

**CENTRO PAULA SOUZA
FATEC DE SANTO ANDRÉ
Curso de Tecnologia em Mecatrônica Industrial**

Matheus dos Santos Ezidro

ASSISTENTE DE ESTACIONAMENTO AUTOMÁTICO

**Santo André
2022**

Matheus dos Santos Ezidro

ASSISTENTE DE ESTACIONAMENTO AUTOMÁTICO

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial da Fatec de Santo André, orientado pelo prof. Dr. Edson C. Kitani e co-orientado pelo prof. Msc Wesley Torres para obtenção do título de Tecnólogo em Mecatrônica Industrial

Santo André

2022

E99a

Ezidro, Matheus dos Santos

Assistente de estacionamento automático / Matheus dos Santos Ezidro. - Santo André, 2022. – 52f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Mecatrônica Industrial, 2022.

Orientador: Prof. Dr. Edson Caoru Kitani

1. Mecatrônica. 2. Automação. 3. Projeto. 4. Tecnologia. 5. Park Assist. 6. Sensor ultrassônico. 7. Servo motor. 8. Programação. 9. Estacionamento automático. 10. Arduino UNO. I. Assistente de estacionamento automático.

629.89

LISTA DE PRESENÇA

Santo André, 07 DE JULHO DE 2022.

**LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO
TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA:
“ASSISTENTE DE ESTACIONAMENTO AUTOMÁTICO ” DOS
ALUNOS DO 6º SEMESTRE DESTA U.E.**

BANCAPRESIDENTE:
PROF. EDSON CAORU KITANI _____MEMBROS:
PROF. FERNANDO GARUP DALBO _____

PROF. WESLEY MEDEIROS TORRES _____

ALUNO:

MATHEUS DOS SANTOS EZIDRO _____

Dedico esse trabalho a todos estudantes que chegam a etapa final de um curso superior, passando por todas dificuldades externas aos estudos e que mesmo assim não desistiram.

AGRADECIMENTO

A todos os professores por compartilhar o conhecimento necessário com destreza e paciência para com os alunos, organizando seu tempo para fazer com que o máximo de alunos possíveis alcançassem sucesso sem interferir no aprendizado, apenas auxiliando a caminhada nessa jornada de ensino superior.

“O sucesso é a soma de pequenos esforços repetidos dia após dia.”

ROBERT COLLIER

RESUMO

O projeto Assistente de Estacionamento Automático se inspirou na tecnologia *Park Assist*. Essa tecnologia é utilizada para auxiliar o motorista a realizar uma manobra de estacionamento paralelo, utilizando sensores ultrassônicos ao redor do veículo para identificar a distância do veículo com objetos. A partir desses sensores, calcula quais movimentos necessários para realizar o movimento de estacionamento entre dois objetos. Este projeto tem o objetivo de utilizar componentes eletrônicos de baixo custo para simular em escala um estacionamento paralelo. Iniciou-se então com a organização de todos os componentes, utilizando um Arduino UNO como processador, um servo motor para direção das rodas e sensores ultrassônicos HC-SR04 para determinar a distância do veículo. Para posicionar os componentes, foi utilizado um chassi de um carrinho de controle remoto, com suportes impressos em 3D para sustentação dos sensores. Após montar o protótipo, uma pista de testes foi montada com papelão. Com o protótipo montado, todos os componentes testados, iniciou a programação para realizar os testes práticos. Para atingir o objetivo foram realizados dois testes, o primeiro teste com um espaço disponível para realizar o movimento de estacionamento, e o segundo teste realizado com um espaço que não comporte o protótipo, para testar se a programação conseguiu verificar o tamanho da vaga. Ambos os testes foram repetidos 10 vezes para verificar a precisão do projeto, atingindo uma precisão total de 80% nos testes. Com toda montagem do protótipo, testes nas pistas e a verificação da precisão do projeto, pudemos verificar que apesar dos componentes de baixo custo foi possível atingir uma boa precisão do objetivo estabelecido.

Palavras-chave: Estacionar. Assistente. Arduino. Sensores.

ABSTRACT

The Automatic Parking Assistant project was inspired by Park Assist technology. This technology is used to assist the driver in performing a parallel parking manoeuvre, using ultrasonic sensors around the vehicle to identify the distance from the vehicle to objects. From these sensors, it calculates which movements are necessary to carry out the parking movement between two objects. This project aims to use low-cost electronic components to simulate a parallel parking lot in scale. Then it started with the organization of all components, using an Arduino UNO as a processor, a servo motor for steering the wheels and ultrasonic sensors HC-SR04 to determine the distance of the vehicle. To position the components, a chassis of a remote control cart was used, with 3D-printed supports to support the sensors. After assembling the prototype, a test track was assembled with cardboard. With the prototype assembled, all components tested, the program began to carry out the practical tests. To achieve the objective, two tests were carried out, the first test with an available space to carry out the parking movement, and the second test carried out with a space that does not support the prototype, to test if the programming was able to verify the size of the parking space. Both tests were repeated 10 times to verify the accuracy of the design, reaching a total accuracy of 80% in the tests. With all the assembly of the prototype, tests in the tracks and the verification of the precision of the project, we were able to verify that despite the low cost components it was possible to reach a good precision of the established objective.

Keywords: Parking. Assistant. Arduino. Sensors.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Exame do DETRAN | 12 |
| Figura 2 - Park Assist..... | 13 |
| Figura 3 - Park Assist BMW Serie 7 | 15 |
| Figura 4 - Fluxograma projeto paralelo..... | 17 |
| Figura 5 - Raios internos e externos de uma baliza | 17 |
| Figura 6 - Teste de sensores pista 1 | 18 |
| Figura 7 - Teste de sensores pista 2 | 18 |
| Figura 8 - Chassi versão 1 com componentes | 20 |
| Figura 9 - Chassi de MDF | 21 |
| Figura 10 - Chassi de PLA | 21 |
| Figura 11 – Chassi versão 3 | 22 |
| Figura 12 - Arduino UNO | 23 |
| Figura 13 - Sensor ultrassônico HC-SR04 | 23 |
| Figura 14 - Funcionamento do sensor..... | 24 |
| Figura 15 - Gráfico do sinal ultrassônico do HC-SR04 | 25 |
| Figura 16 - Chassi da Ferrari F40 RF..... | 25 |
| Figura 17 - Conceito de ponte H | 26 |
| Figura 18 - Driver L293D..... | 27 |
| Figura 19 - Servo motor MG996R | 27 |
| Figura 20 - Conversor LM2596 | 28 |
| Figura 21 - Esquema de ligação do conversor | 28 |
| Figura 22 - Fluxograma de operações | 29 |
| Figura 23 - Esquema digital das ligações..... | 30 |
| Figura 24 - Layout de componentes..... | 30 |

| | |
|--|----|
| Figura 25 - Primeiro protótipo montado..... | 31 |
| Figura 26 - Segundo protótipo montado..... | 31 |
| Figura 27 - Adaptação na barra de ligação | 32 |
| Figura 28 - Representação do chassi..... | 33 |
| Figura 29 - Novo layout dos componentes | 33 |
| Figura 30 - Suporte provisório do servo motor | 34 |
| Figura 31 - Protótipo final montado | 34 |
| Figura 32 - Dimensões da balliza..... | 35 |
| Figura 33 - Papelão para pista de testes..... | 35 |
| Figura 34 – Estruturas para pista de testes..... | 36 |
| Figura 35 - Programação do motor e direção | 37 |
| Figura 36 - Primeiro teste de vaga na dimensão correta | 38 |
| Figura 37 - Vaga de dimensão incorreta | 38 |
| Figura 38 - Alimentação direta na tomada..... | 39 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Comparação de projetos | 19 |
| Tabela 2 - Resultado do teste 1 | 40 |
| Tabela 3 - Resultado do teste 2 | 40 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|--|
| CONTRAN | Conselho Nacional de Trânsito |
| DC | <i>Direct Current</i> |
| KHZ | kiloHertz |
| LED | <i>Light Emission Diode</i> |
| LI-PO | Polímero de íons de Lítio |
| MDF | <i>Medium Density Fiberboard</i> |
| MM | Milímetros |
| NASA | <i>National Aeronautics and Space Administration</i> |
| PLA | <i>PoliLatic Acid</i> |
| PWM | <i>Pulse Width Modulation</i> |
| RF | Rádio Frequência |

SUMÁRIO

| | | |
|------------|---|-----------|
| 1 | INTRODUÇÃO | 12 |
| 1.1 | Inspiração do projeto | 12 |
| 1.2 | Objetivo | 13 |
| 1.3 | Metas | 13 |
| 2 | REVISÃO BIBLIOGRÁFICA E REFERENCIAL TEÓRICO | 15 |
| 2.1 | Sistema de estacionamento automático | 15 |
| 2.2 | Projetos universitários sobre <i>Park Assist</i> | 16 |
| 3 | DESENVOLVIMENTO DO PROJETO | 20 |
| 3.1 | Montagem do chassi versão 1 | 20 |
| 3.2 | Montagem do chassi versão 2 | 21 |
| 3.3 | Montagem do chassi versão 3 | 22 |
| 3.4 | Componentes e suas funções | 22 |
| 3.4.1 | Arduino | 23 |
| 3.4.2 | Sensor HC-SR04 | 23 |
| 3.4.4 | Driver para ponte H | 26 |
| 3.4.5 | Servo motor | 27 |
| 3.4.6 | Módulo conversor de tensão DC/DC | 28 |
| 3.5 | Fluxograma básico | 29 |
| 3.6 | Montagem física | 30 |
| 4 | TESTES E RESULTADOS | 37 |
| 4.1 | Teste dos motores com programação | 37 |
| 4.2 | Teste de vaga com dimensão correta | 37 |
| 4.3 | Teste de vaga com dimensão incorreta | 38 |
| 4.4 | Resultados do testes | 39 |
| 5 | CONCLUSÃO | 42 |
| 6 | PROPOSTAS FUTURAS | 43 |
| | REFERÊNCIAS | 44 |
| | APÊNDICE A – Programação do Arduíno | 47 |

1 INTRODUÇÃO

Neste capítulo será apresentado o início do projeto, mostrando quais foram as motivações para iniciar a pesquisa, o objetivo a ser atingido e por fim quais metas desejáveis são estabelecidas após o cumprimento do objetivo.

1.1 Inspiração do projeto

No Brasil, já existe 1 automóvel para cada quase 5 habitantes, o que mostra o aumento de veículos nas metrópoles (REIS, 2014), o que trouxe dificuldades adicionais para o trânsito, como lentidão e o tópico principal deste projeto que é a dificuldade de encontrar uma vaga para estacionar. Para a maximização da utilização do espaço disponível, os empresários projetam os estacionamentos criando diversas vagas com restrições de acesso e tamanho. Além dos problemas causados pela quantidade de veículos, também deve ser considerada a habilidade de realizar a manobra de estacionamento paralelo. O exame de certificação do Departamento Estadual de Trânsito (DETRAN) prevê até 5 minutos para conclusão da manobra, conforme a Figura 1 (DETRAN, 2022). Com esses pontos observados, fica evidente que um auxiliar de estacionamento é muito útil, pois além de ajudar quem tem dificuldade na direção ou pouca noção espacial, ele consegue poupar tempo de estacionamento, levando em consideração que para estacionar com o assistente ligado necessitamos de 2 minutos nas vagas mais restritas. O sistema utiliza alertas visuais e sonoros sobre a proximidade de veículos ou objetos.

