

SÉRGIO LEONIDAS FERNANDES PINHEIRO FILHO

UMA ABORDAGEM SIMPLIFICADA DE AUTOMÓVEIS AUTÔNOMOS

MANAUS – AMAZONAS

MAIO, 2022

SÉRGIO LEONIDAS FERNANDES PINHEIRO FILHO

UMA ABORDAGEM SIMPLIFICADA DE AUTOMÁVEIS AUTÔNOMOS

Monografia submetida à banca avaliadora do Curso de Engenharia de Controle e Automação, como pré-requisito para a obtenção do título de Engenheiro em Controle e Automação pela Universidade do Estado do Amazonas, Escola Superior de Tecnologia.
Orientador: Marivan Silva Gomes

MANAUS – AMAZONAS

MAIO, 2022

SÉRGIO LEONIDAS FERNANDES PINHEIRO FILHO

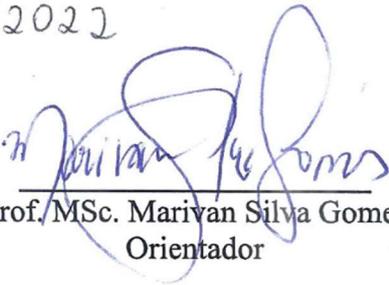
UMA ABORDAGEM SIMPLIFICADA DE AUTOMÁVEIS AUTÔNOMOS

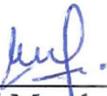
Monografia submetida à banca avaliadora do Curso de Engenharia de Controle e Automação, como pré-requisito para a obtenção do título de Engenheiro em Controle e Automação pela Universidade do Estado do Amazonas, Escola Superior de Tecnologia.

Orientador: Marivan Silva Gomes

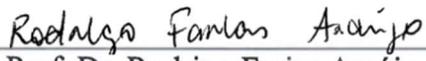
Aprovado em: 01/06/2022

Banca Examinadora:


Prof. MSc. Marivan Silva Gomes
Orientador


Prof. Dr. Israel Morales Mazaira
Coordenador do Curso


Prof. Dr. Jose Ruben Sicchar Vilchez
Membro


Prof. Dr. Rodrigo Farias Araújo
Membro

MANAUS – AMAZONAS

MAIO, 2022

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

P654ua Pinheiro Filho, Sérgio Leonidas Fernandes
Uma abordagem simplificada em carros autônomos /
Sérgio Leonidas Fernandes Pinheiro Filho. Manaus :
[s.n], 2022.
49 f.: color.; 29 cm.

TCC - Graduação em Engenharia de Controle e
Automação; - Universidade do Estado do Amazonas,
Manaus, 2022.

Inclui bibliografia
Orientador: Marivan Silva Gomes

1. Carros Autônomos. 2. MQTT. 3. Seguidor de
Linha. 4. Comunicação Veicular. I. Marivan Silva Gomes
(Orient.). II. Universidade do Estado do Amazonas. III.
Uma abordagem simplificada em carros autônomos

Elaborado por Jeane Macelino Galves - CRB-11/463

Agradecimentos

Agradeço aos meus familiares e amigos que me deram suporte durante a minha jornada acadêmica, aos meus professores do curso e em especial ao professor Marivan Gomes pela orientação, disponibilização do tempo e espaço para realização do projeto, e ao professor Moises Bastos pelo auxílio quanto a idealização do projeto. Às oportunidades que me foram fornecidas enquanto aluno da Universidade Estadual do Amazonas, nas quais pude crescer pessoal e profissionalmente. À coordenação do curso de Engenharia de controle e automação, que deram suporte para a passagem dos conhecimentos requeridos para se graduar engenheiro, sempre procurando fornecer materiais e recursos necessários.

“Você deve ir atrás das oportunidades que aparecem na sua vida com o que você tiver de melhor para realizá-las.” (Soichiro Honda - 本田宗一郎)

Resumo

Ao verificar os dados de um estudo realizado em dezembro de 2018 presentes no “*Global status report on road safety 2018*”, podemos perceber que há um aumento no número de mortes causadas por acidentes veiculares chegando a uma média próxima de 1,35 milhão de óbitos todos os anos em decorrência de acidentes de trânsito. O documento revela ainda que as lesões causadas pelo trânsito são hoje a principal causa de morte de crianças e jovens entre 5 e 29 anos.

