

Ministério da Educação - Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina
Especialização em Automação Industrial

Unidade Curricular: Especialização em Automação Industrial

Trabalho de Conclusão de Curso (TCC)

IDENTIFICAÇÃO
Ministério da Educação - Secretaria de Educação Profissional e Tecnológica Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina
Especialização em Automação Industrial
Unidade Curricular: TCC - Artigo
Polo: Campus Lages/SC
Professor orientador: Rogério da Silva
Autor(a): Rudson João Bitencourte
Título e Subtítulo: Aplicação da Robótica Auxiliando Deficientes Visuais

Lages -SC
2024

1. INTRODUÇÃO

Monteiro, Silva e Lopes (2014) observam que no dia a dia é possível perceber a convivência dos deficientes visuais com desafios para acessar o mundo ao seu redor. Tarefas rotineiras, como caminhar ou correr, tornam-se complicadas para eles.

De acordo com Nunes e Lomonaco (2010), a cegueira é uma deficiência visual que restringe a percepção do mundo visível, ou seja, da visão. Assim, existem duas formas de deficiência visual: cegueira e baixa visão.

Amiralian (1997) apresenta dados dos anos 1970 que conceituam a deficiência visual em termos de acuidade visual para compreender como o indivíduo entende o ambiente ao seu redor. Se uma pessoa compreende o que está acontecendo através do toque, cheiro, percepção de movimento, peso, resistência e estímulos, ela é considerada cega. Por outro lado, se uma pessoa tem alguma deficiência visual, mas ainda consegue usar o restante de sua visão de forma eficaz, é classificada como tendo baixa visão.

O uso de acessórios ópticos (como lupas, óculos binoculares e monculares, lentes de contato) pode melhorar a visão, e medidas de apoio (como o sistema de escrita tátil, como o Braille ou a escrita comum) podem ser úteis. Em casos de baixa visão, auxílios ópticos podem ser empregados para melhorar a visão. No entanto, para indivíduos com a deficiência visual, é necessário que as imagens visuais sejam transmitidas por meio de outros sentidos, como o tato e a audição.

Os deficientes visuais enfrentam diversos obstáculos de deslocamento, como desníveis nas calçadas e ruas, obstáculos físicos como árvores, bancos de praça, pessoas, postes, e algumas pessoas que estacionam veículos em calçadas.

A limitação enfrentada pelos deficientes visuais não é causada apenas pela ausência da visão, mas também pela falta de oportunidades para experimentar diversas situações. Há décadas, diversas pesquisas têm incorporado novas tecnologias que melhoram a vida dos deficientes visuais, facilitando a locomoção, proteção e aquisição de conhecimento através da educação, entre outros recursos. A robótica, mecânica, eletrônica/ elétrica e a ciência da computação estão cada vez

mais presentes na vida dos cegos, auxiliando tanto em objetivos terapêuticos quanto de aprendizagem, tanto no progresso quanto na reabilitação, conforme Burlamaqui et al. (2017).

Bersch (2008) observa que no Brasil e em muitos outros países existem várias formas de tecnologia assistiva, desenvolvidas e ampliadas por diversas empresas e institutos de pesquisa com o intuito de ajudar essas pessoas na execução de suas atividades diárias. Entre as inovações mais usadas atualmente estão dispositivos de reprodução em braile, ampliadores e leitores de tela de computadores, bengalas que detectam obstáculos, passarelas com caminhos exclusivos para a locomoção de deficientes visuais em calçadas ou ruas/avenidas, e aparelhos de controle de ambiente. O objetivo principal dessas tecnologias é garantir aos deficientes visuais mobilidade e acesso.

Em ambientes fechados, a falta de conhecimento sobre a disposição de móveis, objetos e paredes é um desafio, assim como em locais movimentados onde é difícil prever colisões.

1.1. PROBLEMA

Os deficientes visuais se deparam com vários obstáculos no seu dia a dia, até em uma simples caminhada.

Sendo assim, essa pesquisa trata-se de uma aplicação da robótica para auxiliar no deslocamento diário dessas pessoas.

Esse experimento foi aplicado anteriormente na graduação de Engenharia Elétrica do autor Bitencourte (2018) contudo, no trabalho realizado na época, houve limitações funcionais devido à programação, que acarretou diretamente no seu funcionamento e apresentou falhas, não seguindo as rotinas que deveriam ser seguidas.

Portanto, o intuito é que, nesse momento, foram feitas as correções necessárias para que essa aplicação da robótica auxilie o deficiente visual.

1.2. OBJETIVOS

1.2.1. Objetivo geral

Solucionar problemas de funcionamento, buscando melhorias do robô autônomo e aplicar testes de operação para que possa auxiliar deficientes visuais em suas atividades cotidianas.

1.2.2. Objetivos específicos

- Solucionar problemas no código fonte das rotinas do robô;
- Trocar motores de vibrações por outros modelos;
- Avaliar a quantidade de sensores ultrassônicos para melhoria no desempenho;
- Projetar o robô em forma 3D no software *Tinkercard*;
- Executar a prototipagem do robô para realizar os ensaios.

1.3. JUSTIFICATIVA

A aplicação da robótica móvel ajuda a amenizar os problemas enfrentados pelos deficientes visuais para compreender suas principais limitações na locomoção, desde tarefas simples do dia a dia, como caminhar, atravessar ruas e desviar de obstáculos, como galhos de árvores e degraus, até irregularidades nas vias públicas.

Esta abordagem é inovadora para a comunidade científica, pois propõe a utilização da robótica para promover a acessibilidade dos deficientes visuais em suas atividades diárias.

Os deficientes visuais contam com diversos sistemas de locomoção, como bengalas, cães-guia e sistemas de reconhecimento de voz, Bersch (2008). No entanto, esses meios podem ser custosos e demandar muito tempo de treinamento, como no caso dos cães-guia. Além disso, alguns métodos não são tão eficazes e seu suporte é limitado, o que pode dificultar a mobilidade dessas pessoas, especialmente para aqueles com recursos financeiros limitados.

Portanto, este trabalho não só contribui com avanços acadêmicos e científicos por meio das pesquisas realizadas, mas também é de grande relevância prática ao

criar um dispositivo funcional e prático, com baixo custo e método de treinamento mais acessível, tornando a locomoção mais eficiente para os deficientes visuais.

Além disso, esse estudo é significativo para a sociedade, pois representa uma aplicação tecnológica na área da robótica, enriquecendo o currículo e proporcionando as pessoas a oportunidade de compartilhar suas descobertas com a comunidade universitária, mantendo todos conectados com os avanços atuais das engenharias.

2. DESENVOLVIMENTO

2.1. MATERIAL DIDÁTICO

O estudo em questão foi desenvolvido com base em uma pesquisa exploratória, que resultou no desenvolvimento de um protótipo. Esse processo incluiu a aplicação de uma das metodologias estudadas.

O objetivo foi obter um entendimento mais profundo das necessidades cotidianas dessas pessoas e, ao mesmo tempo, visualizar como um robô autônomo poderia ajudar a enfrentar essas dificuldades.

A finalidade do robô é o auxílio mais eficiente do que outros meios, como um cão-guia ou uma bengala. Para isso, o protótipo foi equipado com dois motores de corrente contínua individuais em cada roda e dois sensores ultrassônicos para detecção de obstáculos no lado esquerdo e no lado direito. Sendo que no trabalho anterior do autor Bitencourte (2018), foram utilizados 3 sensores ultrassônicos no modelo HC – SR04.

Houve essa redução de sensores para que a margem de erro de atuação diminua e seja mais fácil o reconhecimento dos obstáculos dessas duas direções e facilitar as manobras que o robô vai executar conforme a atuação dos sensores.

O motor vibracall que iria apresentar a sensibilidade de vibração ao usuário, houve a necessidade de troca. Anteriormente foram utilizados no trabalho do autor Bitencourte (2018) micro motores *vibracalls*, no qual na ponta do eixo tinha formato de meia lua, quando ele recebe o sinal para ser acionado este eixo gira causando um efeito de vibração. No protótipo atual utilizou - se o de formato de moeda ou motor sensor *vibracall* modelo 1027, a diferença um para o outro é que o eixo fica

encapsulado no componente, porém acontece o mesmo efeito de vibração. Essa escolha foi devido ao fácil acesso do componente.