Figura 1 - Exame do DETRAN



Fonte: (PORTAL AUTO SHOPPING, 2018)

A motivação para esta pesquisa nasceu da percepção de que processos, exercícios ou funções podem ser otimizadas através da utilização de tecnologia, como é o caso das melhorias contínuas nas fábricas com a Indústria 4.0. Após introduzir a motivação, o projeto conta com um objetivo a ser alcançado.

1.2 Objetivo

Os primeiros sistemas de estacionamento de veículos por sensores eram simples e assim como esses primeiros sistemas, este projeto tem o objetivo de desenvolver um sistema *Park Assist* ilustrado na Figura 2 em um veículo protótipo em escala. Os componentes deste projeto são os mesmos de um veículo real, porém de fácil acesso e configuração. Na montagem física serão utilizados sensores ultrassônicos para indicar a posição do veículo e um motor DC (*Direct Current*) para movimentação das rodas. A programação do *software* será feita por meio de um Arduino UNO.

Figura 2 - *Park Assist*



Fonte: (TOUME, 2017)

1.3 Metas

Uma vez implementado o sistema *Park Assist* no veículo protótipo, serão realizados testes para determinar a eficiência do projeto. O objetivo do projeto é realizar um estacionamento automaticamente uma única vez. Porém, o ideal é realizar a tarefa

proposta quantas vezes for necessária. Outro ponto é a quantidade de balizas que conseguiremos realizar sem sofrer alterações no trajeto e o tamanho mínimo de vaga para que ocorra o estacionamento de forma correta.

Esses pontos observados se encaixam no objetivo principal, porque apesar de o dispositivo realizar um movimento de estacionamento paralelo, para uma melhoria futura é necessário ter conhecimento das limitações existentes no projeto. E nos testes práticos poderemos obter os resultados desejados, ou se for o caso, procurar pontos de melhorias.

2 REVISÃO BIBLIOGRÁFICA E REFERENCIAL TEÓRICO

Neste capítulo serão apresentados as bases teóricas e os principais trabalhos já realizados sobre sistemas de estacionamento automático. O capítulo fornecerá informações importantes para entender como os sistemas comerciais funcionam, e também, apresentar alguns trabalhos acadêmicos semelhantes ao deste projeto.

2.1 Sistema de estacionamento automático

Os primeiros desenvolvimentos de um assistente automático se iniciaram no ano 2000 com a Volkswagen, mas somente em 2006 que foi implementado na minivan Touareg. Esse primeiro sistema era mais simples do que os atuais, precisando de 1,40 m de espaço a mais do que o tamanho do veículo para realizar o movimento de estacionamento. A segunda geração do *Park Assist* da Volkswagen veio em 2010, com 12 sensores ultrassônicos, melhorando a acurácia do estacionamento, e requerendo 90 cm de espaço a mais na vaga. Já em 2015 o Passat veio com o *Park Assist 3.0*, acrescentando a função de estacionamento em vagas perpendiculares e diminuindo o espaço necessário para 80 cm. (REDAÇÃO BLOGAUTO, 2017).

Atualmente, o sistema *Park Assist* está muito mais sofisticado, podendo estacionar sem a presença do motorista, como é o caso da BMW Série 7 que conta com o *Remote Control Parking* ilustrado na Figura 3, que através da chave do veículo, permite ao motorista apenas acionar um botão para que o veículo entre ou saia da garagem. (R7, 2015).

Figura 3 - *Park Assist* BMW Serie 7



Fonte: (RODRIGUEZ, 2018)

Para iniciar o estacionamento com o assistente, primeiro deve-se passar em baixa velocidade paralelo a vaga e acionar o botão de *Park Assist*, assim o sistema faz o cálculo se o veículo cabe na vaga desejada e os movimentos necessários para estacionar, então o motorista para o veículo paralelamente a frente à vaga e seleciona R no câmbio para engatar a ré, então o volante do veículo inicia os primeiros movimentos e o painel conta com avisos sonoros de proximidade e se o motorista deve parar o carro ou qual seleção de câmbio realizar, dessa forma o motorista deve estar atento apenas quanto aos pedais de freio e acelerador, qual seleção de câmbio realizar e nos avisos sonoros e visuais de proximidade.

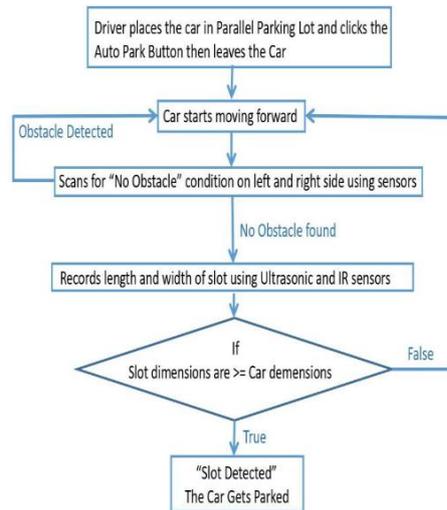
Vale citar que, para que o veículo possa realizar o estacionamento automático a direção do mesmo deve ser elétrica, para controlar adequadamente os movimentos que serão necessários para estacionar na vaga. O lado padrão de estacionamento é sempre do lado do passageiro. Porém, se o motorista desejar estacionar no seu lado, basta acionar a seta na direção da vaga quando o botão do *Park Assist* estiver acionado.

2.2 Projetos universitários sobre *Park Assist*

Alguns trabalhos universitários já realizaram um assistente de estacionamento em escala antes deste projeto, e assim como eles obtiveram sucesso nas metas estipuladas, eles serão usados como referenciais teóricos para este projeto.

A base teórica de quais etapas seguir desde o início do desenvolvimento físico deste projeto, teve como referência alguns projetos com o mesmo tema sobre estacionamento automático. O projeto *Automatic Parallel Car Parking* inicia a pesquisa demonstrando quais serão os componentes utilizados, suas funções e quais aplicações serão propostas no trabalho, demonstrando com *software* de montagem eletrônica a disposição física dos componentes. Então monta um fluxograma que está ilustrado na Figura 4 de como o processo irá acontecer, desde o carro paralelo a vaga até a finalização do estacionamento. Dessa forma a programação já tem uma trajetória pré-estabelecida para que o programa execute o estacionamento paralelo e no caso deste projeto o estacionamento perpendicular também. (GANESH, 2019).

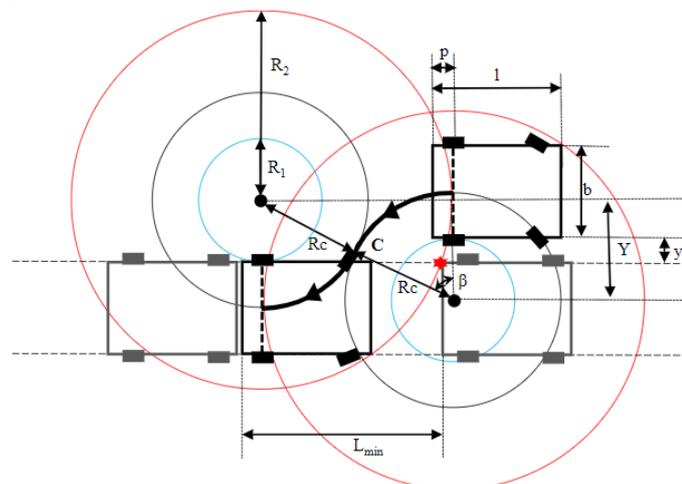
Figura 4 - Fluxograma projeto paralelo



Fonte: (GANESH, 2019)

Outro projeto que é referencial para este trabalho é o *Self Parking Robot*, que inicia sua pesquisa escrevendo sobre a geometria de Ackerman, um estudo físico para determinar os ângulos do sistema de direção com relação aos eixos de um veículo, para realizar a rotação das rodas da melhor maneira afim de realizar uma curva, além da posição dos pneus durante a curva. O projeto também analisa os movimentos necessários para o carrinho realizar o movimento de estacionamento na vaga paralelamente, cuja trajetória está ilustrada na Figura 5. Foram utilizados cálculos geométricos para determinar os raios externos e internos que os pneus dianteiros devem seguir para realizar o estacionamento. (HENRIKSSON, 2017).

Figura 5 - Raios internos e externos de uma baliza

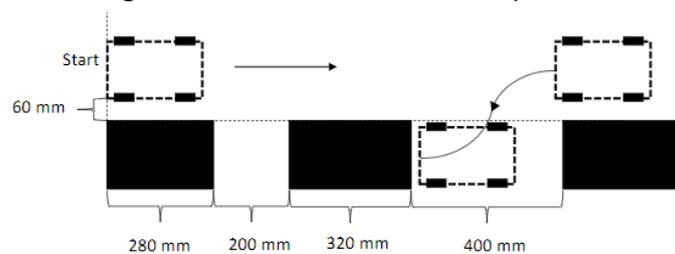


Fonte: (HENRIKSSON, 2017)

O projeto continua o desenvolvimento mencionando o conceito de PWM (*Pulse Width Modulation*), fator muito importante para a programação de microcontroladores. Um PWM é basicamente um sinal digital unipolar de onda quadrada onde a duração do tempo *ON* pode ser ajustada (ou modulada) conforme desejado. (IBRAHIM, 2014). Desta forma, a potência entregue à carga pode ser controlada a partir de um microcontrolador. Com uma base teórica apresentada, o projeto introduz cada componente a ser utilizado pelo protótipo e quais as funções aplicadas a eles, suas configurações para o projeto de estacionamento automático e as dimensões para a construção do *hardware*. Então se dá início a organização e execução do *software*, e assim como o projeto *Automatic Parallel Car Parking*.

O projeto monta um fluxograma de operações para checagem dos sensores, motores e controlador, para que a programação prossiga da maneira desejada. Após a organização, tanto do *software* quanto do *hardware*, o projeto inicia os testes para avaliar os resultados. No primeiro teste o veículo trafega paralelamente à 3 objetos, com espaço de 200 mm e 400 mm, como ilustra a Figura 6.

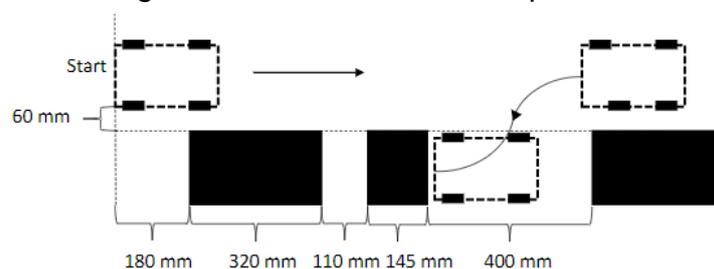
Figura 6 - Teste de sensores pista 1



Fonte: (HENRIKSSON, 2017)

O segundo teste ilustrado na Figura 7, repete o primeiro teste modificando o primeiro espaço para 110 mm, assim testando a precisão dos sensores.

