicação: Suporta comunicação serial TTL (UART), com LEDs RX e TX para monitoramento, e usa a biblioteca `Software Serial` para comunicação nos pinos digitais. Suporta também I2C e SPI.

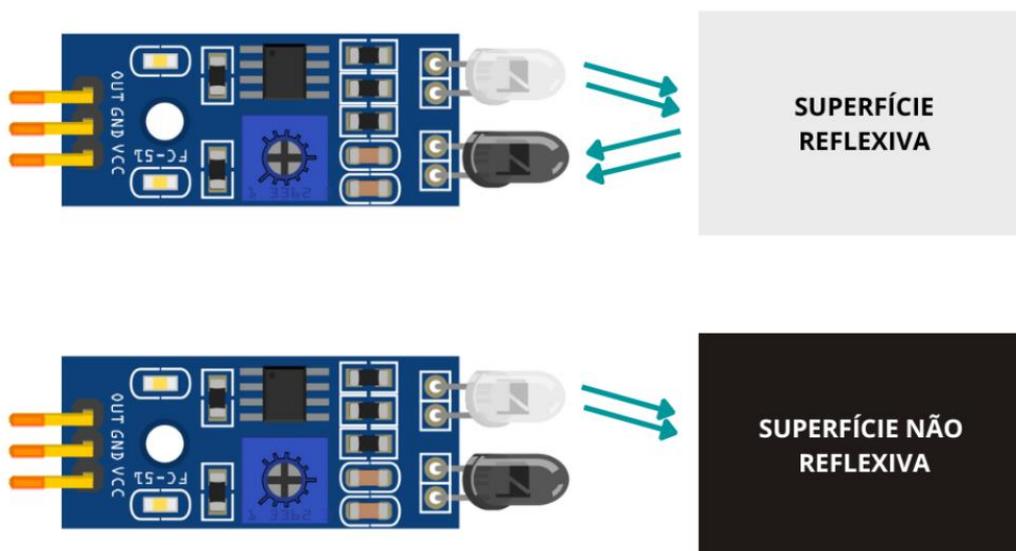
Proteção de Alta Tensão no USB: Possui um fusível para proteger a porta USB de sobretensões, adicionando segurança para o PC.

5.1.7.3 Detalhando Componente Sensor de Obstáculos Reflexivo Infravermelho.

Neste tópico, abordaremos o Sensor de Obstáculos Reflexivo Infravermelho, um componente essencial para detecção de objetos e obstáculos. Explicaremos seu funcionamento detalhado,

destacando como ele utiliza a tecnologia infravermelha para detectar a presença ou ausência de objetos em seu caminho.

Figura 18: Funcionamento do sensor de obstáculo reflexivo infravermelho IR em diferentes superfícies.

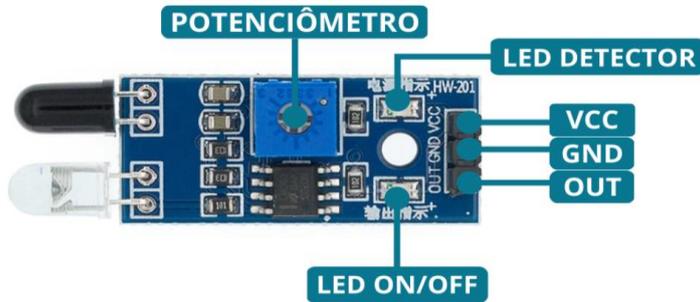


Fonte: Blog da Robótica

O sensor de obstáculo reflexivo IR é formado por dois LEDs: um LED emissor de infravermelho luz, que é transparente, emitindo uma luz em uma frequência completamente invisível ao olho humano e um foto transistor receptor, que é preto, e recebe este sinal de luz enviado. (Blog da Robótica,2023)

Estes dois funcionam em conjunto, quando o sinal atinge um objeto próximo, ele é refletido, e, quando isso acontece, o foto transistor o enxerga. Também pode haver alterações na resposta do sensor de acordo com a cor e material do objeto. Por exemplo, uma cor branca de um objeto é um excelente refletor de luz, um preto é um péssimo. (Blog da Robótica,2023)

Figura 19: Pinos e componentes do sensor de obstáculo reflexivo infravermelho IR.



Fonte: blogdarobotica

Os pinos e componentes do sensor de obstáculo reflexivo infravermelho IR possuem as seguintes funções:

VCC: Pino de alimentação do sensor com tensão entre 3,3V ~ 5V;

GND: Pino de aterramento do sensor;

OUT: Pino de saída digital do sensor;

LED ON/OFF: LED que indica quando o sensor está ligado (ON) ou desligado (OFF);

LED DETECTOR: LED que indica quando ocorre a detecção de um objeto pelo sensor;

POTENCIÔMETRO: Potenciômetro de ajuste da sensibilidade do sensor.