O usuário é informado sobre os obstáculos detectados por meio de um sinal de vibração por motores *vibracalls*, junto com um sinal luminoso de um *led*, para sinalizar para pessoas ao seu redor. Esses motores vibram de acordo com as informações recebidas dos sensores ultrassônicos, permitindo ao usuário identificar a localização do obstáculo e agir em conformidade. Também, em paralelo, o usuário vai ter um sinal por voz, vai sinalizar a manobra que o robô irá fazer a esquerda ou direita.

Para a montagem do protótipo foram utilizados os itens abaixo da tabela 1.

Tabela 1: Componentes utilizados no protótipo.

Componentes	Unidades	Quantidades
Protoboard 1680 Furos Icel MSB-300	uni	1
Sensor Ultrassônico HC-SR04	uni	2
Resistor 1 kΩ	uni	6
Motor De Vibração Vibracall 1027	uni	2
Microcontrolador Arduino ATMEGA 2560	uni	1
Cabos Jumper's Macho-Fêmea	uni	30
Ponte H – CI L293D	uni	1
Motor DC – Fabr. Akiyama – (Cód. AK280/1.1PF5R193SC)	uni	2
Transistor BC 547	uni	2
Led's Alto Brilho	uni	4
Fonte 9V	uni	1
Módulo ISD 1820 com um alto falante de 5W.	uni	1
Cabo USB para comunicação	uni	1

Fonte: Autor (2024)

A linguagem utilizada foi C/C++ dentro do ambiente de programação da plataforma *IDE* Arduino para a compilação do programa.

A elaboração desse trabalho teve como base *datasheets*, manuais e literaturas aplicadas nesse seguimento. Antes de montar o protótipo foram realizados todos os testes no *Tinkercard*¹, contribuindo para que as simulações e resultados estivessem próximos da realidade prática do protótipo.

2.2. PROCEDIMENTO

Primeiramente foram identificadas as trajetórias necessárias quanto a movimentação do robô, para que fosse possível atender às necessidades dos deficientes visuais. Após essa análise, as rotinas levantadas foram aplicadas nas sintaxes da programação C/C++, tais rotinas são explicadas ao longo do trabalho.

O laço principal do microcontrolador, vai monitorar o comportamento dos sensores ultrassônicos e a oscilação do *Led* azul e branco, no qual não consta nenhuma detecção do sensor esquerda e direita. A explicação de cada laço, será apresentada em seguida com mais detalhes conforme a programação realizada em C / C++. A execução dos laços será conforme atuação dos sensores, porém cada sensor tem seu campo de atuação para fazer a leitura e comparação, conforme a sua resposta vai acarretar a um movimento do robô autônomo.

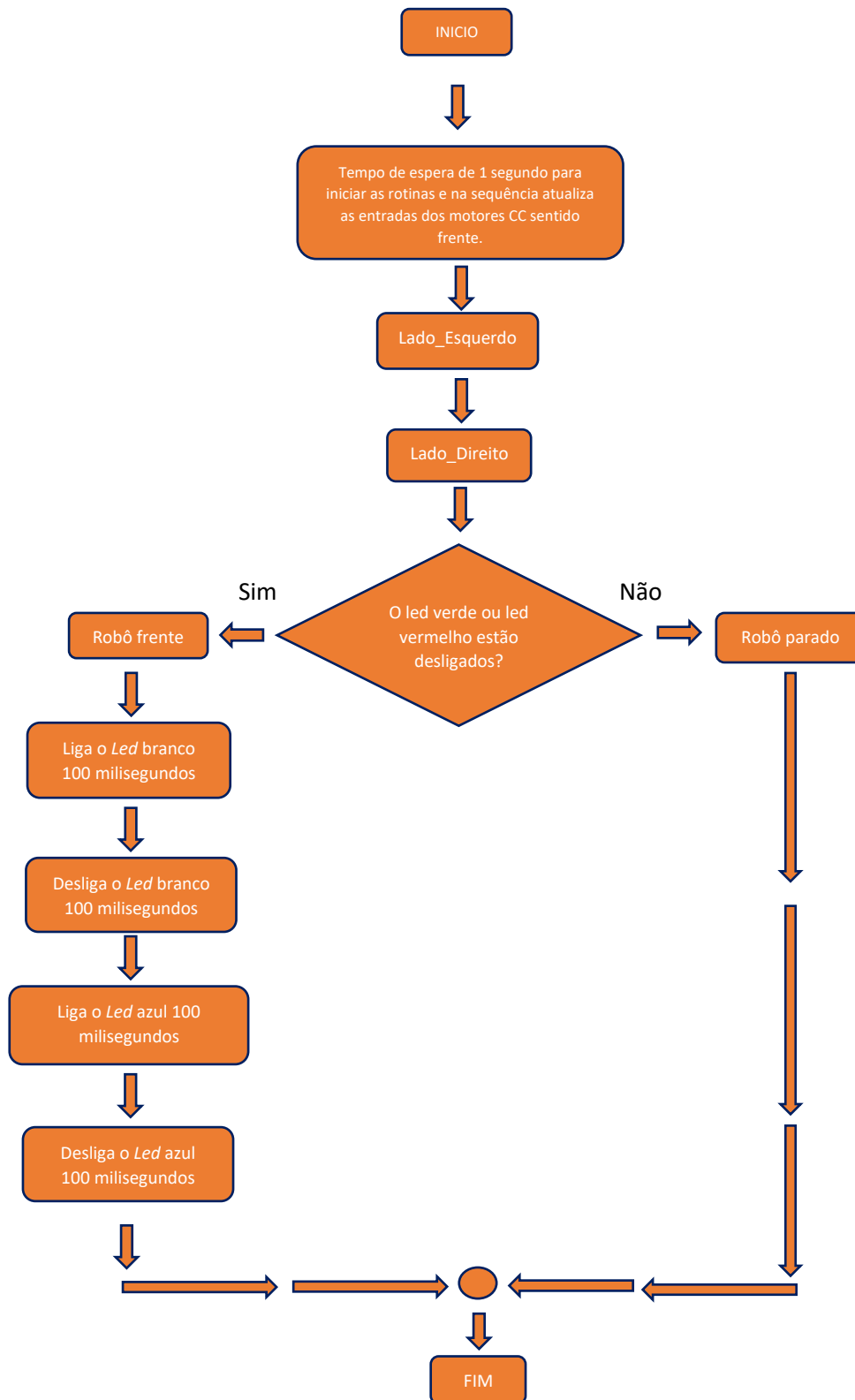
Toda programação foi realizada dentro do ambiente *Sketch*² *Arduino IDE*, tendo como base as sintaxes e as realizações dos testes do código. Então logo abaixo será explicado o laço principal do robô, mostra o passo a passo da rotina e de uma forma explicativa.

A Figura 1 representa o fluxograma do laço principal da programação.

¹ O Tinkercad é uma ferramenta online e gratuita, de criação simulação de arduino e também de design de modelos em 3D.

² Um projeto do Arduino é chamado de sketch, e consiste tipicamente em duas partes (rotinas): a rotina de setup, que inicializa o sketch, e a rotina de loop, que normalmente contém o código principal do programa (é como a função `main()` da linguagem C).