Figura 7 - Teste de sensores pista 2



Fonte: (HENRIKSSON, 2017)

Segundo Henriksson (2017), 14 testes foram realizados, e dentro desses testes apenas 2 não obtiveram sucesso, sendo no primeiro caso o robô entrou na vaga mas ficou desalinhado, e no segundo caso não detectou a vaga para estacionar.

A pesquisa Construção de Automóvel para Simulação de Estacionamento (*Park Assist*) também utilizou a geometria de Ackerman para calcular as dimensões da vaga para realizar o movimento de estacionamento paralelo e os ângulos dos pneus para o movimento, em sequência são trabalhados todos os componentes utilizados e são realizados testes individuais para uma montagem física. Com a montagem finalizada, o projeto segue com o desenvolvimento do *software* e diferente dos outros projetos citados, o fluxograma de operações é realizado apenas após a montagem do projeto em si. As simulações são realizadas a fim de se obter uma acurácia do projeto, a qual foi alcançada em 83,33%, com uma observação de que no primeiro momento que o equipamento é ligado, a precisão é menor e vai melhorando com os testes subsequentes. (GIROTTI, 2018).

As informações para construção dos projetos acadêmicos analisados foram comparadas com este projeto, e podemos observar os dados obtidos na tabela 1.

Tabela 1 - Comparação de projetos

| <i>Automatic Parallel Car Parking</i> | <i>Self Parking Robot</i> | Construção de Automóvel para Simulação de Estacionamento (<i>Park Assist</i>) | Assistente de estacionamento automático |
|---------------------------------------|-----------------------------|--|--|
| Desenvolvimento do tema | Desenvolvimento do tema | Desenvolvimento do tema | Desenvolvimento do tema |
| Componentes | Geometria de Ackermann | Geometria de Ackermann | Trabalhos relacionados |
| Esquema montado digitalmente | Dimensões de vaga | Componentes | Componentes |
| Fluxograma do passo a passo | Componentes | Teste de componentes | Fluxograma básico |
| Testes práticos | Fluxograma do passo a passo | Montagem física | Esquema montado digitalmente |
| Resultados | Testes práticos | Software | Montagem Física |
| Conclusão | Resultados | Fluxograma básico | Testes |
| | Conclusão | Simulação | Resultados |
| | | Resultados | Conclusão |
| | | Conclusão | |

Fonte: Autoria própria (2022)

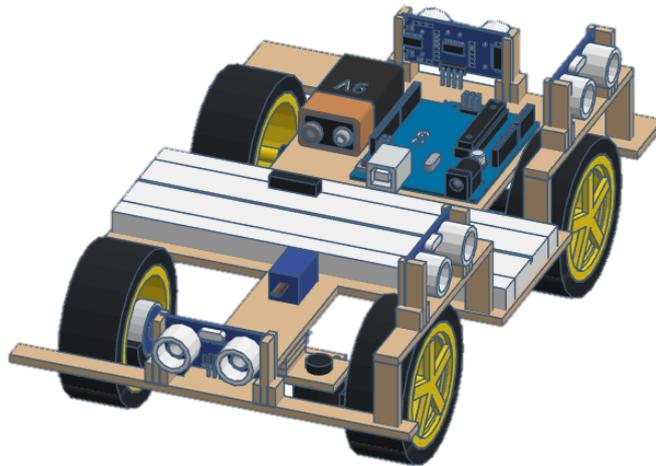
3 DESENVOLVIMENTO DO PROJETO

Este capítulo tem o objetivo de explicar o desenvolvimento do projeto sob o ponto de vista mecânico, elétrico, eletrônico e de *software*, fornecendo as bases necessárias para compreender o funcionamento do sistema e como ele foi implementado.

3.1 Montagem do chassi versão 1

A primeira ideia para construção do chassi foi montá-lo inteiramente em MDF (*Medium Density Fiberboard*) de espessura de 3 mm, por ser um material de baixo custo e fácil manuseio. Para isso, uma versão digital foi montada para que a visualização de como construir as peças em mdf fosse facilitada, e esse esquema digital conta com uma pré-visualização da posição dos componentes eletrônicos. O chassi em sua primeira versão está ilustrado na Figura 8.

Figura 8 - Chassi versão 1 com componentes



Fonte: Autoria própria (2022)

Após desenhar o chassi em sua primeira versão no site *TinkerCad*, o próximo passo é realizar o corte das chapas mdf na dimensão estabelecida no desenho. Os corte necessários foram realizados em uma serra de bancada e após cortadas foram unidas com cola instantânea. A Figura 9 ilustra o MDF cortado e colado onde foi necessário

Figura 9 - Chassi de MDF



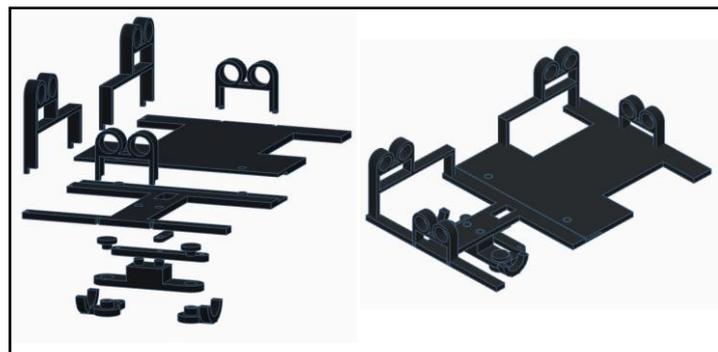
Fonte: Autoria própria (2022)

Porém, observou-se ao longo dos primeiros testes que o material não é resistente a umidade, tornando a estrutura do chassi muito instável e prejudicando o funcionamento do protótipo.

3.2 Montagem do chassi versão 2

Uma segunda solução encontrada foi construir o chassi em material PLA (*Poli-Latic Acid*), que é um polímero utilizado para impressão 3D. Portanto, o chassi foi desenhado também no *TinkerCad*, peça a peça pensando em uma montagem, justamente por conta da limitação da árvore de volumes de impressão na impressora 3D que foi usada. A Figura 10 ilustra uma vista explodida do chassi à esquerda e ele montado à direita.

Figura 10 - Chassi de PLA

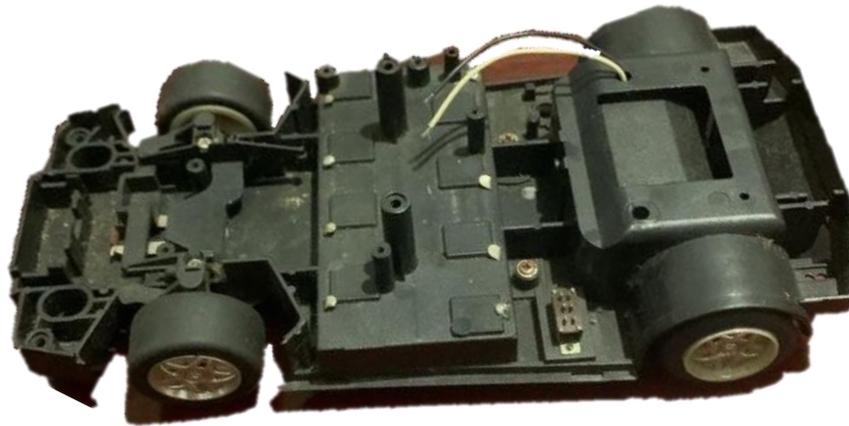


Fonte: Autoria própria (2022)

3.3 Montagem do chassi versão 3

O problema encontrado no chassi versão 2 foi a sua movimentação, que não conseguiu manter um percurso em linha reta e possuiu uma limitação quanto ao ângulo de direção máximo que as rodas poderiam oferecer que não foi suficiente para atingir o objetivo do projeto. A terceira solução foi utilizar o chassi de um carrinho de controle remoto que está ilustrado na Figura 11.

Figura 11 – Chassi versão 3



Fonte: Autoria própria (2022)

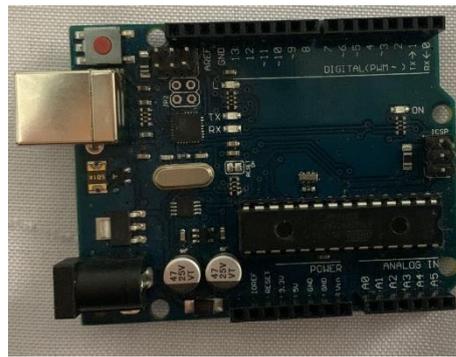
3.4 Componentes e suas funções

Os componentes utilizados no projeto Assistente de Estacionamento Automático são baseados no *Park Assist* de um veículo real. O *Park Assist* quando acionado visa identificar um espaço adequado para realização da manobra de estacionamento. Se o sistema identifica uma vaga com dimensão possível de se realizar o movimento de estacionamento, comanda a direção, freio e o acelerador para posicionar o veículo na vaga. O comando da posição do veículo depende de sensores ultrassônicos que enviam sinais para o módulo e o módulo comanda os pedais e o volante. Paralelamente o projeto utiliza os mesmos componentes, porém com uma complexidade muito menor.

3.4.1 Arduino

Assim como os veículos reais possuem o módulo de controle para processamento das informações dos sensores, o projeto conta com um microcontrolador responsável por coordenar as funções do projeto. O microcontrolador utilizado no presente projeto é o Arduino UNO, ilustrado na Figura 12. Ele será o responsável por receber sinais dos sensores e enviar sinais para os motores e controlar toda lógica do estacionamento.

Figura 12 - Arduino UNO



Fonte: Autoria própria (2022)

3.4.2 Sensor HC-SR04

Para identificar as distâncias em tempo real do veículo em relação à obstáculos, serão utilizados sensores ultrassônicos modelo HC-SR04, como ilustra a Figura 13. A escolha desses sensores se deve pela versatilidade de programação deles através do Arduino.

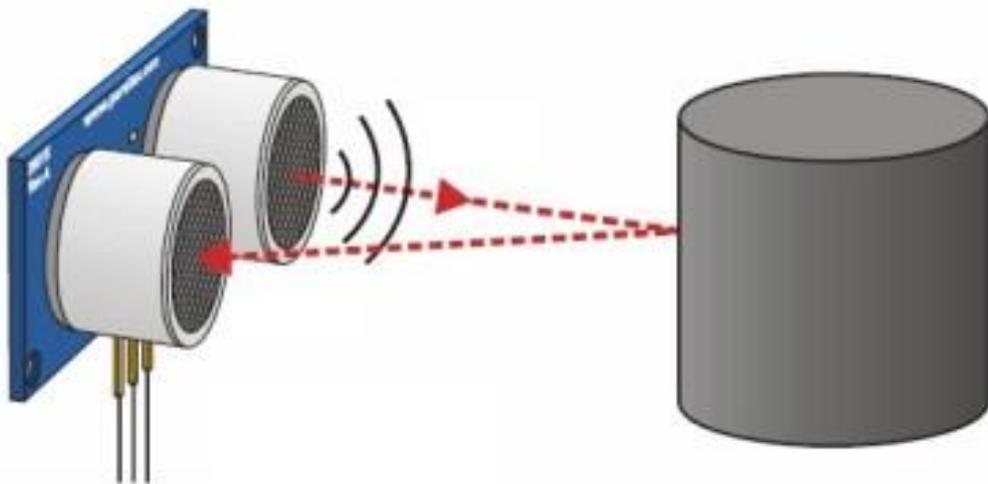
Figura 13 - Sensor ultrassônico HC-SR04



Fonte: (MOTA, 2017)

O sensor possui 4 pinos de ligação, como podemos observar acima, VCC, Trig, Echo e GND. Os pinos VCC e GND são a alimentação +5V e 0V respectivamente, o pino Trig envia pulso de comando para o transmissor emitir um sinal sonoro e o pino Echo recebe o sinal de eco do objeto. O sensor por ultrassom funciona por meio da emissão de um pulso sonoro de alta frequência que ao atingir um obstáculo ocorre um sinal eco que retorna ao sensor. Dessa forma, possuindo as informações do tempo de emissão, tempo de recepção do eco, meio de propagação e velocidade do som nesse meio de propagação, o programa consegue calcular a distância entre o sensor e o objeto. A Figura 14 ilustra simplificada o funcionamento desse sensor de ultrassom.