5.2 Métodos

5.2.1 Montagem do chassi

Figura 20: Montagem do chassi



Fonte: Autoria própria

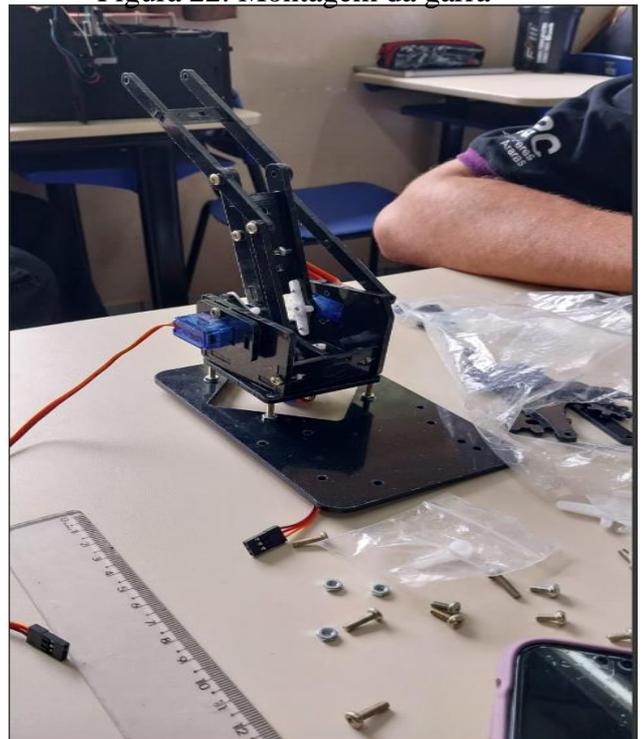
A montagem do chassi foi feita em nossa escola, com cuidado e precisão. No laboratório do curso de mecatrônica, fixamos as duas bases de acrílico com parafusos e porcas, de modo que elas garantam um suporte forte para o projeto. As rodas foram conectadas da mesma maneira, visando ao equilíbrio do projeto do robô. Todos os elementos foram pensados para a funcionalidade e resistência do projeto.