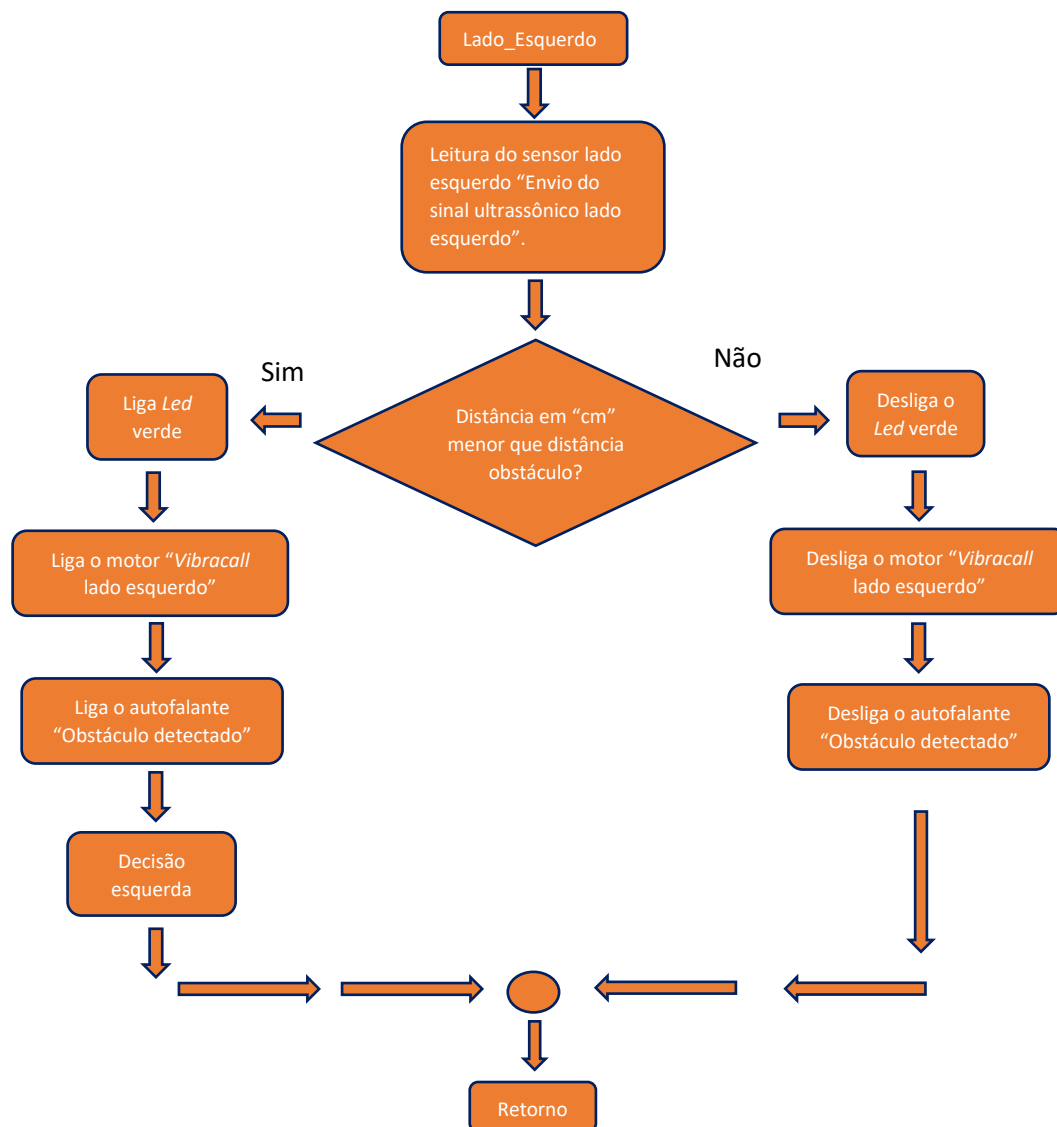
Figura 1: Fluxograma do laço principal da programação



Fonte: Autor (2024)

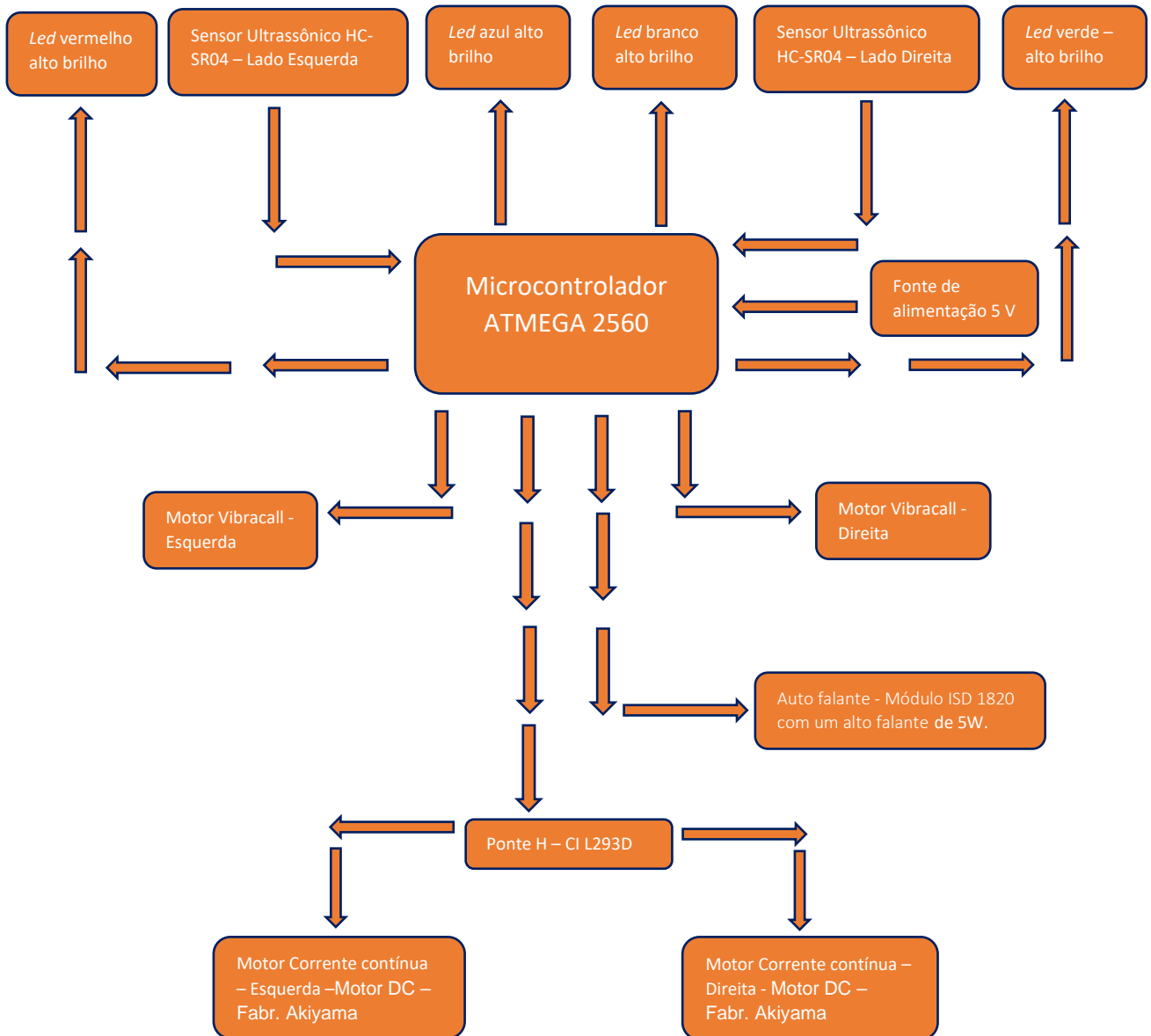
A Figura 2 é ilustrado o fluxograma da rotina para a movimentação do robô para o lado esquerdo. A tomada de decisão do robô após a detecção de um obstáculo na distância 10 cm. Esta distância pode ser alterada na programação, mas devido aos testes realizados é uma distância segura para identificar o risco ao obstáculo e tomar a decisão de realizar a manobra de desvio do mesmo. Na Figura 3 vai ilustrar um diagrama de blocos dos hardwares utilizados no protótipo.

Figura 2: Fluxograma da movimentação para o lado esquerdo



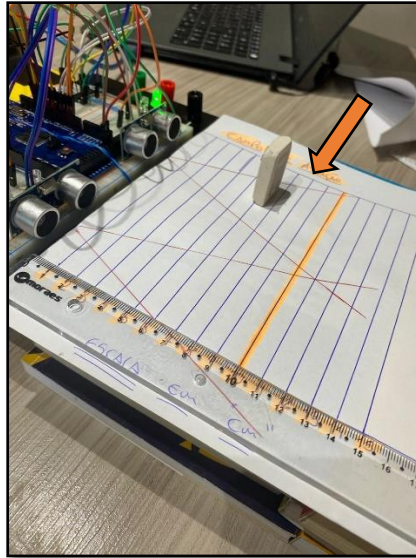
Fonte: Autor (2024)

Figura 3: Diagrama de bloco dos Hardware do protótipo



Fonte: Autor (2024)

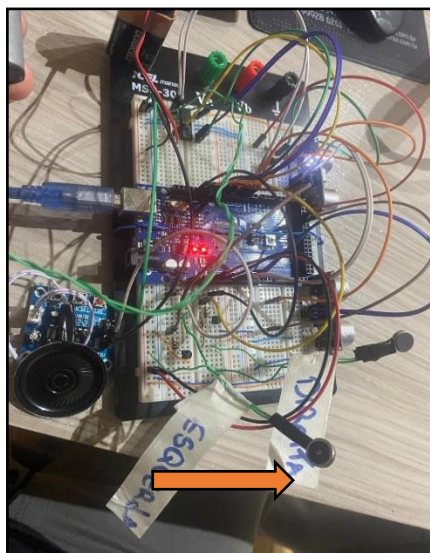
Figura 4: Detecção obstáculo lado esquerdo e sinalizando com *Led* verde ligado.