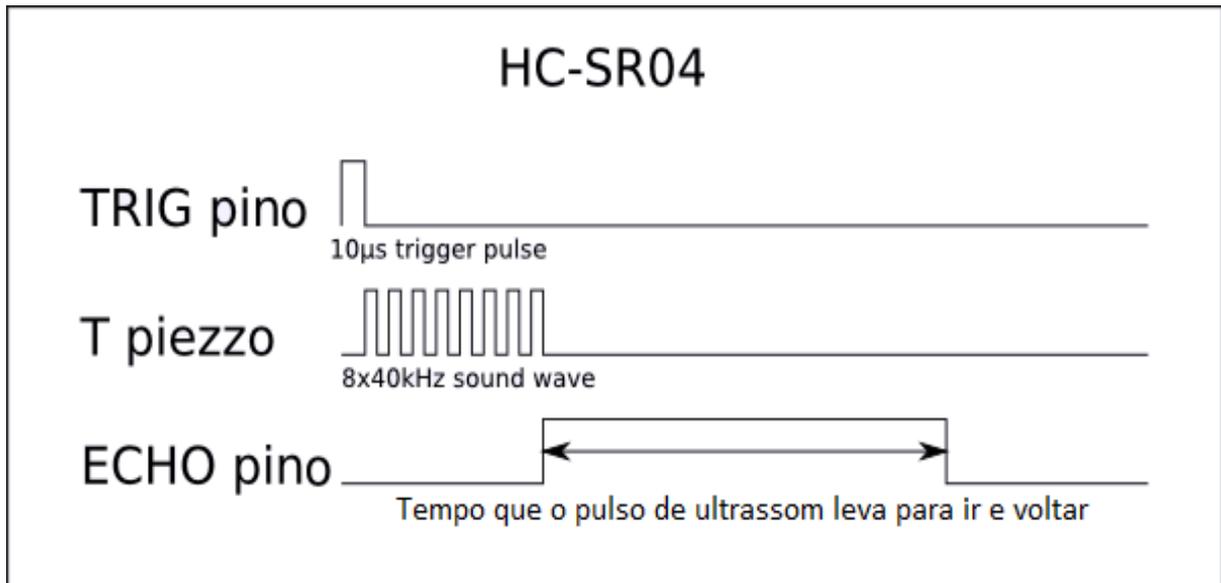
Figura 14 - Funcionamento do sensor



Fonte: (MOTA, 2017)

O HC-SR04 funciona da seguinte maneira: o pino Trig ao receber um pulso de 5V de pelo menos 10 microssegundos. Em seguida ele envia 8 pulsos ultrassônicos de 40 kHz, no mesmo instante do pulso. O programa aciona também um temporizador, que contará o tempo até a recepção do sinal do eco, que será sinalizado pelo pino Echo. Quando o sinal retornar ao sensor, o pino Echo fornecerá um sinal de borda de descida para o Arduino. Portanto, o tempo que o pino Echo fica acionado é proporcional à distância do objeto para o sensor, e todo esse processo está exemplificado no gráfico da Figura 15.

Figura 15 - Gráfico do sinal ultrassônico do HC-SR04

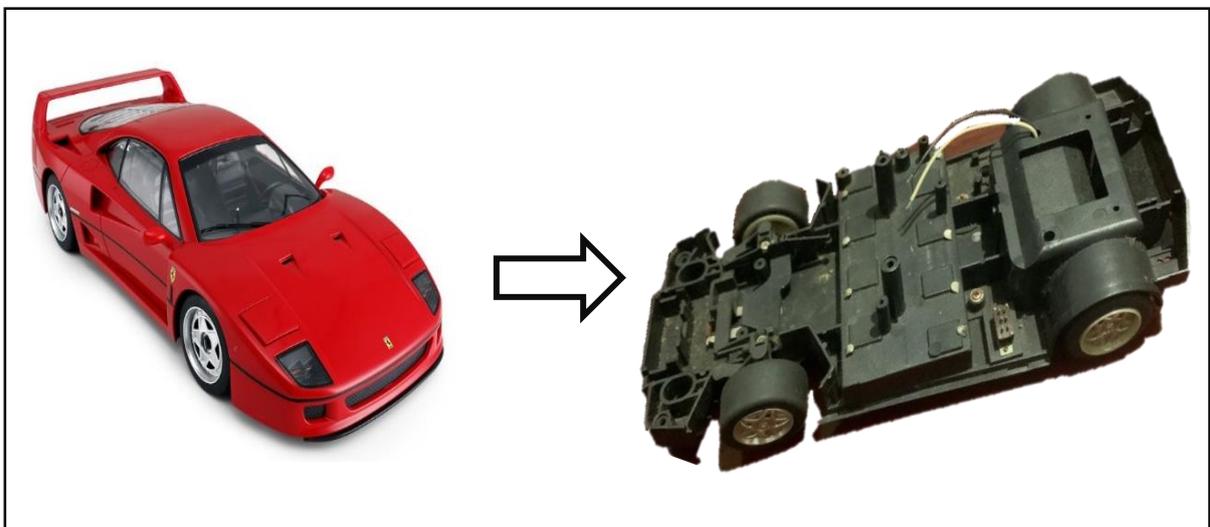


Fonte: (MOTA, 2017)

3.4.3 Chassi de base

O veículo protótipo final utiliza o chassi de um carrinho RF (Rádio Frequência) de uma Ferrari F40 conforme ilustrado na Figura 16, para aproveitar o sistema de tração das rodas, direção, suporte de pilhas e o motor DC.

Figura 16 - Chassi da Ferrari F40 RF

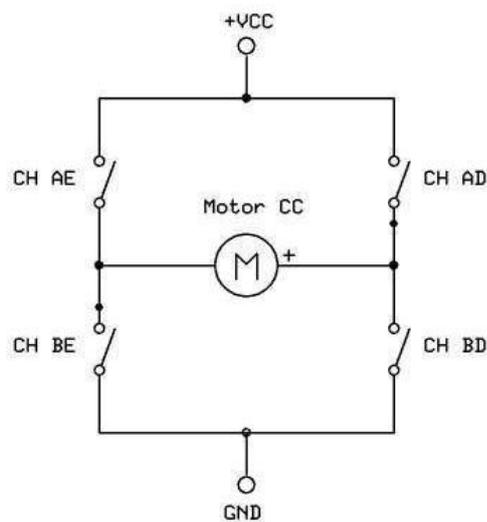


Fonte: Autoria própria (2022)

3.4.4 Driver para ponte H

De acordo com a proposta do projeto, o veículo se movimenta para frente e para trás, e para isso é necessária uma forma de inverter a polaridade do motor. Uma opção para se inverter a polaridade do motor DC do projeto é utilizar uma ponte H. A ponte H utiliza o corte e condução de transistores para realizar um chaveamento entre as saídas, enviando o sinal positivo para uma saída e negativo para a outra, fazendo com que o motor gire em um sentido. Invertendo a polaridade das saídas com os transistores, o motor gira no sentido oposto. A Figura 17 ilustra o processo de chaveamento da ponte H. Note que os transistores foram substituídos por chaves mecânicas, apenas para facilitar a compreensão do conceito.

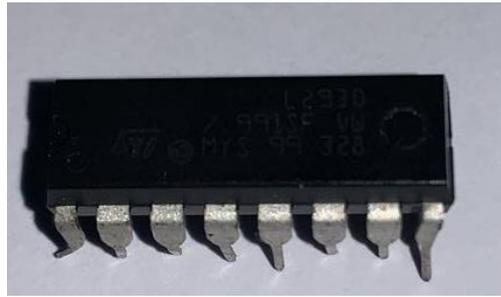
Figura 17 - Conceito de ponte H



Fonte: (ELETROGATE, 2019)

Conceitualmente, o chaveamento mencionado anteriormente consiste em comutar os contatos diagonais da representação acima, comutando as chaves AE e BD, a corrente passa pelo motor da esquerda para a direita, girando o motor em um sentido. Comutando as chaves AD e BE, a corrente passa pelo motor da direita para a esquerda, girando o motor no sentido oposto. Contudo, a ponte H real usada nesse projeto é o driver L293D que está ilustrado na Figura 18.

Figura 18 - Driver L293D



Fonte: Aatoria Própria (2022)

3.4.5 Servo motor

Para realizar os movimentos de direção das rodas, o componente utilizado é um servo motor MG996R, ilustrado na Figura 19. O servo motor funciona através do sinal PWM enviado pelo microcontrolador. O movimento do motor interno posiciona um potenciômetro que está acoplado no eixo de saída do servo. Esse potenciômetro é o responsável pela garantia da posição angular do servo motor. O servo motor também possui uma redução de engrenagens no seu eixo de saída para diminuir a rotação e aumentar o torque do motor.

Figura 19 - Servo motor MG996R

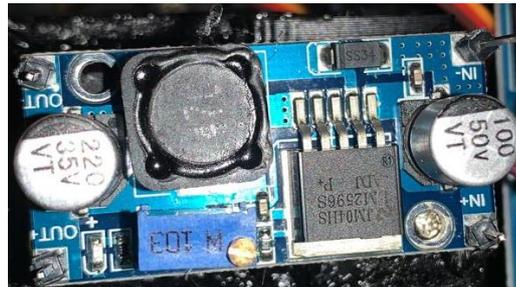


Fonte: (SERMAKER, 2022)

3.4.6 Módulo conversor de tensão DC/DC

Para um servo motor de baixo torque, somente a alimentação 5V do Arduino já é suficiente. Porém, como o servo utilizado é de alto torque, se for ligado diretamente na saída do Arduino, ele drena muita corrente e acaba resetando o microcontrolador. Para contornar esse problema, foi utilizado um conversor de tensão DC/DC LM2596, ilustrado na Figura 20, ligado diretamente a alimentação externa de 12V, alimentando o servo paralelamente, evitando o problema descrito acima.