5.2.2 Montagem da garra

Figura 21: Peças da garra



Figura 22: Montagem da garra



Fonte: Autoria própria

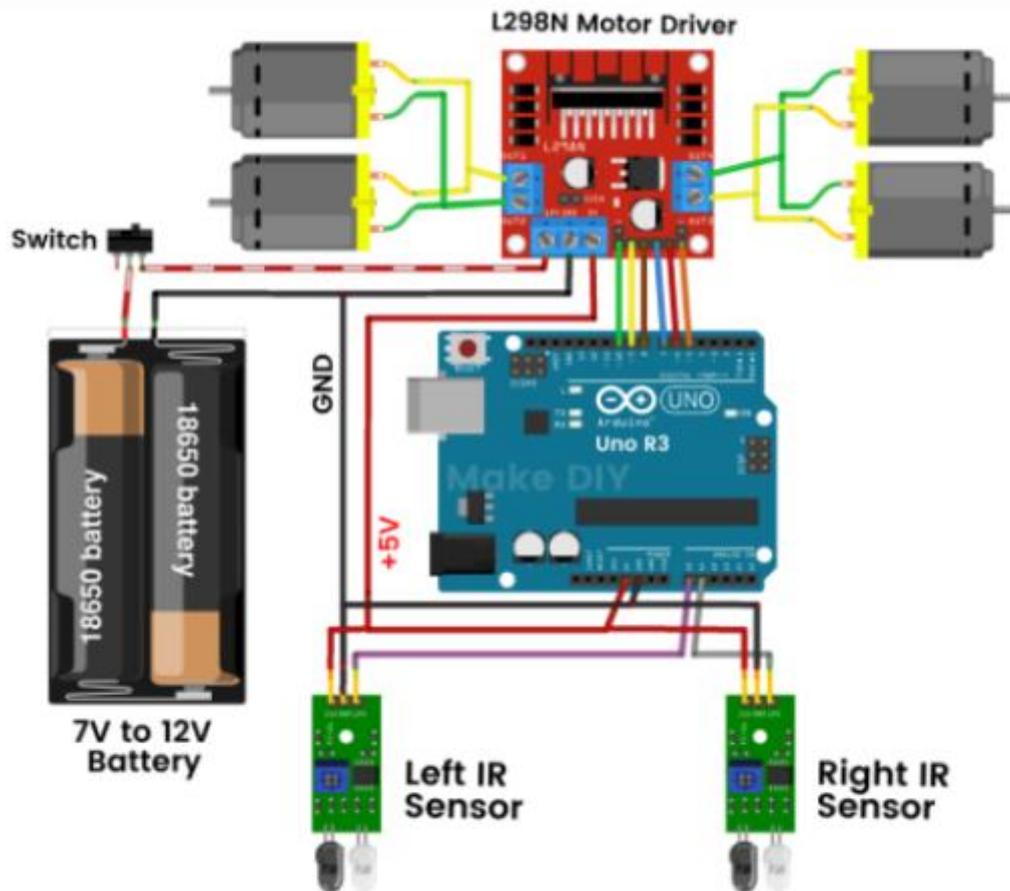
Fonte: Autoria Própria

A montagem de um mini-braço robótico de acrílico envolve a fixação de quatro servo motores SG90 em posições estratégicas: um na base para giro em 360 graus, outro no meio para inclinação do braço principal, um terceiro para levantar ou abaixar a garra, e o último para abrir e fechar a garra.

O braço é montado sobre uma base estável de acrílico, conectada ao chassi do robô. A estrutura do robô deve suportar o peso do braço e acomodar a eletrônica necessária. Combinado com o sistema segue-linha, o robô pode seguir trajetórias e manipular objetos.

5.2.3 Sistema elétrico do robô segue linha

Figura 23: Line following robot



Fonte: Autoria Própria

Este esquema representa um robô seguidor de linha com Arduino Uno, sensores infravermelhos (IR), driver de motor L298N, e motores DC.

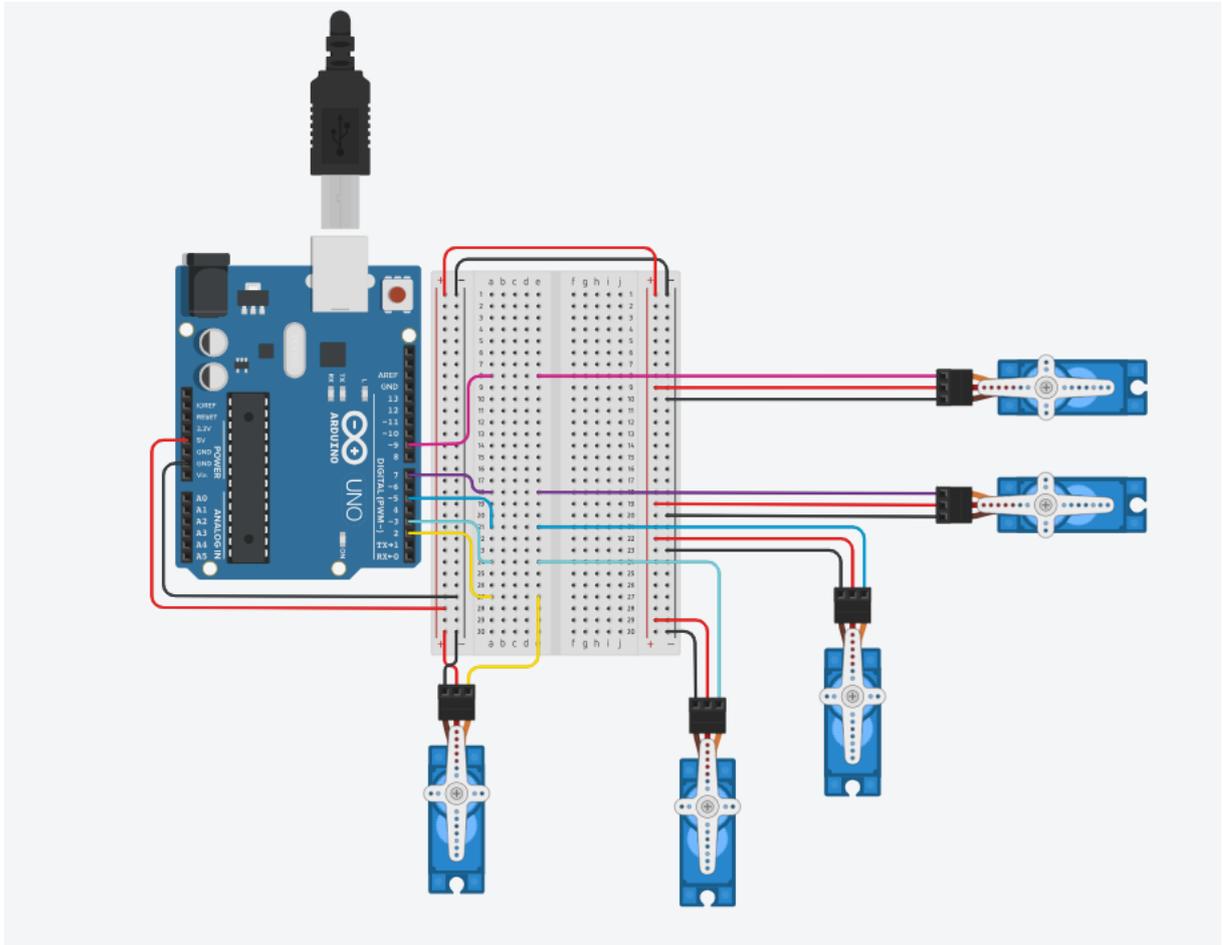
- **Bateria:** Alimenta o circuito e os motores através do L298N.
- **Arduino Uno:** Processa os sinais dos sensores e controla o driver de motor.
- **Sensores IR:** Detectam a linha no chão (preta ou branca) e enviam sinais ao Arduino.
- **Driver L298N:** Controla a direção e velocidade dos motores DC com base nos comandos do Arduino.
- **Motores DC:** Movem o robô conforme os comandos do L298N.