Embora muitas medidas de segurança adotadas pelos governos ao redor do mundo ajudem na redução de acidentes, o maior fator de risco ainda persiste, que é justamente a condução humana. Para contornar esse fator, muito se estuda sobre veículos autodirigidos, onde um veículo pode locomover-se por vias públicas juntamente com o fluxo normal de trânsito sem que seja necessária a intervenção humana.

Neste trabalho iremos explorar as características de um carro autônomo, representado por robôs seguidores de linha, em uma situação propensa a acidentes e estudaremos o comportamento dos veículos diante de uma colisão iminente. Os robôs, utilizando Raspberry Pi 4, irão seguir uma pista em loop infinito munidos de sensores RFID para detectar sua posição e acesso à internet, fazendo uso dos protocolos MQTT e SSH para comunicação com o ambiente.

O objetivo de comprovar a segurança provida por um sistema inteligente de locomoção quando em comunicação com o ambiente, é alcançado ao evidenciar a contínua locomoção dos veículos de forma independente, sem que haja colisões entre os mesmos. Usou-se do protocolo MQTT para enviar os comandos de parada antes do cruzamento e continuação do trajeto após constatada a segurança da via para os elementos da maquete, organizando assim a travessia das vias seguindo o critério de ordem de chegada. Este método provou ser efetivo para uma comunicação de baixa escala, como utilizada no projeto.

Palavras-Chave: Carros autônomos. MQTT. Seguidor de linha. Comunicação veicular.

Abstract

When checking the data from a study carried out in December 2018, present in the “Global status report on road safety 2018”, we can see that there is an increase in the number of deaths caused by vehicular accidents reaching an average close to 1.35 million deaths. every year as a result of traffic accidents. The document also reveals that road traffic injuries are now the leading cause of death for children and young people between the ages of 5 and 29. Although many safety measures adopted by governments around the world help to reduce accidents, the biggest risk factor still persists, which is precisely human driving. To circumvent this factor, much has been studied about self-driving vehicles, where a vehicle can travel on public roads along with the normal flow of traffic without the need for human intervention.

In this work we will explore the characteristics of an autonomous car, represented by line-following robots, in an accident-prone situation and we will study the behavior of vehicles in the face of an imminent collision. The robots, built with raspberry Pi 4, will follow an infinite loop track equipped with RFID sensors to detect their position and access the internet, making use of the MQTT and SSH protocols to communicate with the environment. The objective of proving the safety provided by an intelligent locomotion system when in communication with the environment is achieved by showing the continuous locomotion of vehicles independently, without collisions between them. The MQTT protocol was used to send stop commands before the crossing and continuation of the route after verifying the safety of the road to the elements of the model, thus organizing the crossing of the roads following the criterion of arrival order. This method proved to be effective for low-scale communication, as used in the project.

Key-words: Autonomous cars. MQTT. Line follower. Vehicle communication.