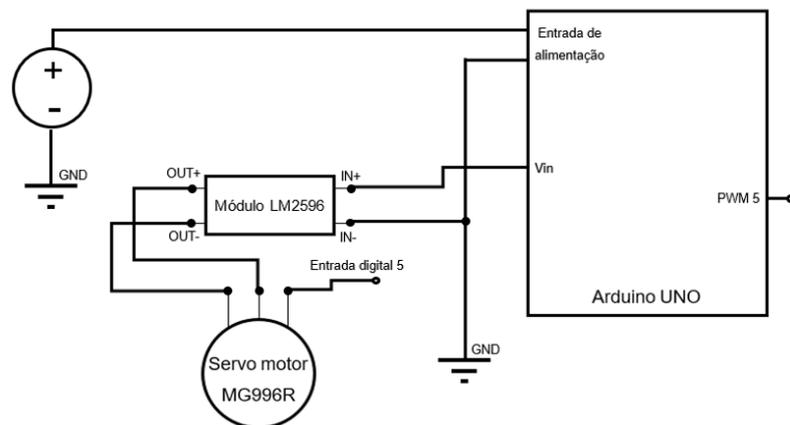
Figura 20 - Conversor LM2596



Fonte: Autoria própria (2022)

O conversor conta com 2 entradas (IN+ e IN-) e 2 saídas (OUT+ e OUT-), as entradas recebem 12V da fonte e as saídas são ligadas no servo motor enviando 5-6V necessários para ele se movimentar. O esquema de ligação do conversor pode ser visto na Figura 21.

Figura 21 - Esquema de ligação do conversor

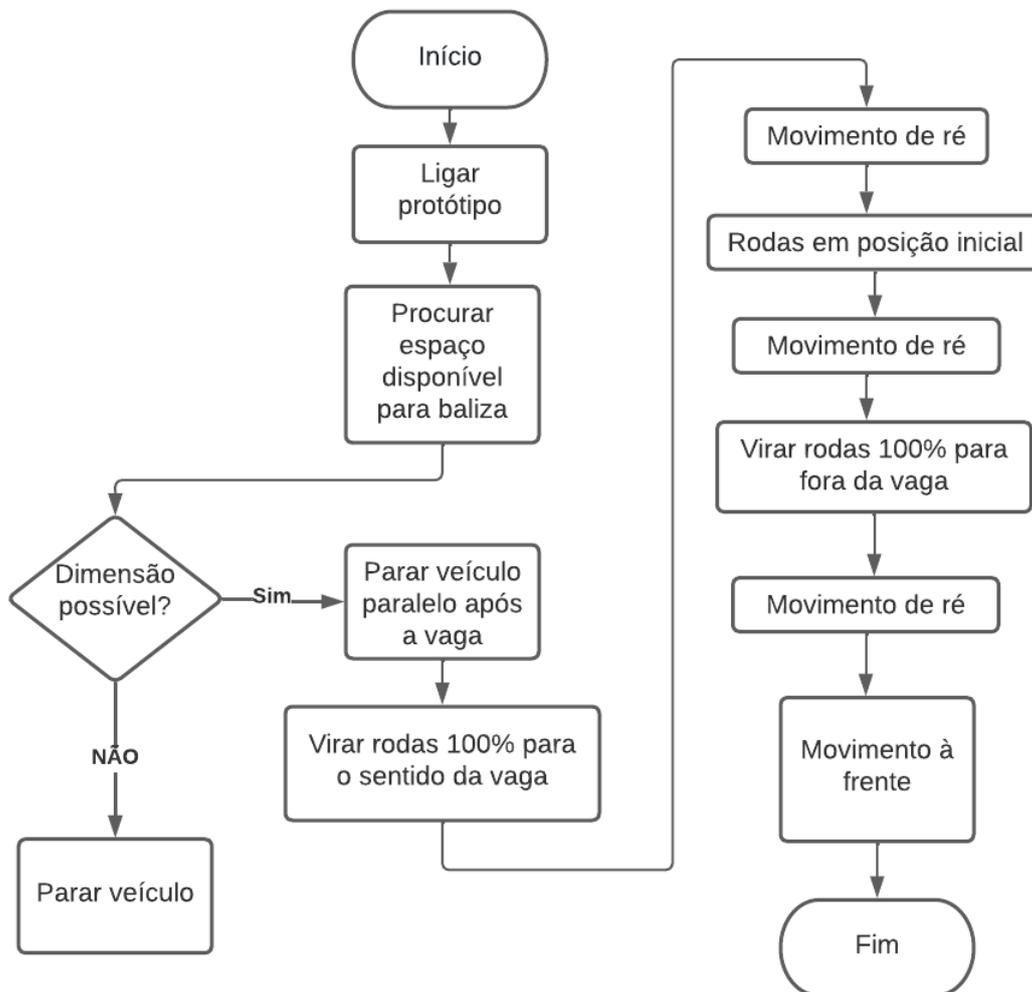


Fonte: Autoria própria (2022)

3.5 Fluxograma básico

Para que a programação dos passos a serem executados no projeto sejam visualizados de uma maneira geral, foi construído um fluxograma para facilitar a análise do processo como um todo. O fluxograma está ilustrado na Figura 22 .

Figura 22 - Fluxograma de operações

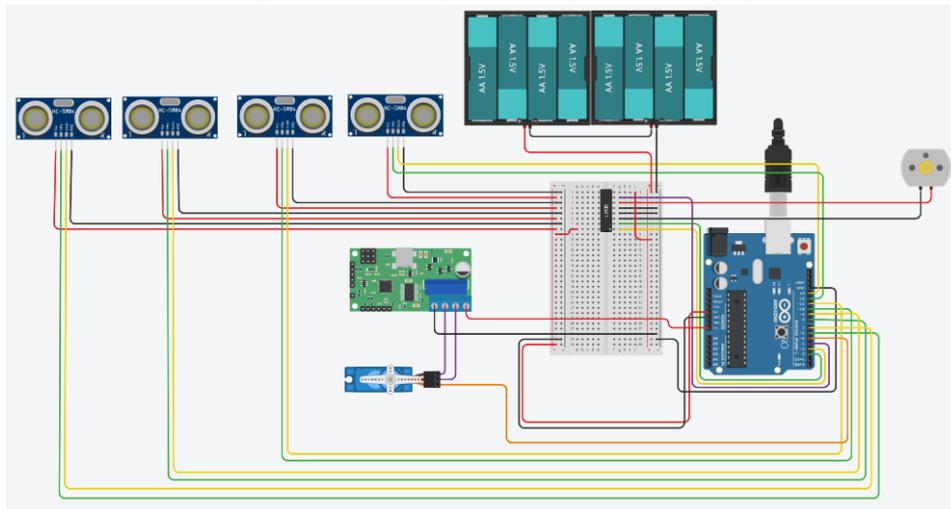


Fonte: Autoria própria (2022)

3.6 Montagem física

Para iniciar a montagem física dos componentes, primeiramente foi montado um esquema digital das ligações dos componentes e suas entradas com o Arduino, o esquema digital está ilustrado na Figura 23.

Figura 23 - Esquema digital das ligações



Fonte: Autoria própria (2022)

Após a organização das ligações, o próximo passo é determinar o posicionamento dos componentes no chassi, para isso foi montado um *layout* de componentes pensando nas duas primeiras versões de chassi, o *layout* está ilustrado na Figura 24.

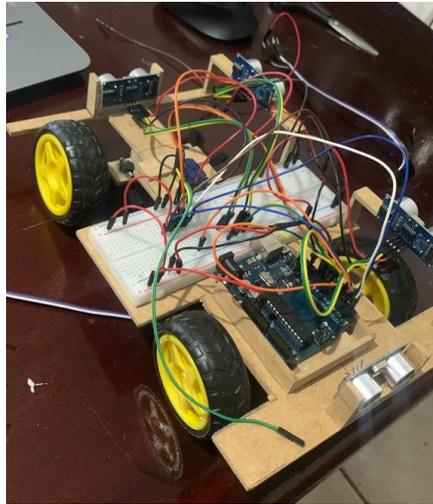
Figura 24 - *Layout* de componentes



Fonte: Autoria própria (2022)

Tanto o primeiro chassi de MDF quanto o segundo de material PLA impresso, se basearam no *layout* de componentes da Figura 24, posicionando os componentes, e usando as ligações elétricas da Figura 23. O primeiro protótipo montado está ilustrado na Figura 25.

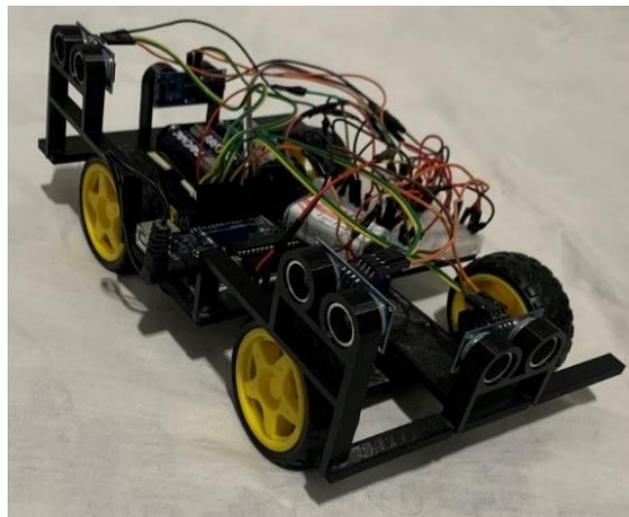
Figura 25 - Primeiro protótipo montado



Fonte: Autoria própria (2022)

O chassi de MDF não foi resistente a umidade, e o segundo chassi de PLA foi montado colando as peças impressas separadamente e seguindo as posições dos componentes do *layout* e o esquema elétrico. O segundo protótipo totalmente montado está ilustrado na Figura 26.

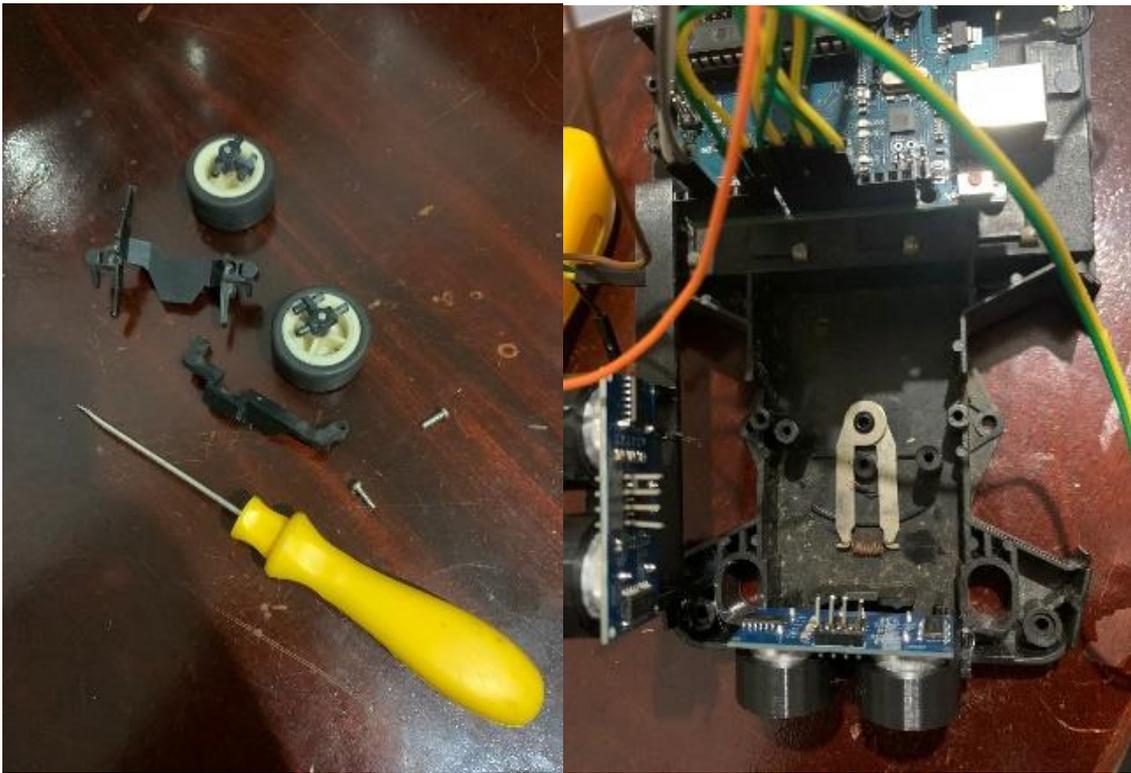
Figura 26 - Segundo protótipo montado



Fonte: Autoria própria (2022)

O problema encontrado do segundo protótipo foi a sua movimentação, que não conseguiu manter um percurso em linha reta e possuiu uma limitação quanto ao ângulo de direção máximo que as rodas poderiam oferecer que não foi suficiente para atingir o objetivo do projeto. A terceira solução foi utilizar o chassi de um carrinho de controle remoto. Porém para utilizar o servo motor no chassi do carrinho de controle remoto, um furo na barra de ligação entre as duas rodas foi realizado, a adaptação na barra está ilustrada na Figura 27.