Funcionamento:

Os sensores IR detectam a linha e enviam sinais ao Arduino. Este processa os dados e comanda o L298N para ajustar os motores, direcionando o robô para seguir a linha.

5.2.4 Sistema elétrico da Garra Robótica

Figura 24: Sistema Elétrico Garra



Fonte: afonsofarias

Este sistema apresenta a eletrônica básica para o controle de servomotores, frequentemente utilizado em garras robóticas controladas por um Arduino Uno. A montagem inclui componentes essenciais como o Arduino Uno, que atua como a placa microcontroladora responsável por coordenar o controle dos servomotores, e uma protoboard, que facilita as conexões entre o Arduino, os servomotores e a alimentação elétrica. Os servomotores, elementos principais da garra robótica, são utilizados para realizar os movimentos necessários, com cada motor controlando uma articulação ou eixo específico da garra. O sistema também conta com um cabo de alimentação USB, que conecta o Arduino ao computador para fornecer energia e permitir a programação.

As conexões incluem fios de alimentação (VCC e GND), onde os fios vermelhos são ligados ao barramento de alimentação (+5V) na protoboard, enquanto os fios pretos conectam os GNDs dos servomotores ao Arduino, completando o circuito elétrico. Cada servomotor possui três fios: o VCC (vermelho), o GND (preto ou marrom) e o Sinal (geralmente laranja ou amarelo). Esses fios são conectados aos pinos digitais do Arduino, como os pinos D3, D5, D6 e D9. O Arduino envia sinais PWM (modulação por largura de pulso) através desses pinos digitais para controlar o ângulo de rotação dos servomotores, permitindo que eles movimentem as articulações da garra robótica, ajustando os movimentos de abrir, fechar ou reposicionar as partes da garra.

O funcionamento geral do sistema é controlado por um código escrito na IDE do Arduino, frequentemente utilizando a biblioteca Servo.h para simplificar o gerenciamento dos motores. O Arduino ajusta os ângulos de cada servomotor com base nos comandos programados ou recebidos, que podem vir de fontes como joysticks, sensores ou softwares externos. A alimentação dos servomotores é fornecida diretamente pelo barramento da protoboard, que recebe energia do pino de 5V do Arduino. Este sistema permite a movimentação precisa da garra robótica, coordenada por um código programado no Arduino.

5.2.5 Programação do Robô segue Linha

Definições e Configurações Iniciais

Definição de pinos:

Figura 25: Definição de pinos para o motor.

```
#define in1 9
#define in2 8
#define in3 7
#define in4 6
#define enA 10
#define enB 5
```

Fonte: A autoria Própria

Esses pinos controlam os motores:

in1 e in2: Controlam a direção do motor 1 (esquerdo).

in3 e in4: Controlam a direção do motor 2 (direito).

enA e enB: Controlam a velocidade dos motores (PWM).

Figura 26: Variáveis de velocidade:

```
int M1_Speed = 80; // Velocidade do motor 1
int M2_Speed = 80; // Velocidade do motor 2
int LeftRotationSpeed = 250; // Velocidade ao virar para a esquerda
int RightRotationSpeed = 250; // Velocidade ao virar para a direita
```

Fonte: Autoria Própria

Figura 27: Configuração dos pinos no setup

```
void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(A0, INPUT); // Sensor esquerdo como entrada
  pinMode(A1, INPUT); // Sensor direito como entrada
}
```

Fonte: Autoria Própria

Aqui, os pinos para os motores e sensores são configurados como saídas e entradas, respectivamente.

Figura 28: Leitura dos sensores

```
int LEFT_SENSOR = digitalRead(A0); // Lê o sensor esquerdo
int RIGHT_SENSOR = digitalRead(A1); // Lê o sensor direito
```

Fonte: Autoria Própria

Figura 29: Decisão de movimento

```
if (RIGHT_SENSOR == 0 && LEFT_SENSOR == 0) {
    forward(); // Ambos sensores não detectam a linha -> Avançar
} else if (RIGHT_SENSOR == 0 && LEFT_SENSOR == 1) {
    right(); // Sensor esquerdo detecta a linha -> Virar à direita
} else if (RIGHT_SENSOR == 1 && LEFT_SENSOR == 0) {
    left(); // Sensor direito detecta a linha -> Virar à esquerda
} else if (RIGHT_SENSOR == 1 && LEFT_SENSOR == 1) {
    Stop(); // Ambos detectam a linha -> Parar
}
```

Fonte: Autoria Própria

Decisão de movimento com base nos valores dos sensores:

- **0**: Linha **não** detectada.
- **1**: Linha detectada.