Lista de Ilustrações

Figura 1 – Esquemática do robô seguidor de linha em um escritório	18
Figura 2 - Pinagem GPIO Rapsberry Pi	18
Figura 3 - I.o.T. em veículos	21
Figura 4 - Arquitetura da interação no projeto	25
Figura 5 – Pulso para controle do motor	25
Figura 6 – Reconhecimento em arranjo de sensores infravermelho 1	27
Figura 7 - Reconhecimento em arranjo de sensores infravermelho 2	27
Figura 8 – Correção da trajetória do robô	29
Figura 9 - Malha fechada potência do motor	30
Figura 10 – Cadastro dos códigos de cartão RFID.....	31
Figura 11 – Configuração cliente MQTT.....	31
Figura 12 - Fluxograma da interação entre veículos	33
Figura 13 - Leitura dos sensores Infravermelhos no veículo vermelho	34
Figura 14 – Pista de testes, 1ª versão	35
Figura 15 - Pista de testes, 2ª versão	35
Figura 16 - Leitura dos cartões RFID.....	35
Figura 17 - Tela subscriber MQTT	36
Figura 18 - Tela do Raspberry Pi usado como servidor.....	37
Figura 19 - Gráfico da interação entre veículos	38
Figura 20 - Carregador de Bateria.....	44
Figura 21 - Bateria LIPO.....	44
Figura 22 - Raspberry Pi 4	44
Figura 23 - Sensor Infravermelho	44
Figura 24 - Motor Ponte H	44
Figura 25 - Pneu+Motor DC	44
Figura 26 - Conversor DC/DC	45
Figura 27 - Leitor RFID	45
Figura 28 – “Tags” RFID	45
Figura 29 - Pista de Testes	45
Figura 30 – Esquemático do robô seguidor de linha.....	46
Figura 31 - Contagem manual do teste prático	47

Lista de Tabelas

Tabela 1 - Relação das representações de posição com variável de erro	28
Tabela 2 - Resultados experimento prático	37
Tabela 3 - Materiais Utilizados	44
Tabela 4 - Conexões dos componentes às portas GPIO	46

Lista de Abreviaturas e Siglas

AGV	Automated Guided Vehicle
I.o.T	Internet of Things
IDE	Integrated Development Environment
IEEE	Instituto de Engenheiros Eletrônicos e Eletricistas
IrDA	Infrared Data Associations
ISM	Industrial, Scientific and Medical
LED	Light Emitter Diode
MIT	Massachusetts Institute of Technology
MQTT	Message Queuing Telemetry Transport
PID	Proporcional, Integral e Derivativo
RFID	Radio Frequency Identification
RPM	Rotações Por Minuto
RSSF	Redes de Sensores sem Fio
SSH	Secure Shell
SUV	Sport Utility Vehicle
WI-FI	Wireless Fidelity
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

Sumário

1. Introdução.....	11
1.1 Problemática	12
1.2 Hipótese	12
1.3 Justificativa.....	12
1.4 Objetivos.....	12
1.4.1 Geral	13
1.4.2 Específicos	13
1.5 Organização do trabalho.....	13
2. Referencial Teórico	15
2.1 Veículos autônomos	15
2.2 Robô Seguidor de linha	17
2.3 Raspberry Pi 4	18
2.4 Tecnologia Wireless	19
2.5 Wifi.....	20
2.6 Internet das Coisas.....	21
2.7 Protocolo SSH	21
2.8 Protocolo MQTT	22
2.9 RFID	22
3. Materiais e Métodos	24
3.1 Lista de materiais:.....	24
3.2 Robôs seguidores de Linha.....	24
3.3 Ajuste dos motores	25
3.4 Adicionando os sensores de linha.....	26
3.5 Implementando a lógica dos sensores de linha.....	26
3.6 Controlando a direção (Controle Proporcional – p)	29
3.7 Implementando identificação por RFID	30
3.8 Estabelecendo comunicação por protocolo MQTT	31
3.9 Testes em campo	32
3.9.1 Robô seguidor de Linha	32
3.9.2 Interação dos robôs em ambiente não controlado	32

3.9.3	Rastreamento de posição dos robôs na maquete	32
3.9.4	Controle de tráfego realizado automaticamente.....	33
4.	Resultados e Discussão	34
4.1	Da movimentação	34
4.2	Do trajeto	34
4.3	Da leitura RFID	35
4.4	Da comunicação via MQTT	36
4.5	Da interação entre os veículos	36
5.	Considerações Finais e Trabalhos Futuros.....	39
6.	Referências Bibliográficas	40

1. Introdução

Devido à concentração populacional em torno de grandes centros comerciais, presentes nas regiões metropolitanas, o tráfego diário de veículos tem atingido proporções críticas no cenário brasileiro. A busca por conforto e a facilidade na aquisição, especialmente no quesito financeiro, por meio de crédito e financiamentos, contribui com o aumento de veículos individuais em circulação (Neis, L. F., 2019). Deve-se reiterar, também, que devido as condições atuais do transporte público coletivo, este desagrada a maior parte da população, levando quem possui condições financeiras a optar pelo transporte individual motorizado. Tais mudanças, contudo, favorecem o aparecimento de problemas relacionados à mobilidade.