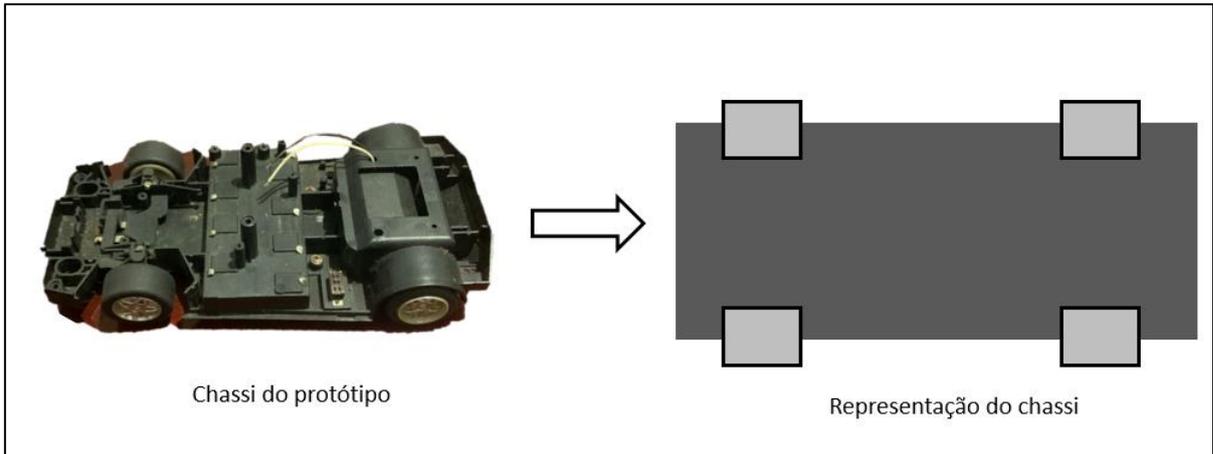
Figura 27 - Adaptação na barra de ligação



Fonte: Autoria própria (2022)

Como o chassi do protótipo final possui dimensões diferentes dos dois primeiros chassis protótipos, a criação de um novo *layout* dos componentes foi necessária. Para isso, uma representação do chassi final foi montada a partir de figuras básicas. A representação do novo chassi está ilustrada na Figura 28.

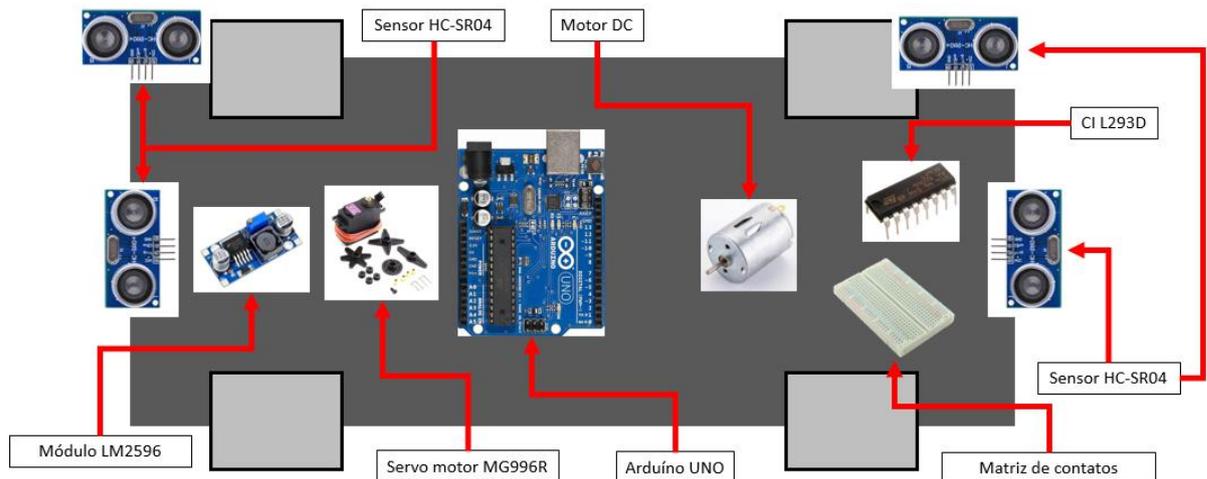
Figura 28 - Representação do chassi



Fonte: Autoria própria (2022)

A partir dessa representação, fica mais claro o posicionamento dos componentes no *layout*, o novo *layout* dos componentes está ilustrado na Figura 29.

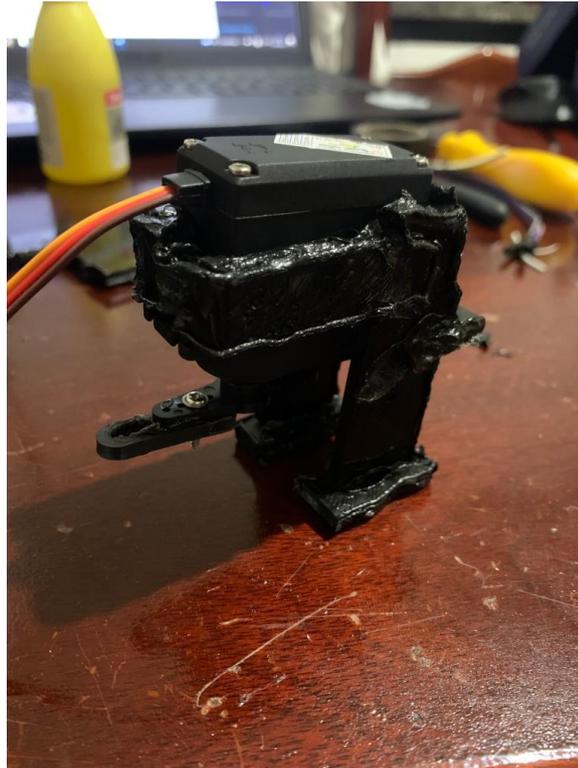
Figura 29 - Novo layout dos componentes



Fonte: Autoria própria (2022)

Além da adaptação na barra de ligação, um suporte provisório para o servo motor foi construído a partir de retalhos de material PLA, sendo fixado com cola quente no chassi, o suporte está representado na Figura 30.

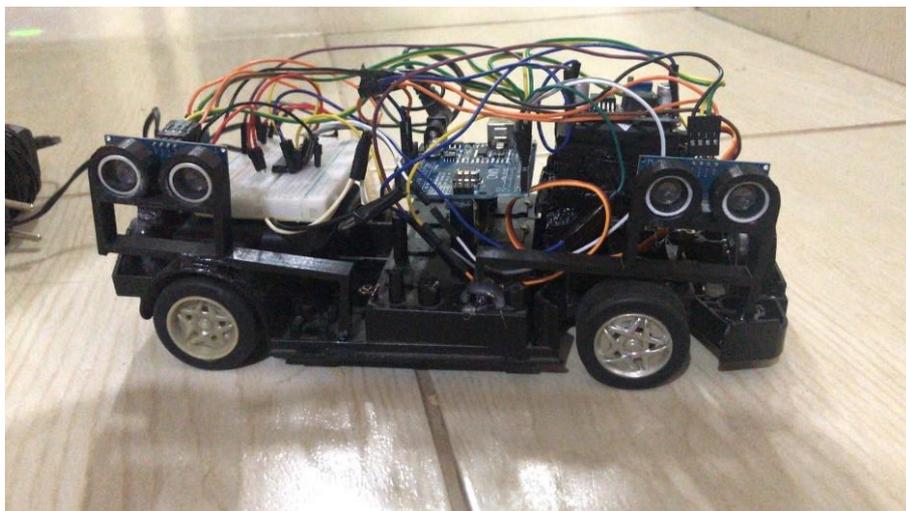
Figura 30 - Suporte provisório do servo motor



Fonte: A autoria própria (2022)

Com a organização de componentes seguindo o novo *layout*, o protótipo está pronto para montagem elétrica dos componentes, também seguindo as ligações realizadas no esquema digital. Assim a montagem do protótipo versão 3 e final está pronta e ilustrada na figura 31.

Figura 31 - Protótipo final montado



Fonte: A autoria própria (2022)

Com o protótipo montado, o último passo para iniciar os testes práticos é a montagem da pista de testes, e para isso devemos ter noção das dimensões do carrinho para a construção da vaga. De acordo com o CONTRAN (Conselho Nacional de Trânsito), a dimensão da vaga para baliza é o tamanho do veículo acrescido de 40%, como ilustrado na Figura 32. Porém, o objetivo do projeto não é atingir uma acurácia de um veículo real no primeiro instante, assim o projeto utiliza um espaçamento maior.

Figura 32 - Dimensões da balliza



Fonte: (CARDOSO, 2020)

Portanto, como a dimensão do carrinho protótipo é de 150 mm de largura e 265 mm de comprimento, a largura para a vaga ideal seria de 210 mm e o comprimento de 371 mm. Porém, foram utilizados folhas de papelão cortadas com estilete para construir duas estruturas para simular as distâncias desejadas da vaga, as folhas de papelão cortadas estão ilustradas na Figura 33.

Figura 33 - Papelão para pista de testes



Fonte: Autoria própria (2022)

Para facilitar os testes, apenas duas estruturas de simulação de vaga foram confeccionadas, sendo possível adequar o tamanho da vaga da maneira desejada, hora em um espaço que seja possível estacionar e outra em um espaço reduzido, para comprovar o objetivo do projeto. O espaço utilizado para estacionar fica entre as duas estruturas criadas para a vaga, as estruturas criadas para dar o espaço da vaga estão ilustradas na Figura 34.