Figura 30: Motor esquerdo e direito giram para frente.

```
void forward() {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enA, M1_Speed);
    analogWrite(enB, M2_Speed);
}
```

Fonte: Autoria Própria

Avançar = (forward)

Figura 31: Ambos os motores giram para trás.

```
void backward() {  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, HIGH);  
    analogWrite(enA, M1_Speed);  
    analogWrite(enB, M2_Speed);  
}
```

Fonte: A autoria Própria

Figura 32: Motor esquerdo gira para frente e o direito para trás, fazendo o robô virar à direita.

```
void right() {  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
    analogWrite(enA, LeftRotationSpeed);  
    analogWrite(enB, RightRotationSpeed);  
}
```

Fonte: A autoria Própria

Figura 33: Motor direito gira para frente e o esquerdo para trás.

```
void left() {  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, HIGH);  
    analogWrite(enA, LeftRotationSpeed);  
    analogWrite(enB, RightRotationSpeed);  
}
```

Fonte: A autoria Própria.

Figura 34: Ambos os motores são desligados.

```
void Stop() {  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, LOW);  
}
```

Fonte: Autoria Própria

Resumo do Funcionamento

1. Os **sensores IR** detectam a linha (ou ausência dela).
2. O **Arduino** processa os sinais dos sensores e decide o movimento:
 - Avançar, virar à esquerda, à direita, ou parar.
3. O **L298N** executa os comandos do Arduino, controlando a direção e a velocidade dos motores DC.

Com isso, o robô ajusta sua trajetória para seguir a linha detectada pelos sensores.

5.2.6 Programação da Garra Robótica

Figura 35: Programação da Garra

```
1 #include <Servo.h>
2
3   Servo ombro;// Cria um objeto servo
4   Servo cotovelo;// Cria um objeto servo
5   Servo pulso;// Cria um objeto servo
6   Servo indicador;// Cria um objeto servo
7   Servo polegar; // Cria um objeto servo
8
9
10 void setup () {
11
12   ombro.attach(2); //define o pino conectado no arduino
13   cotovelo.attach(3);//define o pino conectado no arduino
14   pulso.attach(5);//define o pino conectado no arduino
15   indicador.attach(9);//define o pino conectado no arduino
16   polegar.attach(7);//define o pino conectado no arduino
17
18 }
19
20 void loop () {
21   ////////////////////////////////// MOVIMENTO POSIÇÃO INICIAL //////////////////////////////////
22
23   ombro.write(160);
24   delay(1000);
25
26   cotovelo.write(120);
27   delay(1000);
28
29   pulso.write(45);
30   delay(1000);
31
32   indicador.write(180);
```

Fonte: Autoria Própria

Figura 36: Programação da Garra Continuação.

```
33     polegar.write(0);
34         delay(1000);
35
36     //////////////// MOVIMENTO ALONGAMENTO ////////////////
37
38     polegar.write(40);
39     indicador.write(140);
40         delay(1000);
41
42     pulso.write(0);
43         delay(1000);
44
45     cotovelo.write(0);
46         delay(1000);
47
48     ombro.write(0);
49         delay(1000);
50
51 }
52
```

Fonte: Autoria Própria

O código controla uma garra robótica com cinco servomotores conectados a um Arduino, representando articulações como ombro, cotovelo, pulso, indicador e polegar. No **setup**, cada motor é associado a um pino do Arduino usando o método `attach` (O método `attach` () da biblioteca **Servo.h** conecta um servomotor a um pino do Arduino, permitindo que ele receba sinais PWM para controlar seu ângulo. Por exemplo, `ombro.attach(2)`; associa o motor ao pino digital 2 para movimentá-lo.). No **loop**, o programa define dois movimentos:

1. **Posição Inicial:** Define ângulos específicos para cada motor, posicionando a garra em uma configuração inicial com comandos como `write` () e pausas (`delay` ()).
2. **Movimento de Alongamento:** Move os motores gradualmente para esticar a garra, alterando os ângulos de cada articulação em sequência, com pausas entre os ajustes.