No Brasil, segundo **Cni & Ibope** (2019), 47% dos habitantes possuem veículo particular e o utilizam como meio de transporte. A alta concentração de veículos nas ruas faz com que acidentes sejam mais prováveis. A condução do veículo por seres humanos traz consigo diversos problemas, entre eles o risco de colisão por falha humana especialmente onde há cruzamento de vias.

É neste cenário que se abre oportunidades para a automatização da direção veicular. Os veículos autônomos são elementos integrantes do conceito de “Mobilidade 4.0” que, segundo **M Azmat** (2015), é descrita como “mobilidade inteligente na quarta revolução industrial”. Mobilidade esta que será focada na completa automatização, alta dependência de inteligência artificial e equipamentos de alta tecnologia a bordo dos veículos.

Antes de adentrar o trânsito das cidades, já se era visto veículos autônomos em indústrias e estabelecimentos que realizam transporte de cargas em ambiente interno, estes veículos, porém, necessitam de elementos auxiliares para funcionarem. O AGV ou Veículo Autônomo Guiado (*Automated Guided Vehicle*), é uma tecnologia mais antiga que são guiados por faixas, guias ou fitas magnéticas (**KAR, Aniket K. et al.**, 2016).

No trabalho, iremos abordar o conceito de carros autônomos transitando em vias com cruzamentos e interagindo com outros veículos. Para isso, faremos uso de AGV's, veículos autônomos guiados, ou seja, veículos que tem uma rota pré-definida, e irão percorrê-la automaticamente, o escolhido para este projeto é o robô seguidor de linha de pequeno porte. O que nos permite, com um baixo custo, simular uma situação de tomada de decisão por parte de um veículo autônomo sem *input* humano.

1.1 Problemática

Veículos autônomos são inseridos no mercado como uma das formas de reduzir acidentes e melhorar o fluxo de trânsito. Entretanto, sistemas de controle autônomo de veículos não contemplam todas as fontes de colisão, incluindo cruzamento de vias.

1.2 Hipótese

Atualmente é possível observar em alguns países, carros sendo movidos sem a intervenção humana em vias de alta velocidade, onde o carro não passa por encruzilhadas e não há a necessidade de entrar em ruas perpendiculares. Segundo **Pearl** (2019), podemos esperar que a tecnologia de carros autônomos sirva também para evitar acidentes em cruzamento de vias, realizando comunicações entre os veículos que seriam impossíveis para humanos realizarem entre si. Portanto, neste trabalho iremos simular uma situação de colisão iminente utilizando maquetes e robôs seguidores de linha, a fim de comprovar que carros autônomos são capazes de evitar tal acidente sem intervenção humana.

1.3 Justificativa

De acordo com **Campbell** et al. (2010), se tratando de carros autônomos, controle de velocidade, obediência às faixas, noção de espaço e regras de preferência, tudo pode ser programado no próprio carro, assim, em um ambiente totalmente automatizado, as taxas de acidentes seriam reduzidas a quase zero e o fluxo de trânsito se tornaria próximo do ideal. As condições necessárias para uma sociedade sem condução humana ainda estão longe de serem cumpridas, por isso, as tecnologias utilizadas em carros autônomos necessitam de constante evolução para que os recursos sejam barateados e a situação ideal esteja cada vez mais próxima. Com o resultado desse trabalho, pretendemos demonstrar que, mesmo com equipamentos de baixo custo, podemos incluir recursos autônomos nos veículos e contribuir pouco a pouco para a transição da direção humana para a automatizada.

1.4 Objetivos

1.4.1 Geral

Controlar fluxo de trânsito de forma a priorizar passagem de acordo com quantidade, tamanho das vias, ordem de chegada ou distância até o ponto de encontro, de uma via, em escala, com encruzilhadas assistidas por comunicação Veículo-Servidor. Utilizaremos de dois robôs seguidores de linha, montados pelo autor, para representar os carros autônomos.