Fonte: Autor (2024)

Na Figura 4 para fim de ilustração é apresentado o protótipo instalado na *protoboard*. Conforme observado o obstáculo está sendo a borracha à uma distância menor que 10 cm, para realização de testes, conseqüentemente a sinalização luminosa verde é acionada. Em paralelo, o motor de vibração também é acionado, conforme indicado na Figura 5.

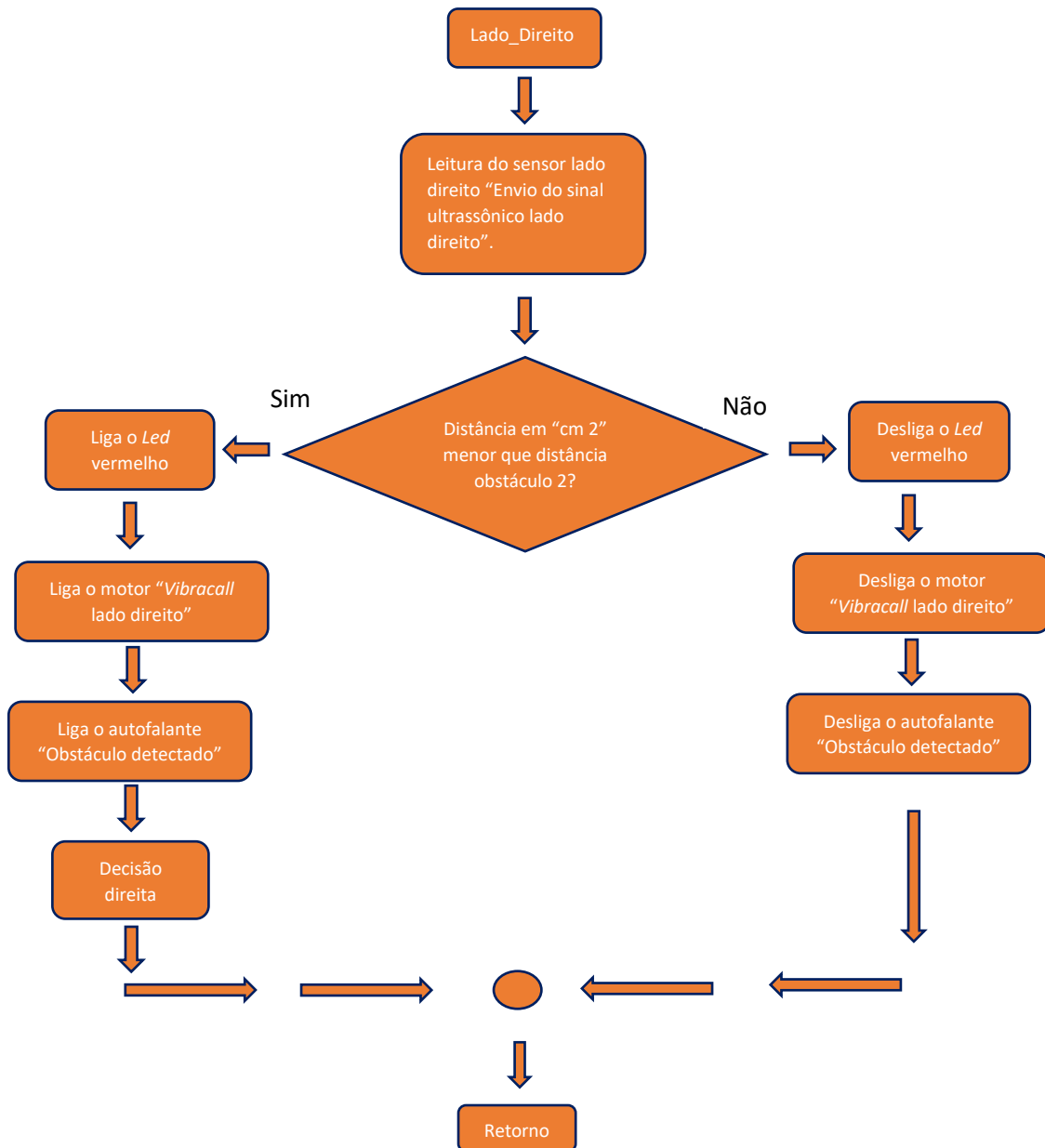
Figura 5: Após a detecção será recebido o sinal de vibração no motor esquerdo.



Fonte: Autor (2024)

Analogamente a Figura 6 apresenta a tomada de decisão para os obstáculos do lado direito.

Figura 6: Fluxograma da movimentação para o lado direito

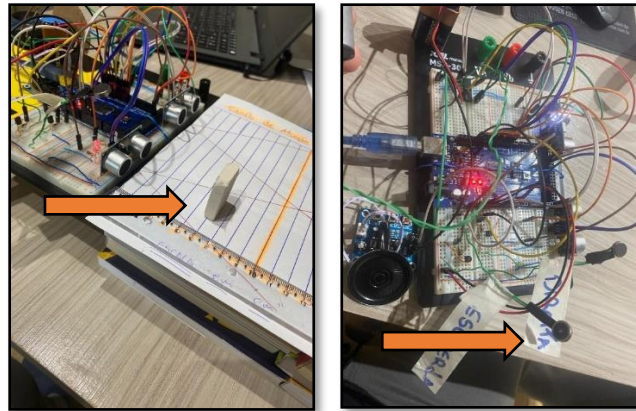


Fonte: Autor (2024)

Já na Figura 7 para fim de ilustração, o protótipo ilustra a atuação do sensor ultrassônico do lado direito do robô, nesse caso o obstáculo está sendo a borracha, pela atuação do sensor o obstáculo está menor que 10 cm de distância, após o reconhecimento pelo sensor é feita a sinalização luminosa e de vibração do motor de vibração do lado direito.

Figura 7: Detecção obstáculo lado direito e sinalizando com *Led* vermelho ligado e vibração do motor

vibracall

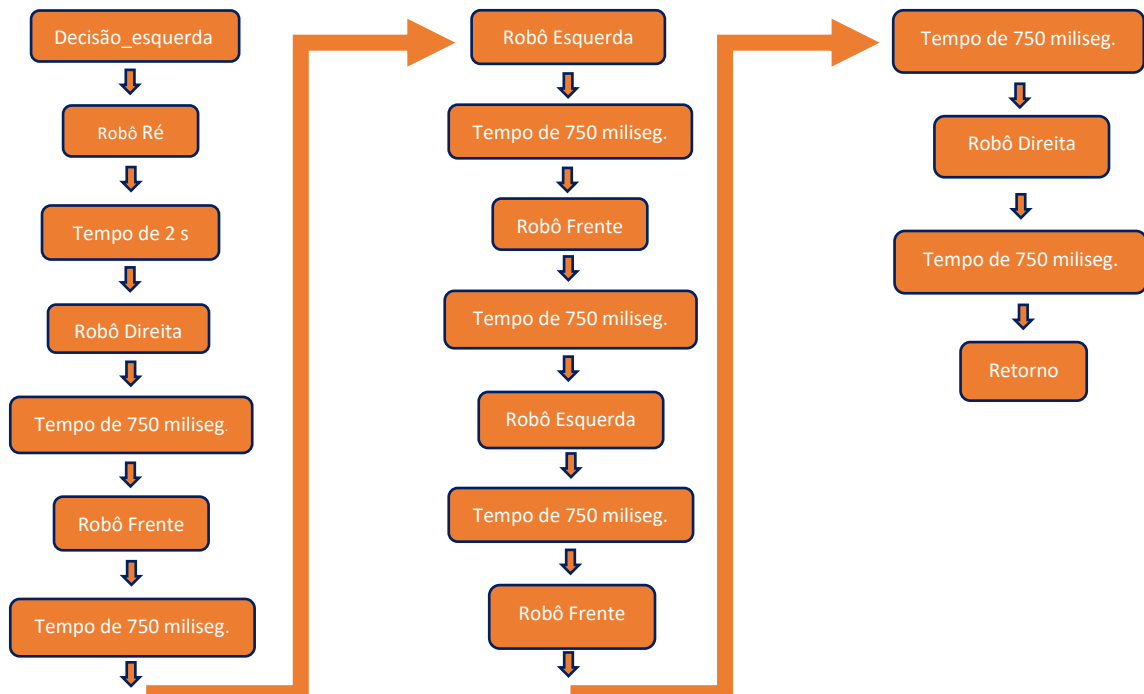


Fonte: Autor (2024)

A próxima rotina a ser explicada e detalhada é como o robô vai tomar as decisões de mudança de direção quando o obstáculo for reconhecido pelo respectivo sensor ultrassônico, conforme foi ilustrado na Figura 2.