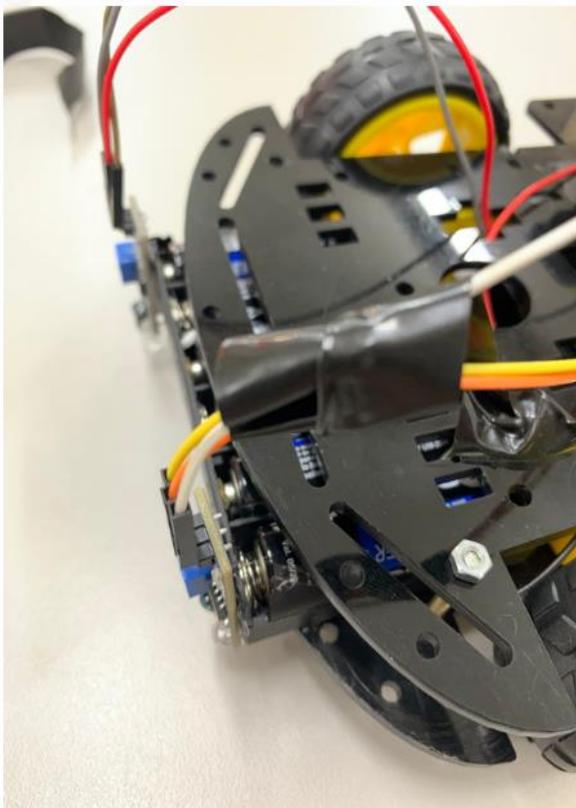
Os comandos `write` () ajustam o ângulo de cada motor (entre 0° e 180°), permitindo movimentos coordenados.

RESULTADOS E DISCUSSÃO

O desenvolvimento do carro robótico seguidor de linha envolveu uma série de desafios técnicos, mas foi uma experiência valiosa para aprimorarmos nossas habilidades em robótica e programação. Durante a execução do projeto, tivemos que lidar com diversos problemas, que foram resolvidos pelo método de tentativa e erro.

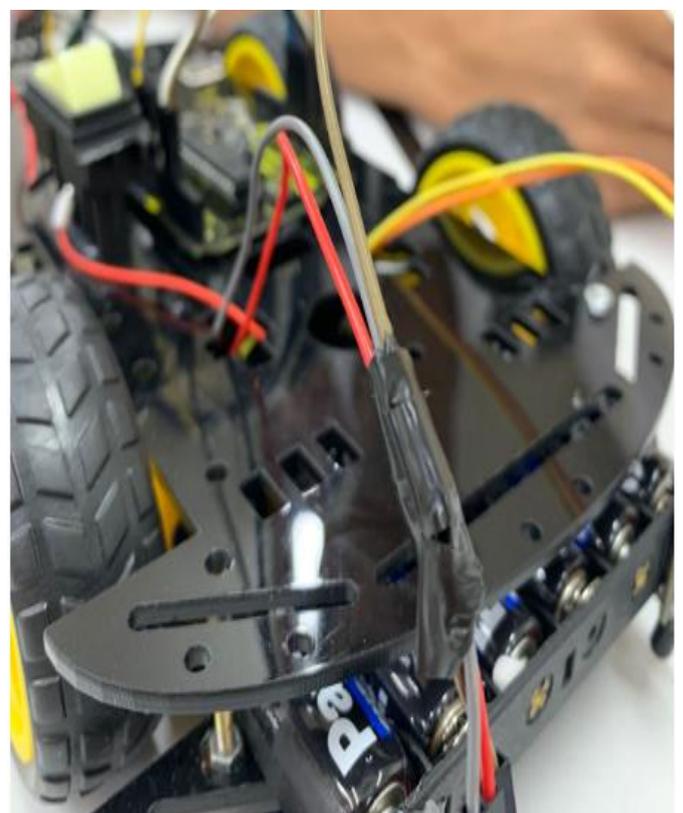
O primeiro desafio surgiu no sistema elétrico, onde a bateria estava descarregando muito rapidamente devido a uma falha em uma emenda, que causou o vazamento de energia. Esse problema gerou instabilidade no funcionamento do robô. Após vários testes de diagnóstico e manutenção, conseguimos resolver o problema, ajustando a parte elétrica e garantindo que a energia fosse distribuída adequadamente.

Figura 37: Emenda feita errada.