1.4.2 Específicos

- Pesquisar referências bibliográficas referentes ao controle de carros autônomos e comunicação veicular;
- Construir uma pista de teste utilizando MDF para validação dos testes, evitando possíveis danos aos veículos nas fases de testes;
- Construir veículos seguidores de linha com módulos infravermelho, Raspberry PI 4 e RFIID, assim como maquete de vias encruzilhadas para testes;
- Desenvolver código fonte em C/C++ em ambiente Debian GNU/Linux versão 11.0 a comunicação MQTT entres os veículos com o servidor a fim de identificar suas posições e velocidades e realizar o controle do tráfego;
- Projetar circuito eletrônicos contendo todos os dispositivos utilizado nos veículos;
- Utilizar elementos de controle para identificar local de parada do veículo quando identificado o sinal fechado;
- Automatizar movimentação dos veículos na maquete construída observando comportamento e interação dos elementos de controle, afim de que se obtenha uma movimentação contínua e harmoniosa por pelo menos 50 voltas, sem a interação externa;
- Testes de validação do trabalho.

1.5 Organização do trabalho

Este trabalho está dividido em cinco capítulos para seu melhor entendimento, onde:

No Capítulo 1 se dá uma breve introdução à temática abordada por esse trabalho e é apresentada a problemática, a hipótese a ser provada, os objetivos gerais e específicos e a justificativa para o mesmo.

O Capítulo 2 traz uma revisão da bibliografia necessária para o desenvolvimento deste trabalho.

O Capítulo 3 descreve o processo de montagem e desenvolvimento do projeto.

O Capítulo 4 apresenta os resultados obtidos, dos quais se faz uma análise e avaliação da performance dos métodos propostos.

Por último, o Capítulo 5 apresenta as conclusões feitas deste trabalho, assim como propostas para trabalhos futuros.

2. Referencial Teórico

2.1 Veículos autônomos

Embora ideias de carros autônomos datem da década de 20, o primeiro conceito de um carro autônomo remonta a Futurama, uma exposição na Feira Mundial de Nova Iorque de 1939. A General Motors criou a exposição para mostrar sua visão de como o mundo seria em 20 anos, e essa visão incluía um sistema rodoviário automatizado que orientaria os carros autônomos. Embora um mundo repleto de veículos robóticos ainda não seja uma realidade, os carros hoje contêm muitos recursos autônomos, como estacionamento assistido e sistemas de freio (Fotsch, 2001). A presença desses recursos dentro de um veículo, pode ser classificada atualmente em 6 níveis (Koopman, 2017; Pereira, 2018), sendo eles:

Nível 0: Automação presente em elementos do carro que não auxiliam na direção diretamente. Nesta categoria estão modelos equipados com sistemas como frenagem autônoma e monitor de pontos cegos. Exemplos: Hyundai HB20 e Chevrolet Onix.

Nível 1: Automação presente em elementos do carro que auxiliam levemente na direção, porém o veículo ainda é dependente do motorista. Se enquadram nessa categoria os veículos equipados com controlador adaptativo de velocidade de cruzeiro ou assistente de manutenção em faixa. Exemplos: Volkswagen Nivus e o Jeep Compass.

Nível 2: Automação que intercala com o motorista a direção do carro, atualmente os mais vistos como lançamento. Em cenários seguros, o veículo consegue acelerar, frear e se manter dentro de faixas de rodagem. Mas o motorista deve permanecer atento à via todo o tempo para assumir a direção em situações de emergência. Exemplos: SUV Volvo XC40 e Mercedes-Benz Classe E

Nível 3: Presente apenas no modelo Honda Legend, é atualmente o mais avançado oferecido em um carro de produção. O sistema autônomo é capaz de guiar sozinho o veículo em congestionamentos. Mas o condutor deve se manter pronto para assumir o controle do veículo quando solicitado.