Figura 34 – Estruturas para pista de testes



Fonte: Autoria própria (2022)

4 TESTES E RESULTADOS

Este capítulo tem o objetivo de utilizar todas as partes físicas juntas e aplicar a programação necessária para que o projeto alcance o objetivo estabelecido.

4.1 Teste dos motores com programação

O primeiro teste realizado foi o teste da rotação dos motores nos dois sentidos e a direção do veículo para a esquerda e direita através de programação. Para realizar essa função foi utilizado o código detalhado e comentado na Figura 35.

Figura 35 - Programação do motor e direção

```
ang=110; //ângulo determinado para direita
servo.write(ang); // escrita do ângulo no servo

delay(1000); // tempo de 1 segundo

vel = 190; // rpm do motor
analogWrite(velmotor,vel); // saída analógica para o motor
digitalWrite(hor,LOW); // Marcha a frente desligada
digitalWrite(ant,HIGH); //Marcha ré ligada
delay(500);
```

Fonte: Autoria própria (2022)

4.2 Teste de vaga com dimensão correta

Após aferir a funcionalidade dos motores, o próximo teste é conferir se a vaga está na dimensão possível para realizar o movimento de estacionamento paralelo, e para isso, o veículo é posicionado antes da vaga, e se movimenta a frente, utilizando os sensores laterais para medir a vaga. Se a dimensão da vaga for possível, o veículo avança até ficar paralelo após a vaga e inicia o movimento de estacionamento, até ficar encaixado na vaga, como ilustra a Figura 36.

Figura 36 - Primeiro teste de vaga na dimensão correta



Fonte: Autoria própria (2022)

4.3 Teste de vaga com dimensão incorreta

Assim como descrito, o veículo utiliza os sensores laterais para medir a vaga, e caso a dimensão não seja possível, o veículo deve parar assim que detectar que a vaga não possui a dimensão ideal como ilustra a Figura 37.

Figura 37 - Vaga de dimensão incorreta



Fonte: Autoria própria (2022)

Os testes de programação são baseados inteiramente nessas duas posições de vagas, então ao obter sucesso em estacionar na vaga de dimensão possível e evitar realizar o movimento na vaga de dimensão incorreta, o projeto atingirá com êxito o objetivo estabelecido no início de todo trabalho.

4.4 Resultados do testes

Os testes são realizados para atingir o objetivo através da tentativa e erro, e no caso do projeto Assistente de Estacionamento Automático, os erros serviram para aperfeiçoamento do projeto, como o caso citado do chassi deteriorado pela umidade. Outra interferência no desenvolvimento do projeto foi a duração da bateria 9V, que por testar os motores repetidas vezes descarregou quatro vezes, dessa forma, a solução para alimentar o projeto foi ligar o protótipo diretamente na tomada com ilustra a Figura 38.

Figura 38 - Alimentação direta na tomada



Fonte: A autoria própria (2022)

O primeiro teste foi realizar o movimento de estacionamento paralelo em uma vaga de dimensão correta, posicionada pelas estruturas de papelão, e o resultado obtido após 10 repetições com a correta programação está descrito na Tabela 2.

Tabela 2 - Resultado do teste 1

| Conclusão | Repetição |
|-----------------------------|------------------|
| Veículo fora de alinhamento | 1 |
| Baliza OK | 2 |
| Baliza OK | 3 |
| Baliza OK | 4 |
| Baliza OK | 5 |
| Veículo fora de alinhamento | 6 |
| Baliza OK | 7 |
| Baliza OK | 8 |
| Baliza OK | 9 |
| Veículo não acionou servo | 10 |

Fonte: Autoria própria (2022)

Seguindo com o segundo teste da vaga de dimensão incorreta, os dados obtidos com o teste estão descritos na Tabela 3.

Tabela 3 - Resultado do teste 2

| Conclusão | Repetição |
|--------------------------|------------------|
| Não realizou o movimento | 1 |
| Não realizou o movimento | 2 |
| Não realizou o movimento | 3 |
| Não realizou o movimento | 4 |
| Não realizou o movimento | 5 |
| Não se movimentou | 6 |
| Não realizou o movimento | 7 |
| Não realizou o movimento | 8 |
| Não realizou o movimento | 9 |
| Não realizou o movimento | 10 |

Fonte: Autoria própria (2022)

Observando os dados das duas tabelas, a precisão do projeto no primeiro teste foi de 70% de aproveitamento e do segundo teste em 90 % de aproveitamento, totalizando uma média de 80% de aprovação. Os problemas que impediram o protótipo de atingir 100% de aproveitamento são o alinhamento e o mal contato dos fios com os componentes.

5 CONCLUSÃO

A partir dos fatos apresentados nesse trabalho, podemos observar que para um projeto atingir o sucesso, o mesmo precisa interagir com outros projetos que possuem um tema próximo para servir de base de pesquisa, e além disso os conhecimentos necessários são obtidos de forma gradual no decorrer do curso, além de conhecimentos adjacentes que são adquiridos a partir das pesquisas em websites e artigos científicos já publicados. Em relação ao projeto de estacionamento automático, o desafio consiste em organizar a parte física do projeto, no caso específico deste projeto, entender de processos mecânicos para montagem do chassi, além do conhecimento automobilístico, o software montado inspirado em temas paralelos já existentes trabalhando em conjunto com a parte física descreve exatamente o objetivo de obter o título de tecnólogo em Mecatrônica Industrial, que une a eletrônica com o físico que executa um trabalho.

6 PROPOSTAS FUTURAS

Como o projeto pensado para este trabalho existe em escala maior e em produção de veículos reais, para a melhoria futura desse projeto, podemos listar alguns tópicos:

- Aumentar a dificuldade do objetivo diminuindo o espaço de estacionamento para provar a melhoria de como funcionam os sensores.
- Aumentar a escala do projeto para 1:1 e analisar os problemas atuais de um *Park Assist* para buscar soluções em novas pesquisas.
- Iniciar o movimento fora da vaga para que o projeto se aproxime do espaço disponível automaticamente.
- Substituir a alimentação por fonte AC/DC por uma bateria de Li-Po (Polímero de íons de Lítio) para o projeto funcionar independente da tomada.
- Utilizar sensores ultrassônicos de melhor precisão para solucionar o problema de posicionamento nos testes.
- Substituir a matriz de contatos por conexões soldadas para evitar o mal contato dos componentes.

REFERÊNCIAS

CARDOSO, Ronaldo. **Como fazer uma baliza perfeita em 10 passos**. [S. l.], 12 maio 2020. Disponível em: <https://www.autoescolaonline.net/como-fazer-uma-baliza-perfeita-em-10-passos/>. Acesso em: 17 ago. 2022.

DETRAN (São Paulo). **DETRAN.SP DÁ DICAS PARA O CANDIDATO GANHAR NOTA 10 NO EXAME PRÁTICO DE HABILITAÇÃO**. [S. l.], 23 fev. 2022. Disponível em: [https://www.detran.sp.gov.br/wps/portal/portaldetran/cidadao/noticias/detalhes/c257cf4a-7a13-48fd-9466-7832f2b78916!/ut/p/z0/jU27DolwFP0WB8fmloJQRk-KiRtSExAG7mEtLsUpaxMbH34smrsbnpJPzAgEVCIs306l3zml36r2ID6uS5sslZ2u-Xcxpud6FRUQXBS0orED8DowL5nS5iAyEdNY3Dw9V7waPnWr8gHZKvviNQo-VuSq3zRhq8fizsjs3lJJsIUkdIEgxCEnGtSBrFMUI4yDSrE54G8fuLDZt804Lo0R-JsdpB9Ve1P4v6ec8mL3Gq6NY!/#:~:text=Para%20essa%20etapa%2C%20cada%20candidato,na%20E%20de%209%20min](https://www.detran.sp.gov.br/wps/portal/portaldetran/cidadao/noticias/detalhes/c257cf4a-7a13-48fd-9466-7832f2b78916!/ut/p/z0/jU27DolwFP0WB8fmloJQRk-KiRtSExAG7mEtLsUpaxMbH34smrsbnpJPzAgEVCIs306l3zml36r2ID6uS5sslZ2u-Xcxpud6FRUQXBS0orED8DowL5nS5iAyEdNY3Dw9V7waPnWr8gHZKvviNQo-VuSq3zRhq8fizsjs3lJJsIUkdIEgxCEnGtSBrFMUI4yDSrE54G8fuLDZt804Lo0R-JsdpB9Ve1P4v6ec8mL3Gq6NY!/#:~:text=Para%20essa%20etapa%2C%20cada%20candidato,na%20E%20de%209%20min.). Acesso em: 17 ago. 2022.

ELETROGATE (Brasil). **Como Montar uma Ponte H com Relés**. [S. l.], 19 maio 2019. Disponível em: <https://blog.eletrogate.com/arduino-ponte-h-com-rele/>. Acesso em: 17 ago. 2022.

ELETROGATE (Brasil). **Sensor Ultrassônico HC-SR04 com Arduino**. [S. l.], 8 jul. 2017. Disponível em: <https://blog.eletrogate.com/sensor-ultrassonico-hc-sr04-com-arduino/>. Acesso em: 12 nov. 2021.

EMILIA, Santa. **PARK ASSIST: O QUE É E COMO FUNCIONA?** [S. l.], 7 out. 2020. Disponível em: <https://blog.santaemilia.com.br/2020/10/07/park-assist-o-que-e-e-como-funciona/>. Acesso em: 29 out. 2021.

GANESH, Soma; GANESH, V. Sai; HAMZA, Mir; RAMWAL, Sahil; GHOSH, Siddhartha. Automatic Parallel Car Parking System using Sensors and Arduino UNO. **International Journal of Recent Technology and Engineering (IJRTE)**, [S. l.], p. 1-4, 11 set. 2019.

HENRIKSSON, Lovisa; LUNDELL, Victor. **Self Parking Robot**. 2017. Trabalho de Conclusão de Curso (Bacharelado) - KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT, [S. l.], 2017.

IBRAHIM, Dogan. **Pulse Width Modulation**: Pulse width modulation (PWM) provides a way of controlling certain analog quantities, by varying the pulse width of a fixed frequency rectangular waveform.. [S. l.], 1 jan. 2014. Disponível em: <https://www.sciencedirect.com/topics/engineering/pulse-width-modulation>. Acesso em: 28 jul. 2022.

MONTOIA, Vinicius. **Não gosta de manobrar? Então conheça o assistente de estacionamento**: Também conhecido como Park Assist, recurso utiliza sensores para colocar e retirar o carro da vaga. [S. l.], 24 mar. 2021. Disponível em: <https://www.auto.com.br/nao-gosta-de-manobrar-entao-conheca-o-assistente-de-estacionamento/>. Acesso em: 10 set. 2021.

MOTA, Allan. **HC-SR04 – Sensor Ultrassônico de distância com Arduino**. [S. l.], 14 maio 2017. Disponível em: <https://portal.vidadesilicio.com.br/hc-sr04-sensor-ultrassonico/>. Acesso em: 28 out. 2021.

O'CONNELL, Jackson. **PLA Plastic / Material: The Ultimate Guide**. [S. l.], 24 jul. 2021. Disponível em: <https://all3dp.com/2/what-is-pla-plastic-material-properties/>. Acesso em: 9 jun. 2022.