Fonte: Autoria Própria

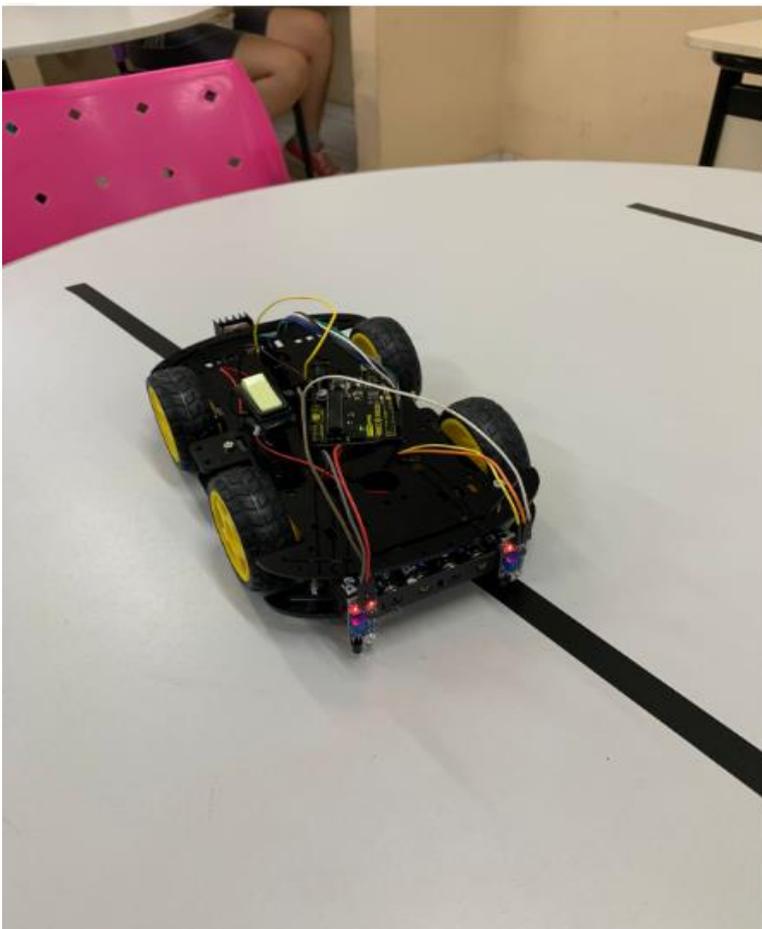
Figura 38: Emenda certa



Fonte: Autoria Própria

Outro problema importante ocorreu na programação dos motores. Inicialmente, os motores giravam em sentidos opostos ao que era necessário, o que impedia o robô de seguir a linha corretamente. A solução foi revisar e ajustar a programação dos motores para que eles girassem na direção certa. Após esse ajuste, o robô a seguiu a linha com mais precisão, o que representou um avanço importante

Figura 39: Carro Andando



Fonte: Autoria Própria

Além desses problemas, a garra embutida no robô também apresentou dificuldades. A fixação dos parafusos da garra foi um desafio, pois não conseguiu apertá-los de maneira ideal, comprometendo a estabilidade do componente. Além disso, a programação da garra não

respondeu como esperado. O servo motor não executava os movimentos previstos ou impedia a garra de funcionar corretamente. Tentamos ajustar a programação, mas não conseguimos resolver completamente esse problema no prazo determinado.

Apesar dessas dificuldades, o protótipo final do robô foi capaz de cumprir a função principal de seguir uma linha de forma eficaz. Embora a garra ainda apresente falhas, os problemas encontrados durante o desenvolvimento foram essenciais para o aprendizado sobre os desafios da robótica, elétrica e eletrônica, usando assim muitos dos conhecimentos do nosso curso de mecatrônica.

Em resumo, embora o projeto apresentado tenha obstáculos técnicos significativos, conseguimos resolvê-los de forma gradual, ajustando tanto o hardware quanto o software. A experiência adquirida ao longo do processo de resolução de problemas e a interação entre os componentes do robô será fundamental para nossos futuros.

Figura 40:Robô com a garra.



Fonte: Autoria Própria.

CONCLUSÃO

O desenvolvimento deste projeto de um robô seguidor de linha com garra robótica embutida teve como objetivo principal integrar dois sistemas: a entrega autônoma do robô e a capacidade de manipulação através de uma garra.

A primeira parte do projeto, que envolveu o desenvolvimento do carro seguidor de linha, foi bem-sucedida. Após ajustes na programação e no sistema elétrico, conseguimos garantir que o robô seguisse a linha com precisão. A programação dos sensores, motores e o ajuste da estrutura

elétrica permitiram que o robô realizasse uma tarefa de maneira eficaz, atendendo ao objetivo inicial de desenvolver um sistema independente para movimento.

Porém, a integração da garra robótica no protótipo apresentou desafios inovadores. Embora a ideia fosse que a garra estivesse embutida no carro, ela não funcionou como esperado. Problemas de abertura e fixação dos componentes da garra dificultaram sua instalação e comprometeram a estabilidade. Além disso, a programação do servo motor que controlava os movimentos da garra não respondia de forma adequada, impedindo a execução das ações desejadas.