Nível 4: Sistema autônomo capaz de assumir todas as funções do condutor. Porém, em situações adversas, como mudanças climáticas, o veículo pode solicitar para o motorista assumir a direção.

Nível 5: Dispensa totalmente os motoristas e os comandos manuais, permitindo dar ordens ao veículo por meio de comando de voz, por exemplo.

Assim como **DE SOUZA PISSARDINI** (2013) retrata em seu trabalho, diversos componentes trabalham em conjunto para garantir que um carro autônomo tenha a percepção e a orientação necessárias a uma direção segura. Câmeras, sensores e radares funcionam como os olhos do veículo, detectando obstáculos, sinais de trânsito, semáforos, pedestres, relevo etc.

Os dados recebidos por esses equipamentos são enviados a uma unidade de controle, uma espécie de cérebro do sistema. Ela é responsável por transformar esses dados em ações, controlando a frenagem, a aceleração e a direção do veículo. Segundo **Devi & Rukmini**, (2017), um veículo possui diversas formas de se comunicar com o exterior ou interior do caso, adotando tecnologias de comunicação como *wi-fi*, *bluetooth* e até *Zigbee*.

Ao mesmo tempo que o carro passa a ter condições de identificar e evitar possíveis acidentes, o motorista é alertado de tudo o que se passa a sua volta através do sistema de monitoramento, como **Wang et al.** (2017) bem elabora, o humano em posse do sistema de monitoramento possui as mais diversas informações, e é capaz de agir, caso julgue necessário.

As medidas de segurança previstas na regulamentação de países cujo uso de carros autônomos está mais avançado, como Estados Unidos e Alemanha, determinam que os carros autônomos devem rodar apenas com algum condutor habilitado (**UZAIR, M.**, 2021). Dessa forma, ele poderá assumir o controle da direção quando houver algum tipo de risco à segurança, como no caso de situações não parametrizadas ou mesmo se ocorrer alguma falha no sistema.

Segundo uma pesquisa do canal comunicativo Canalys, um dos maiores comunicadores sobre tecnologia do mundo, no último quadrimestre de 2020, foram vendidos mais de 3,5 milhões de veículos para passeio com grau de automação nível 2, esse número compõe de cerca de 30% da frota vendida nos Estados Unidos e cerca de 20% no Japão, seguidos de alguns outros países da Europa e China.

Por mais que vários países estejam investindo nesse sistema, a transição entre a direção humana e a autônoma não será feita da noite para o dia. segundo **Jung et al.** (2005):

Não se pode, entretanto, prever se os veículos serão, no futuro, completamente autônomos. Diferentes diretrizes, que seguem em direções opostas, devem ser discutidas e estabelecidas pela sociedade. Se, por um lado, a diretriz básica for a segurança total nas estradas, então o fator humano deve ser minimizado e os veículos deverão seguir de forma autônoma seus caminhos. (p. 42).

Ainda segundo **Jung et al.** (2005) existem alguns impasses ligados a essa tecnologia, como a definição da responsabilidade em caso de acidentes, otimização dos sensores para maior percepção das condições de tráfego e do ambiente, adaptação da infraestrutura das cidades (exemplo: faixas de trânsito mal pintadas).

2.2 Robô Seguidor de linha

Seguidor de linha é o nome dado ao robô móvel que possui a capacidade de determinar sua trajetória através de uma faixa posta no chão feita de material de tonalidade contrastante à superfície do solo. O modo mais comum de se determinar a tonalidade do chão, é através de um sensor infravermelho, posicionado próximo ao chão.

Cada sensor é constituído por um LED e um fotodiodo, ambos infravermelhos. A luz emitida pelo LED atinge a superfície e é refletida de volta para o fotodiodo, que em seguida, gera uma tensão de saída proporcional à reflectância da superfície.

Considerando que a trajetória terá, em algum momento, uma curva, ou até mesmo o próprio veículo desviar a trajetória por si só, utilizar um conjunto de sensores se torna mais efetivo quanto a manter o veículo seguindo o trajeto desejado.