A Figura 8 apresenta o fluxograma de decisão esquerda, possui a rotina que o robô vai fazer para desviar do obstáculo.

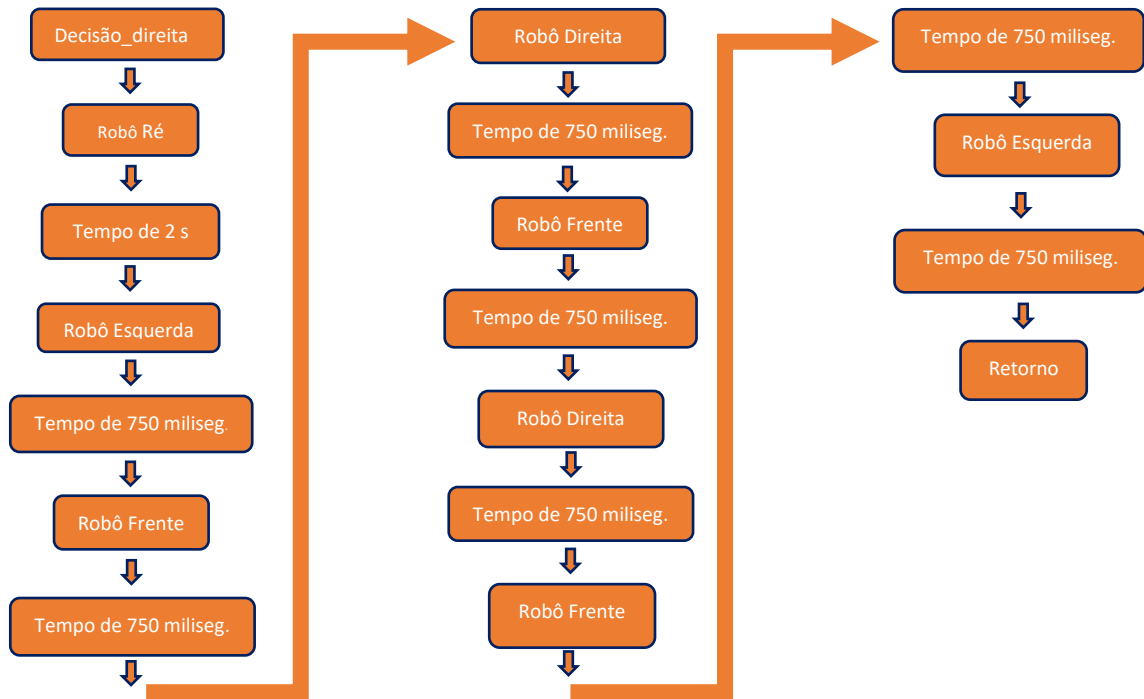
Figura 8: Fluxograma de decisão esquerda



Fonte: Autor (2024)

A Figura 9 apresenta o fluxograma de decisão direita, no qual possui a rotina que o robô vai fazer para desviar do obstáculo.

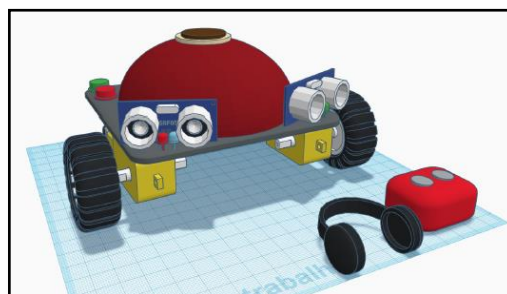
Figura 9: Fluxograma de decisão direita



Fonte: Autor (2024)

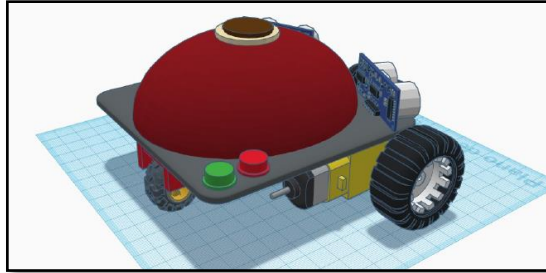
Como o artigo está apresentando um protótipo em uma *protoboard*, as Figuras 10,11 e 12 apresentam algumas imagens do robô elaborado no *Tinkercard*, tanto a parte eletrônica e como o modelo em 3D. O modelo proposto apresenta um designer compacto e interativo com o deficiente visual.

Figura 10: Projeto elaborado no *Tinkercard* em modelo 3D



Fonte: Autor (2024)

Figura 11: Robô projetado no *Tinkercard* vista lateral.



Fonte: Autor (2024)

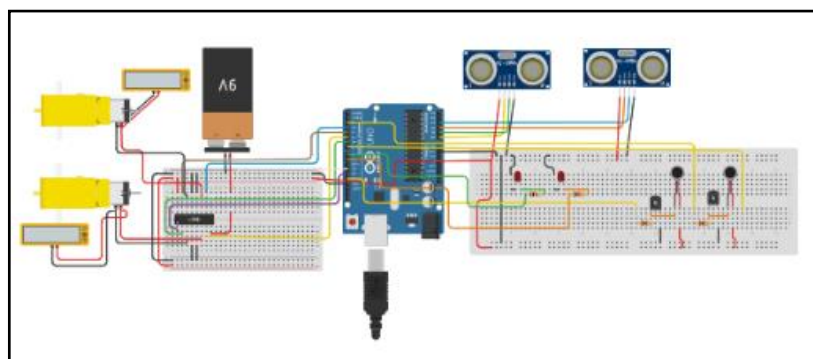
A elaboração desse projeto em 3D no simulador teve como intuito remoldar o estudo aplicado na graduação Bitencourte (2018), com as devidas correções realizadas.

Como adição, o robô apresenta agora um comando por voz e outro por um fone, também há um controle de mão no qual vai possuir os dois motores *vibracalls* indicando os obstáculos direita ou da esquerda.

Também foi projetado um botão de liga e desliga conforme está ilustrado na Figura 11. Ainda na parte superior é possível observar o autofalante externo que sinaliza “Obstáculo detectado”, conforme programado e pela atuação dos sensores ultrassônicos esquerda e direita. Na parte frontal existe os dois *Led’s* do robô em estado livre, ou seja sem detecção de obstáculos, e os outros dois *Led’s* na cor verde e vermelho que irão indicar o objeto no seu feixe de alcance.

A Figura 12 ilustra toda a parte eletrônica e simulado no *software Tinkercard*, com todos os componentes eletrônicos, sensores e medidores de tensão (voltímetro) para monitorar o comportamento dos motores de corrente contínua.

Figura 12: Projeto da parte eletrônica do simulador *Tinkercard*.



Fonte: Autor (2024)

Antes de realizar qualquer montagem e teste prático, foram realizados testes funcionais no simulador *Tinkercard*, quando o robô estava atuando conforme o proposto, foi realizado a montagem na *protoboard* para ilustração e testes do protótipo.