REFERENCIAS

Associação Brasileira de Automação – GS1 Brasil. **Índice de Automação – Mercado Brasileiro**. Disponível em: < <https://www.gs1br.org/indices-e-pesquisas/indices/Documents/%C3%8Dndice%20de%20Automa%C3%A7%C3%A3o%20do%20Mercado%20Brasileiro%20-%20Outubro%202020.pdf>> Acesso em: 25/08/2024

Bruno, M. D. C. C. **Desenvolvimento De Um Robô Móvel Com Trajetória Programável Utilizando A Plataforma Arduino**. 2016. Trabalho De Conclusão De Curso- Bacharel em Ciência da Computação. Ponta Grossa-2016.

ASSOCIAÇÃO BRASILEIRA DE RESÍDUOS E MEIO AMBIENTE. **Panorama Dos Resíduos Sólidos No Brasil**. 2023.

MCROBERTS, M. **Arduino básico**. 1. ed. São Paulo: Novatec Editora, 2011.

Arduino.cc. **O que é Arduino?** 2018. Disponível em:

<<https://www.arduino.cc/en/Guide/Introduction> >. Acesso em: 18 agosto 2024

KALATEC AUTOMOÇÃO. **Servo Motor: o que é um, como funciona e quais as vantagens?**. 2024. Disponível em: <<https://blog.kalatec.com.br/o-que-e-servo-motor/>>.

Acesso em: 14 setembro 2024

CABRAL. **Robôs Industriais**. 1. ed. São Paulo: USP. 11 p.

Manual Técnico. INDRAMAT GmbH Bgm. -Dr.-Nebel-Str.2. D-97816 Lohr Dept.

ECD(RR/JR) 1997 ed 1.

MAIMON, Felipe. Projeto de um Sistema Eletrônico para o Controle de Motores de Alta Potência por PWM. PUC Rio. 2004.

ELETROGATE. **Sensor de Obstáculos Reflexivo Infravermelho**. 2024. Disponível em:

<[https://www.eletrogate.com/sensor-de-obstaculo-reflexivo-](https://www.eletrogate.com/sensor-de-obstaculo-reflexivo-infravermelho#:~:text=O%20M%C3%B3dulo%20Sensor%20de%20Obst%C3%A1culo,sa%C3%ADda%20em%20n%C3%ADvel%20l%C3%B3gico%20BAIXO.)

[infravermelho#:~:text=O%20M%C3%B3dulo%20Sensor%20de%20Obst%C3%A1culo,sa%C3%ADda%20em%20n%C3%ADvel%20l%C3%B3gico%20BAIXO.](https://www.eletrogate.com/sensor-de-obstaculo-reflexivo-infravermelho#:~:text=O%20M%C3%B3dulo%20Sensor%20de%20Obst%C3%A1culo,sa%C3%ADda%20em%20n%C3%ADvel%20l%C3%B3gico%20BAIXO.)>. Acesso em:

15/10/2024

ELETROGATE. **Guia Definitivo de uso da Ponte H L298N**. 2020. Disponível em: <

<https://blog.eletrogate.com/guia-definitivo-de-uso-da-ponte-h-l298n/>>. Acesso em:

15/10/2024

Blog da Robótica. **Como utilizar o sensor de obstáculo reflexivo infravermelho IR com**

Arduino. 2023. Disponível em: <<https://www.blogdarobotica.com/2023/04/18/como-utilizar-o-sensor-de-obstaculo-reflexivo-infravermelho-ir-com-arduino/>>. Acesso em: 20/10/2024

Fábio Souza. Placa Arduino da Robocore – BlackBoard. 2014. Disponível em: <

<https://embarcados.com.br/placa-arduino-da-robocore-blackboard/>>. Acesso em: 28/10/2024

Rontek. **CARACTERÍSTICAS PRINCIPAIS DO ARDUINO UNO**. 2024. Disponível em:
< <https://www.sta-eletronica.com.br/artigos/arduinos/caracteristicas-principais-do-arduino-uno>>.
Acesso em: 26/10/2024

ELETROGATE. **Micro Servo 9g SG90**. 2024. Disponível em: <
<https://www.eletrogate.com/micro-servo-9g-sg90-towerpro>>. Acesso em: 09/11/2024

Robótica educacional Brasil. **Braço Robótico em Acrílico com parafusos e Manual de Montagem para Arduino**. 2024. Disponível em: <
[https://www.roboticaeducacional.art.br/braco-robotico-em-acrilico-com-parafuso-e-manual-de-montagem-para-arduino#:~:text=%E2%9C%94%20Dimens%C3%B5es%20do%20bra%C3%A7o%20\(CxLxA,Arduino%20ou%20o%20Raspberry%20Pi.>](https://www.roboticaeducacional.art.br/braco-robotico-em-acrilico-com-parafuso-e-manual-de-montagem-para-arduino#:~:text=%E2%9C%94%20Dimens%C3%B5es%20do%20bra%C3%A7o%20(CxLxA,Arduino%20ou%20o%20Raspberry%20Pi.>)>. Acesso em: 09/11/2024