Em um caso de pista em forma circular, ou formatos onde o fim encontra o início, um robô seguidor de linha pode continuar a mover-se automaticamente pela sua trajetória por um período de tempo determinado apenas pela duração de sua fonte de energia, se aproximando assim de um veículo autônomo.

Os robôs seguidores de linhas são utilizados de várias formas, desde uso pedagógico, auxiliando no ensino da robótica em todas as faixas etárias, até uso empresarial de larga escala e automação.

Um dos exemplos de sua utilização é explicado por **Kader et al.** (2018), ilustrado na Figura 1, em seu trabalho voltado para utilização do robô no envio de ofícios, lanches e objetos diversos dentro de um escritório. Na dissertação, o robô pode ser chamado por um botão acoplado na mesa de cada funcionário, ele então traça sua trajetória seguindo as linhas marcadas no chão e completa a tarefa requisitada automaticamente.

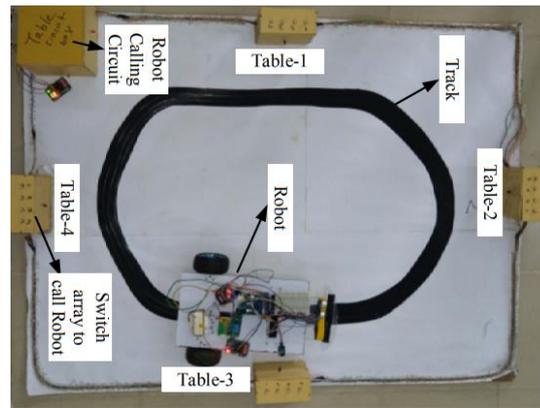


Figura 1 – Esquemática do robô seguidor de linha em um escritório

Fonte: KADER, 2018

2.3 Raspberry Pi 4

A Fundação Raspberry Pi, no Reino Unido, desenvolveu uma série de pequenos computadores a um baixo custo onde todo o hardware se encontra em uma única placa, chamados Raspberry Pi, assim como um desktop, eles possuem saídas para periféricos padrão, como monitores, teclados e mouses. O objetivo da fundação era de promover acessibilidade à informática e tecnologia às pessoas de baixa renda (**Raspberry Foundation**, 2022).

Uma das características do Raspberry Pi é a presença de uma fileira com 40 pinos GPIO (*general-purpose input/output*), como vemos na figura 2. Como o nome sugere, os pinos podem ser usados para *Input* (entrada) ou *Output* (saída), permitindo assim uma gama de opções para utilização do recurso.

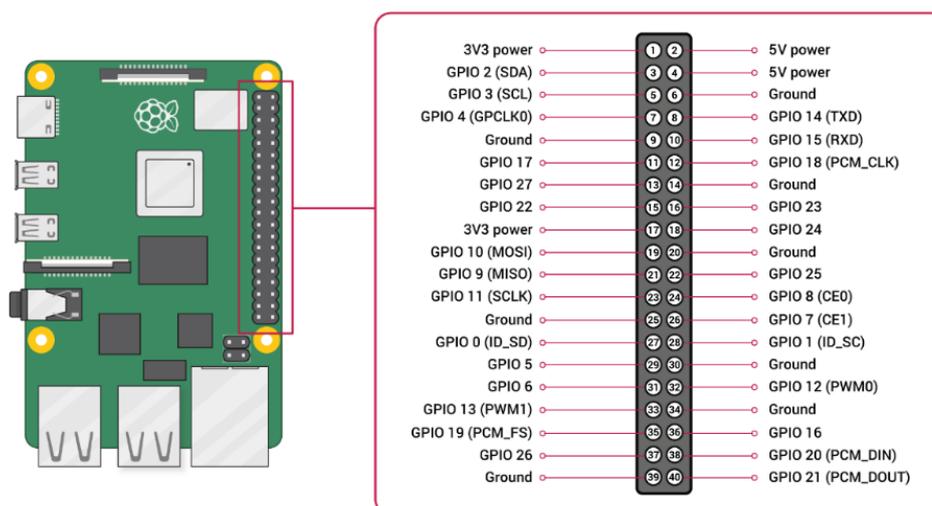


Figura 2 - Pinagem GPIO Raspberry Pi

Fonte: <https://docs.microsoft.com/pt-br/dotnet/iot/tutorials/blink-led>. Acesso em Jan/2021

O Raspberry Pi contém uma vasta coleção de conexões que podem ser utilizadas para as mais diversas atividades, tais conexões incluem: portas USB, Slots de Cartão SD, Porta Ethernet, saída de Áudio, Porta HDMI, Conector de fonte Micro USB e portas GPIO.