PORTAL AUTO SHOPPING. **Baliza: passo a passo para não errar nunca mais!**. [S. l.], 27 jul. 2018. Disponível em: <https://www.portalautoshopping.com.br/blog/baliza-passo-a-passo/>. Acesso em: 17 ago. 2022.

R7 (Brasil). **Novo BMW Série 7 estaciona sozinho com sistema inédito controlado pela chave e pelo smartphone**: Além do estacionamento autônomo, sedã mais caro da marca bávara terá opção de faróis a laser. [S. l.], 13 jun. 2015. Disponível em: <https://noticias.r7.com/carros/fotos/novo-bmw-serie-7-estaciona-sozinho-com-sistema-inedito-controlado-pela-chave-e-pelo-smartphone-13062015>. Acesso em: 27 set. 2021.

RASHID, Mamunur; RAHMAN, Mirza; ISLAM, Rashidul; ALWAHEDY, Omar. Autonomous 4WD Smart Car Parallel Self-Parking System by Using Fuzzy Logic Controller. **American International Journal of Sciences and Engineering Research**, [S. l.], p. 2-31, 1 fev. 2019.

REDAÇÃO BLOGAUTO. **LANÇADO PELA VW, PARK ASSIST CELEBRA UMA DÉCADA**. [S. l.], 3 jan. 2017. Disponível em: <https://www.blogauto.com.br/park-assist-celebra-uma-decada/>. Acesso em: 17 ago. 2022.

REIS, Thiago. **Com aumento da frota, país tem 1 automóvel para cada 4 habitantes**: São Caetano, SP, é a cidade com mais carros por habitante: 2 para cada 3. Porto Walter, no AC, é a única das 5.570 que não possui automóvel.. [S. l.], 10 mar. 2014. Disponível em: <https://g1.globo.com/brasil/noticia/2014/03/com-aumento-da-frota-pais-tem-1-automovel-para-cada-4-habitantes.html>. Acesso em: 17 ago. 2022.

RODRIGUEZ, Henrique. **BMW Série 7 ganha estacionamento por controle remoto no Brasil Leia mais em: <https://quatorrodas.abril.com.br/noticias/bmw-serie-7-ganha-estacionamento-por-controle-remoto-no-brasil/>**: Motorista pode tirar e colocar o carro na vaga por comando na chave Leia mais em: <https://quatorrodas.abril.com.br/noticias/bmw-serie-7-ganha-estacionamento-por-controle-remoto-no-brasil/>. [S. l.], 29 out. 2018. Disponível em: <https://quatorrodas.abril.com.br/noticias/bmw-serie-7-ganha-estacionamento-por-controle-remoto-no-brasil/>. Acesso em: 17 ago. 2022.

SERMAKER. **SERVO MOTOR MG996R**. [S. l.], 17 maio 2022. Disponível em: <https://www.sermaker.com/servo-motor-mg996r>. Acesso em: 17 ago. 2022.

SINDAUTOESCOLA.SP (São Paulo). **Delimitação da vaga de baliza para exame prático na categoria**. [S. l.], 17 nov. 2015. Disponível em: <https://sp.sindautoescola.org.br/noticias/item/detrان/161>. Acesso em: 12 nov. 2021.

SILVEIRA, Cristiano. **O que é PWM e Para que Serve?**. [S. l.], 26 nov. 2016. Disponível em: <https://www.citisystems.com.br/pwm/>. Acesso em: 24 nov. 2021.

SOS AUTOMÓVEL (Brasil). **Como fazer uma baliza perfeita**: Vi essa dica numa rede social e achei bacana tinha até um vídeo do auto esporte confira. [S. l.], 29 jul. 2013. Disponível em: <http://sosautomovel.blogspot.com/2013/07/como-fazer-uma-baliza-perfeita.html>. Acesso em: 18 nov. 2021.

TECHTUDO (Brasil). **Li-Ion ou Li-Po? Conheça as diferenças dos tipos de bateria de celular:** Capacidade de armazenamento, tamanho, peso e preço estão entre os itens que diferenciam os dois tipos de bateria.. [S. l.], 5 ago. 2016. Disponível em: <https://www.techtudo.com.br/noticias/2016/08/li-ion-ou-li-po-conheca-diferencas-dos-tipos-de-bateria-de-celular.ghtml>. Acesso em: 6 jul. 2022.

TOUME, Wilson. **Inovação # 10: Assistentes de estacionamento:** Conheça as 50 inovações que estão transformando a relação entre pessoas e carros. [S. l.], 28 jul. 2017. Disponível em: <https://autoesporte.globo.com/carros/noticia/2017/07/inovacao-10-assistentes-de-estacionamento.ghtml>. Acesso em: 17 ago. 2022.

APÊNDICE A – Programação do Arduíno

```

#include <Servo.h>
#define pinServo 4 //define pwm do servo no pino 5
#define velmotor 3 // define controle de velocidade no pino 3
#define hor 5 // sentido de rotação horária no pino 4
#define ant 2 // sentido de rotação anti-horária no pino 2

int vel = 0; //variavel para controle de velocidade
int estado; // variável da máquina de estados
int T1 = 8; // definir entrada e saídas dos sensores(ECHO e TRIGGER)
int E1 = 9;
int T2 = 6;
int E2 = 7;
int T3 = 10;
int E3 = 11;
int T4 = 12;
int E4 = 13;

long dur1; // variáveis para cálculo de distância por tempo
long dis1;
long dur2;
long dis2;
long dur3;
long dis3;
long dur4;
long dis4;

Servo servo1;
int ang; // variável para ângulo de servo motor

void setup() {

    pinMode(T1,OUTPUT); //define pinos dos sensores
    pinMode(E1,INPUT);
    pinMode(T2,OUTPUT);
    pinMode(E2,INPUT);
    pinMode(T3,OUTPUT);
    pinMode(E3,INPUT);
    pinMode(T4,OUTPUT);
    pinMode(E4,INPUT);

    pinMode(velmotor,OUTPUT); //define pino do controle de velocidade
    pinMode(hor,OUTPUT);
    pinMode(ant,OUTPUT);

    digitalWrite(hor,LOW); //segurança do motor, desliga as duas chaves
    digitalWrite(ant,LOW);
    analogWrite(velmotor,vel);

    servo1.attach(pinServo); //ângulo inicial do servo
    ang=75;

```

```

servo1.write(ang);
Serial.begin(9600);

estado = 'a';
}
void loop() {
  switch(estado) {

    case 'a': // Procurando a vaga

      digitalWrite(T2,LOW);
      delayMicroseconds(2);
      digitalWrite(T2,HIGH);
      delayMicroseconds(10);
      digitalWrite(T2,LOW);
      dur2 = pulseIn(E2,HIGH);
      dis2 = dur2 / 58.2;

      if (dis2<10){

        vel = 200;
        analogWrite(velmotor,vel);
        digitalWrite(hor,HIGH);
        digitalWrite(ant,LOW);
        delay(10);

      }else{

        vel = 0;
        analogWrite(velmotor,vel);
        digitalWrite(hor,LOW);
        digitalWrite(ant,LOW);
        delay(1000);

        estado = 'b';}
      break;

    case 'b': // medindo a vaga
      digitalWrite(T3,LOW);
      delayMicroseconds(2);
      digitalWrite(T3,HIGH);
      delayMicroseconds(10);
      digitalWrite(T3,LOW);
      dur3 = pulseIn(E3,HIGH);
      dis3 = dur3 / 58.2;

      if (dis3<10){

        vel = 200;
        analogWrite(velmotor,vel);
        digitalWrite(hor,HIGH);
        digitalWrite(ant,LOW);
        delay(10);

```

```

}else{

vel = 0;
analogWrite(velmotor, vel);
digitalWrite(hor, LOW);
digitalWrite(ant, LOW);
delay(1000);

estado = 'c';}
break;

case 'c': // condição do tamanho da vaga
digitalWrite(T2, LOW);
delayMicroseconds(2);
digitalWrite(T2, HIGH);
delayMicroseconds(10);
digitalWrite(T2, LOW);
dur2 = pulseIn(E2, HIGH);
dis2 = dur2 / 58.2;

digitalWrite(T3, LOW);
delayMicroseconds(2);
digitalWrite(T3, HIGH);
delayMicroseconds(10);
digitalWrite(T3, LOW);
dur3 = pulseIn(E3, HIGH);
dis3 = dur3 / 58.2;

if (dis2>10 && dis3>10){
estado = 'd';

}else{

estado = 'g';}

break;

case 'd':
digitalWrite(T3, LOW);
delayMicroseconds(2);
digitalWrite(T3, HIGH);
delayMicroseconds(10);
digitalWrite(T3, LOW);
dur3 = pulseIn(E3, HIGH);
dis3 = dur3 / 58.2;

if (dis3>10){

vel = 120;
analogWrite(velmotor, vel);
digitalWrite(hor, HIGH);
digitalWrite(ant, LOW);
delay(10);

```

```
    }else{

    vel = 0;
    analogWrite(velmotor,vel);
    digitalWrite(hor,LOW);
    digitalWrite(ant,LOW);
    delay(1000);

    estado = 'e';}
    break;

    case 'e':

    delay(1000);

    ang=110; //direção para direita
    servol.write(ang);

    delay(1000);

    vel = 190;
    analogWrite(velmotor,vel);
    digitalWrite(hor,LOW);
    digitalWrite(ant,HIGH);
    delay(500);

    vel = 0;
    analogWrite(velmotor,vel);
    digitalWrite(hor,LOW);
    digitalWrite(ant,LOW);
    delay(500);
    estado = 'f';
    break;

    case 'f':
    delay(1000);

    ang=75; //direção em posição inicial
    servol.write(ang);

    delay(1000);

    vel = 190;
    analogWrite(velmotor,vel);
    digitalWrite(hor,LOW);
    digitalWrite(ant,HIGH);
    delay(375);

    vel = 0;
    analogWrite(velmotor,vel);
    digitalWrite(hor,LOW);
    digitalWrite(ant,LOW);
    delay(500);
```

```
    estado = 'h';
    break;

    case 'h':

        delay(1000);

        ang=45; //direção para esquerda
        serv01.write(ang);

        delay(1000);

        vel = 190;
        analogWrite(velmotor, vel);
        digitalWrite(hor, LOW);
        digitalWrite(ant, HIGH);
        delay(400);

        vel = 0;
        analogWrite(velmotor, vel);
        digitalWrite(hor, LOW);
        digitalWrite(ant, LOW);
        delay(500);

        estado = 'i';
        break;

        case 'i':

            delay(1000);

            ang=110;
            serv01.write(ang);

            delay(1000);

            vel = 190;
            analogWrite(velmotor, vel);
            digitalWrite(hor, HIGH);
            digitalWrite(ant, LOW);
            delay(300);

            vel = 0;
            analogWrite(velmotor, vel);
            digitalWrite(hor, LOW);
            digitalWrite(ant, LOW);
            delay(1000);

            estado = 'g';
            break;

            case 'g':

                ang=75;
                serv01.write(ang);

                delay(1000);

                vel = 0;
                analogWrite(velmotor, vel);
                digitalWrite(hor, LOW);
                digitalWrite(ant, LOW);
                delay(1000);
                break;
        }
    }
```