2.3. DISCUSSÃO E RESULTADOS

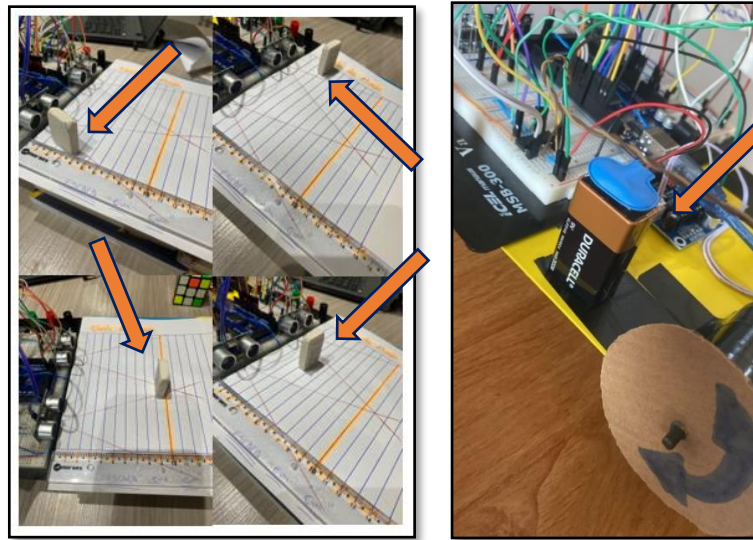
Neste artigo foi desenvolvido um protótipo que permite a interação homem/máquina, no qual o robô é autônomo, porém o usuário consegue identificar qual localização o obstáculo vai estar e assim desviá-lo.

Foi Projetado o robô em forma 3D no software *Tinkercard* para uma demonstração de como vai ficar em uma estrutura mais robusta para alvo de um protótipo futuro.

No decorrer do artigo foram ilustradas as rotinas e subrotinas do código do protótipo, ilustrando as movimentações que o robô vai executar, no apêndice A consta o código fonte em linguagem C/C++.

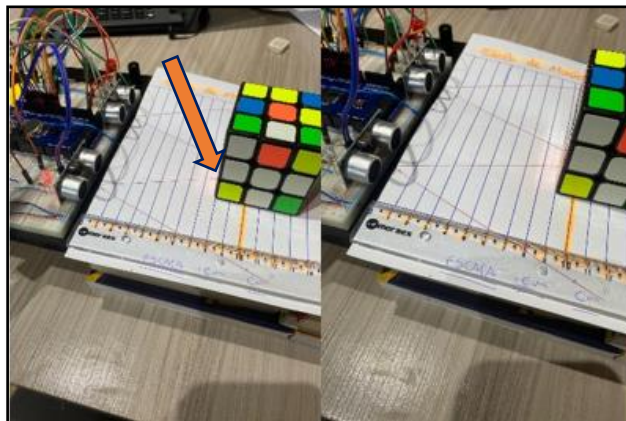
Durante a execução do projeto foi possível analisar algumas restrições. A primeira é sobre o alcance do sensor ultrassônico que possui um feixe de atuação de 15°, resultando em uns pontos de não detecção (pontos cegos). A segunda diz respeito à corrente/hora e tensão da bateria de 9V comercial que energiza os motores, para dar um maior desempenho para o robô, na Figura 13 ilustra o objeto exatamente nos pontos cegos fora da área de atuação dos sensores ultrassônicos.

Figura 13: Demonstrações dos pontos cegos dos sensores e bateria comercial de 9 Vcc



Fonte: Autor (2024)

Figura 14: Pontos cegos dentro campo de atuação do sensor



Fonte: Autor (2024)

A Figura 14 ilustra que o objeto está dentro do feixe conforme está a ilustração, na distância de 8 cm o sensor reconheceu o objeto já na distância de 9 cm ele não atuou, por isso a necessidade de aumento de mais um sensor para colocar na posição central. Ai será possível eliminar esse ponto de não visualização dentro do feixe de atuação do sensor.

Sendo assim, é necessário a ampliação de dois sensores ultrassônicos para três. Uma bateria cuja tensão e corrente seja mais adequada para aplicação.

As velocidades dos moto-redutores foram simuladas em bancada. O ideal para os ensaios do robô é colocar em uma condição real, onde os tempos de respostas dos sensores, algoritmo e dos motores serão melhores avaliados.

Nos testes funcionais em bancada foi desconsiderado o atrito com o solo, ou seja, é importante que faça os testes práticos com o atrito do solo. Nos ensaios práticos se percebeu que sejam feitos testes para que o robô consiga detectar os obstáculos, tomar uma ação de parada, e analisar a rotina de programação a ser executada.

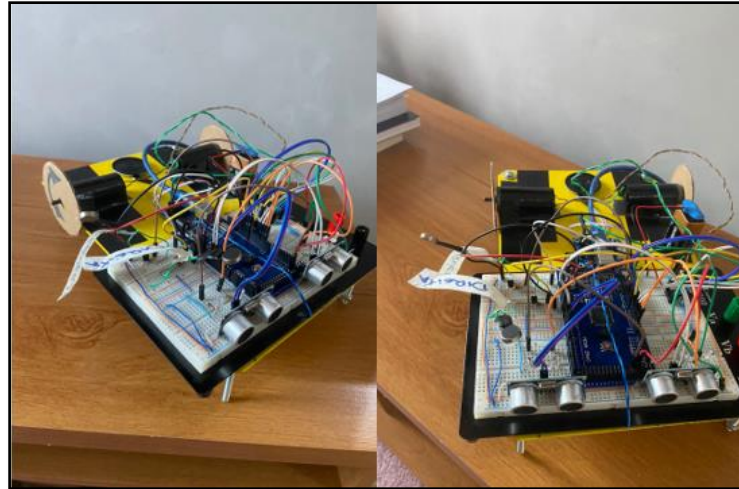
Conforme o protótipo projetado, consta um autofalante de 5W de potência. Esse dispositivo irá informar o usuário sobre a detecção de um obstáculo. Para mais eficiência na comunicação, recomenda-se a utilização de um fone de ouvido para que não ocorra nenhuma interferência de ruídos externos.

Quando o projeto estiver concluído, há necessidade de um treinamento, para instrução e manuseio do robô, que possui uma forma simplificada de utilização, tornando a locomoção do deficiente mais segura e eficaz.

O protótipo do robô foi montado na *protoboard* e os motores foram fixos em uma estrutura para apoiar os motores, porém foi realizado a elaboração do robô no *software Tinkercard*. No projeto atual, foram montados todos os circuitos eletrônicos detalhados com algumas melhorias que o robô anterior não tinha, uma delas e a principal apontada no objetivo do artigo foi a programação que está funcionando corretamente, outro item foram as na mudança das posições dos sensores colocando uma para direita e outro para esquerda.

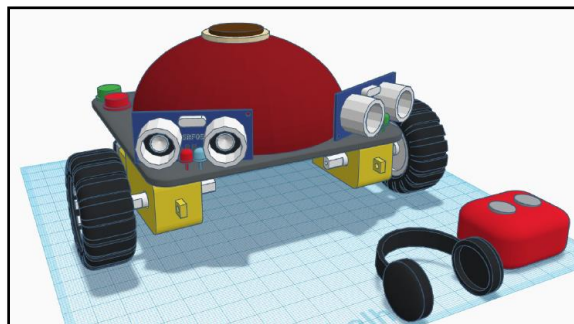
Também foi colocado um módulo de voz, no qual vai avisar: “Obstáculo detectado”. Na Figura 15 apresenta o protótipo em bancada e a Figura 16 ilustra o projeto no *Tinkercard*.

Figura 15: Protótipo desenvolvido para esse artigo



Fonte: Autor (2024)

Figura 16: Protótipo desenvolvido no *Tinkercard*



Fonte: Autor (2024)

Para concluir, foi possível deixar o robô funcionando corretamente e com funções diferentes e itens a mais em comparação com o primeiro robô executado. Portanto, após os testes funcionais com o protótipo, é ideal que seja fabricado um robô na versão projetada em 3D e colocado em operação, para avaliar a funcionalidade e realizar os ajustes necessários. Como outra melhoria, seria oportuno a implementação de um sistema de GPS para que o usuário defina seu destino e o robô guia até ele.

O sistema de rodas é importante realizar a mudança, pois é necessário instalar um tipo de roda que seja compatível para andar em vários locais. Para que não venha dar problemas no deslocamento do robô e acarretar um problema maior para o usuário.

3. CONSIDERAÇÕES FINAIS

O propósito da elaboração desse artigo foi resolver os problemas no código fonte das rotinas e subrotinas do robô anterior do trabalho Bitencourte (2018). Então foi possível resolver esses erros e trazer nesse artigo um robô funcional.

Também foi possível analisar que a redução de sensores ultrassônicos não foi ideal, pois acabou apresentando locais de risco, ou seja, não é possível reconhecer obstáculos devido a limitação do ângulo do seu feixe de alcance. Então, instalando mais sensores em pontos estratégicos acaba eliminando o problema, lembrando que há necessidade de realizar os testes funcionais com o robô na versão final.

Na realização de troca dos sensores de vibração teve a necessidade em que a superfície de vibração fosse lisa, o modelo de micro motor com o eixo aparente não seria o ideal, mas era possível aplica - ló na mesma situação, porém teria que elaborar uma superfície melhor para que o usuário pudesse sentir o sinal de vibração.

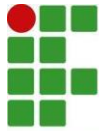
A elaboração da prototipagem foi necessária para realizar os testes funcionais e verificar a funcionalidade de cada componente nas suas atuações.

Para futuros projetos, é essencial desenvolver rodas adequadas para uso em diversos ambientes. Essas rodas devem ser resistentes, não corrosivas e impermeáveis. A comunicação entre o robô e o usuário deve ser remota, sem a necessidade de cabos, e o robô deve ter um botão de parada, caso o usuário tenha a necessidade para falar com alguma pessoa, ou travessia de rua com passagem de veículos. Em relação às baterias, o ideal é utilizar baterias com maior corrente e tensão.

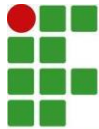
Além disso, para que o robô seja eficaz, é recomendado instalar um módulo GPS compatível com o microcontrolador Arduino.

4. REFERÊNCIAS

- 1- ALLDATASHEET. **Circuito Integrado L293D**. Disponível em: https://www.alldatasheet.com/view.jsp?Searchword=L293d%20datasheet&ga_d_source=1&gclid=Cj0KCQjw3tCyBhDBARIsAEY0XNIhnteNnXdLgabae2aRKBO0HpsKiWc-8BVosMkXnmcGOGtSC0p-DP0aAugrEALw_wcB. Acesso em: 27 maio 2024.



- 2- ALLDATASHEET. **Transistor BC547.** Disponível em:
https://www.alldatasheet.com/view.jsp?Searchword=Bc547%20datasheet&gad_source=1&gclid=Cj0KCQjw3tCyBhDBARIsAEY0XNm-nr6cAZRGTqKFXHJf4__Bd0W6aoETLMRjiSha6DnlBJ2UDQPSrVMaAo11EALw_wcB. Acesso em: 27 maio 2024.
- 3- AMIRANLIAN, M. L. T. M. (1997). **Compreendendo o cego: uma visão psicanalítica da cegueira por meio de desenhos-estórias.** São Paulo: Casa do Psicólogo.
- 4- BERSCH, Rita. **Introdução à tecnologia assistiva: CEDI centro especializado em desenvolvimento infantil.** Porto Alegre:, 2017. 20 p. Disponível em: <http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf>. Acesso em: 20 maio. 2024.
- 5- BITENCOURTE, Rudson João. **Aplicação da Robótica Auxiliando Deficientes Visuais.** 2018. 106 f. TCC (Graduação) - Curso de Engenharia Elétrica, Engenharias, Centro Universitário Facvest, Lages, 2018.
- 6- BURLAMAQUI, Aquiles Medeiros Filgueira et al. CardBot - Assistive Technology for Visually Impaired in Educational Robotics: Experiments and Results. **IEEE Latin America Transactions**, Rio Grande do Norte, v. 15, n. 3, p.517-527, 1mar.2017. Disponível em: <ieeexplore.ieee.org/document/7867603>. Acesso em: 15 abril. 2024.
- 7- CORE, Robo. **Sensor Ultrassônico HC-SR04.** Disponível em: https://www.robocore.net/tutoriais/primeiros-passos-com-sensor-ultrassonico?gad_source=1&gclid=Cj0KCQjw3tCyBhDBARIsAEY0XNm-aOOwopGJj64xhevAvVhQZ8f46hbSUz41jUnolZY4wHPYcLppWFj0aAsVwEALw_wcB. Acesso em: 27 maio 20.
- 8- EMBARCADOS, Sistemas. **Arduino MEGA.** Disponível em: <https://embarcados.com.br/arduino-mega-2560/>. Acesso em: 27 maio 2024.
- 9- MONTEIRO, André L.; SILVA, Eduardo M.; LOPES, Igor C. F. LOCALIZAÇÃO PARA DEFICIENTES VISUAIS UTILIZANDO UM GADGET PARA ORIENTAÇÃO. **Revista Innover**, Fortaleza- CE, v. 1, n. 4, p.93-105, 4 de dez 2014. Disponível



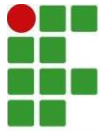
em:<www2.ifma.edu.br/revistainnover/Nova/index.php/inicio/article/download/55/35>. Acesso em: 21 abril. 2024.

10-NUNES, Sylvia; LOMÔNACO, José Fernando Bitencourt. O aluno cego: preconceitos e potencialidades. **Revista Semestral da Associação Brasileira de Psicologia Escolar e Educacional**, São Paulo, v. 14, n. 1, p.55-64, jan-jun 2010. Disponível em: <<http://www.scielo.br/pdf/pee/v14n1/v14n1a06>>. Acesso em: 21 abril. 2024

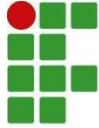
11-SOUZA, Fábio. **Arduino Mega 2560**. Disponível em: <<https://www.embarcados.com.br/arduino-mega-2560/>>. Acesso em: 14 de maio. de 2024.

Apêndice A - Código fonte do protótipo executado.

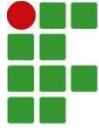
```
// inclusão de bibliotecas.
#include <Ultrasonic.h> // Biblioteca para os Sensores Ultrassônicos
// Variáveis do Sensor Ultrassônico Esquerda.
#define trigPin A0 //Pino TRIG do sensor no pino analógico A0.
#define echoPin A1 //Pino ECHO do sensor no pino analógico A1.
// Variáveis do Sensor Ultrassônico Direita.
#define trigPin2 A2 //Pino TRIG do sensor no pino analógico A2.
#define echoPin2 A3 //Pino ECHO do sensor no pino analógico A3.
//Variáveis LED's esquerdo
#define led1 2 // Led indicando obstáculo Lado Esquerdo cor branco.
#define led2 28 // Led indicando obstáculo Lado Esquedo cor verde.
//Variáveis LED's direita
#define led3 3 // Led indicando obstáculo Lado Direito cor azul.
#define led4 30 // Led indicando obstáculo Lado Direito cor vermelha.
// Motor Direita
#define velocidade1 4 // Pino que está ligado o eneble.
#define m1AntiHorario 5 // Sentido para trás.
#define m1Horario 6 // Sentido frente.
// Motor Esquerda
#define velocidade2 7 // Pino que está ligado o eneble.
#define m2AntiHorario 8 // Sentido para trás.
#define m2Horario 9 // Sentido frente.
// Variáveis Motores Vibracais.
#define motor1_sensor_esquerda 10 // Pino Digital Que Está Ligado Motor Vibracall Lado Esquerdo.
#define motor2_sensor_direita 11 // Pino Digital Que Está Ligado Motor Vibracall Lado Direita.
// Variáveis do Sensor Ultrassônico Esquerda.
long duracao;
long distancia_cm = 0;
int minimumRange = 5; //tempo de resposta do sensor.
int maximumRange = 200;
int distancia_obstaculo = 10; // Distância que o sensor vai detectar obstáculo lado esquerdo em cm.
// Variáveis do Sensor Ultrassônico Direira.
long duracao2;
long distancia_cm2 = 0;
int minimumRange2 = 5; //tempo de resposta do sensor.
int maximumRange2 = 200;
int distancia_obstaculo2 = 10; // Distância que o sensor vai detectar obstáculo lado esquerdo em cm.
void setup()
{
  Serial.begin(115200); //Inicia a comunicação serial com velocidade de 115200 bits por segundo.
  //Saída do alto falante.
  pinMode(12,OUTPUT);
  // Saídas Motores Vibracais dos Dois Lados.
  pinMode(motor1_sensor_esquerda, OUTPUT);
  pinMode(motor2_sensor_direita, OUTPUT);
  // Entrada e Saídas dos Motores DC's.
  pinMode(velocidade1, OUTPUT);
  pinMode(m1AntiHorario, OUTPUT);
  pinMode(m1Horario, OUTPUT);
}
```



```
pinMode(velocidade2, OUTPUT);
pinMode(m2AntiHorario, OUTPUT);
pinMode(m2Horario, OUTPUT);
// Entradas e Saídas dos Sensores Ultrassônicos.
pinMode(trigPin, OUTPUT); //Define o pino TRIG como saída (Sensor 1).
pinMode(echoPin, INPUT); //Define o pino ECHO como entrada (Sensor 1).
pinMode(trigPin2, OUTPUT); //Define o pino TRIG como saída (Sensor 2).
pinMode(echoPin2, INPUT); //Define o pino ECHO como entrada (Sensor 2).
// Saídas dos LED's.
pinMode(led1, OUTPUT); // Saída do Led indicando obstáculo Lado Esquerdo.
pinMode(led2, OUTPUT); // Saída do Led indicando obstáculo Lado Esquerdo
pinMode(led3, OUTPUT); // Saída do Led indicando obstáculo Lado Direita.
pinMode(led4, OUTPUT); // Saída do Led indicando obstáculo Lado Direita.
delay (1000); // Tempo de espera para iniciar a operação do robô.
}
// Laço principal da programação de rotina do robô a ser executada.
void loop()
{
    delay (1000); // Tempo de espera para iniciar a operação do robô.
    digitalWrite(velocidade1, m1Horario); //Atualizando as velocidades dos motores.
    digitalWrite(velocidade2, m2Horario); //Atualizando as velocidades dos motores.
    //Primeira comparação a ser feita pelo robô.
    if (digitalRead(led2) == HIGH || digitalRead(led4) == HIGH) // Se um dos dois led's estiver em sinal alto vai executar a rotina.
    {
        robo_parado();
    }
    else
    {
        robo_frente();
        digitalWrite(led1, HIGH); // Led 1 Ligado cor branca.
        delay (100);
        digitalWrite(led1, LOW); // Led 1 Ligado cor branca.
        delay (100);
        digitalWrite(led3, HIGH); // Led 3 Ligado cor azul.
        delay (100);
        digitalWrite(led3, LOW); // Led 3 Ligado cor azul.
        delay (100);
    }
    lado_esquerdo(); //Chama subrotina.
    lado_direito(); //Chama subrotina.
}
//Subrotina de comparação para detecção do lado esquerdo.
void lado_esquerdo()
{
    leitura_esquerda();
    if (distancia_cm < distancia_obstaculo) // Se a distância for menor que 10 cm.
    {
        digitalWrite(led2, HIGH); // Led 1 Ligado cor verde.
        digitalWrite(motor1_sensor_esquerda, HIGH); // Motor Vibracall esquerdo ligado.
        digitalWrite(12, HIGH); // Sinal de voz no pino L que está ligado no pino 12.
        decisao_esquerda();
    }
    else
    {
        digitalWrite(motor1_sensor_esquerda, LOW); // Motor Vibracall esquerdo desligado.
        digitalWrite(12, LOW);
        digitalWrite(led2, LOW); // Led 2 Desligado.
    }
}
//Subrotina de comparação para detecção do lado direito.
void lado_direito()
{
    leitura_direita();
    if(distancia_cm2 < distancia_obstaculo2) // Se a distância for menor que 10cm.
    {
        digitalWrite(led4, HIGH); // Led 1 Ligado cor vermelho
        digitalWrite(motor2_sensor_direita, HIGH); // Motor Vibracall direita ligado.
        digitalWrite(12, HIGH); // Sinal de voz no pino L que está ligado no pino 12.
        decisao_direita();
    }
    else
    {
        digitalWrite(12, LOW);
        digitalWrite(motor2_sensor_direita, LOW); // Motor Vibracall direita desligado.
        digitalWrite(led4, LOW); // Led 4 Desligado.
    }
}
// Função para ler e calcular a distância do sensor ultrassônico do lado esquerdo.
void leitura_esquerda()
{
    digitalWrite(trigPin, LOW); //não envia som.
```



```
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); //envia som.
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW); //não envia o som e espera o retorno do som enviado.
    duracao = pulseIn(echoPin, HIGH); //Captura a duração em tempo do retorno do som.
    distancia_cm = duracao /58; //Calcula a distância.
    return;
}
// Função para ler e calcular a distância do sensor ultrassônico do lado direito.
void leitura_direita()
{
    digitalWrite(trigPin2, LOW); //não envia som
    delayMicroseconds(2);
    digitalWrite(trigPin2, HIGH); //envia som
    delayMicroseconds(10);
    digitalWrite(trigPin2, LOW); //não envia o som e espera o retorno do som enviado
    duracao2 = pulseIn(echoPin2, HIGH); //Captura a duração em tempo do retorno do som.
    distancia_cm2 = duracao2 /58; //Calcula a distância
    return;
}
// Subrotina de decisão esquerda
void decisao_esquerda()
{
    if (digitalRead(led2) == HIGH) // Se o led estiver em sinal alto vai executar a rotina.
    {
        robo_da_re();
        delay(2000);
        robo_direita();
        delay(750);
        robo_frente();
        delay(750);
        robo_esquerda();
        delay(750);
        robo_frente();
        delay(750);
        robo_esquerda();
        delay(750);
        robo_frente();
        delay(750);
        robo_direita();
        delay(750);
    }
}
// Subrotina de decisão direita
void decisao_direita()
{
    if (digitalRead(led4) == HIGH) // Se o led estiver em sinal alto vai executar a rotina.
    {
        robo_da_re();
        delay(2000);
        robo_esquerda();
        delay(750);
        robo_frente();
        delay(750);
        robo_direita();
        delay(750);
        robo_frente();
        delay(750);
        robo_direita();
        delay(750);
        robo_frente();
        delay(750);
        robo_esquerda();
        delay(750);
    }
}
// Laço que o robo anda pra frente.
void robo_frente()
{
    analogWrite(velocidade1, 255); //Motor lado direito.
    digitalWrite(m1Horario, HIGH); // Motor lado direito sentido frente.
    analogWrite(velocidade2, 255); //Motor lado esquerdo
    digitalWrite(m2Horario, HIGH); // Motor lado esquerdo sentido frente.
}
// Laço que o robo fica parado.
void robo_parado()
{
    analogWrite(velocidade1, 0); //Motor lado direito pulso PWM em 0.
    digitalWrite(m1Horario, LOW); //Sentido lado direito frente desligado.
    analogWrite(velocidade2, 0); //Motor lado esquerdo pulso PWM em 0.
    digitalWrite(m2Horario, LOW); //Sentido lado esquerdo frente desligado.
    analogWrite(velocidade1, 0); //Motor lado direito pulso PWM em 0.
```



```
digitalWrite(m1AntiHorario, LOW); //Sentido lado direito atrás desligado.
analogWrite(velocidade2, 0); //Motor lado esquerdo pulso PWM em 0.
digitalWrite(m2AntiHorario, LOW); //Sentido lado esquerdo atrás desligado.
}
// Laço que o robo anda para atrás.
void robo_da_re()
{
digitalWrite(m1Horario, LOW); // Sentido lado direito frente desligado.
analogWrite(velocidade1, 255); // Sentido lado direito atrás pulso PWM 255.
digitalWrite(m1AntiHorario, HIGH); // Sentido lado direito atrás ligado.
analogWrite(velocidade2, 255); // Sentido lado esquerdo atrás pulso PWM 255.
digitalWrite(m2AntiHorario, HIGH); // Sentido lado esquerdo atrás ligado.
digitalWrite(m2Horario, LOW); // Sentido lado esquerdo frente desligado.
}
// Laço que o robo vira a direita.
void robo_direita()
{
analogWrite(velocidade1,0);
digitalWrite(m1Horario, LOW);
digitalWrite(m1AntiHorario, LOW);
analogWrite(velocidade2, 255);
digitalWrite(m2Horario, HIGH);
digitalWrite(m2AntiHorario, LOW);
}
// Laço que o robo vira a esquerda.
void robo_esquerda()
{
analogWrite(velocidade1,255);
digitalWrite(m1Horario, HIGH);
digitalWrite(m1AntiHorario, LOW);
analogWrite(velocidade2, 0);
digitalWrite(m2Horario, LOW);
digitalWrite(m2AntiHorario, LOW);
}
```