2.4 Tecnologia Wireless

Dispositivos comumente usados para redes sem fio incluem computadores portáteis, computadores *desktop*, telefones celulares, entre outros. As redes sem fio funcionam de forma semelhante para redes com fio, no entanto, as redes sem fio devem converter os sinais de informação em uma forma adequada para transmissão através do meio aéreo (**HARDER, Patrick**, 2011).

As redes sem fio atendem a muitos propósitos. Em alguns casos, elas são usadas em substituição a rede de cabos ou quando a infraestrutura não permite a instalação deles, enquanto em outros casos elas são usadas para fornecer acesso a dados corporativos de locais remotos. A infraestrutura sem fio pode ser construída por um custo muito baixo em comparação com a rede com fio (**DAVIS, Harold**, 2004).

As redes sem fio proporcionam as pessoas acesso mais barato as informações. A hora e esforço economizado por ter acesso à rede global de informações traduz em riqueza em escala local, pois mais trabalho pode ser feito em menos tempo e com menos esforço. Redes sem fio permitem que dispositivos remotos se conectem sem dificuldade, independentemente, esses dispositivos estão a poucos metros ou vários quilômetros de distância. Isso fez o uso desta tecnologia muito popular, se espalhando rapidamente (**GAST, Matthew**, 2005). Existem muitas tecnologias que diferem entre si na frequência de transmissão usadas, velocidade e alcance de suas transmissões. Por outro lado, existem algumas questões relacionadas com a regulamentação de emissões dentro do espectro eletromagnético.

As ondas eletromagnéticas são transmitidas através de diversos dispositivos, mas eventualmente, estes dispositivos estão sujeitos a interferências durante a transmissão / recepção. Por esse motivo, todos os países precisam de regulamentações que definam as faixas de frequência e potência de transmissão permitida para cada tecnologia. As redes de sensores sem fio (RSSF) estão em crescente desenvolvimento e seu uso vêm aumentando em diversos setores da sociedade devido à capacidade de monitorar e controlar ambientes sem necessidade de uma ligação física entre os seus dispositivos (**LOUREIRO, Antonio AF et al**, 2003). A comunicação sem fios (ou wireless) é utilizada em redes como WPANs

(*Wireless Personal Area Network*), WLANs (*Wireless Local Area Network*), WMANs (*Wireless Metropolitan Area Network*) e WWANs (*Wireless Wide Area Network*), onde cada classificação varia de acordo com a área de aplicação e o alcance do sinal (SALAZAR, 2017).

A WWAN consiste em redes sem fio capazes de abranger uma grande área geográfica, a partir de antenas ou satélites. A *Wireless Metropolitan Area Network* (WMAN) permite conectar diferentes localizações numa área metropolitana (por exemplo, edifícios de escritórios numa cidade), utilizando fibra ótica ou cabos de cobre e linhas dedicadas. A *Wireless Local Area Network* (WLAN) abrange apenas uma área limitada, equivalente à um cômodo ou escritórios (PARK, Byeong-Yong; JEONG, Myung-Hun; PARK, Seong-Ook, 2014).

2.5 Wifi

A tecnologia Wifi permite a conexão entre diferentes pontos sem a necessidade de fios, através de aparelhos que utilizam radiofrequência ou luz como infravermelho. Dentre as principais tecnologias wireless temos: (ENGST, Adam; FLEISHMAN, Glenn, 2003):

IrDA – (*Infrared Data Associations*) - Barramento de transmissão através de luz infravermelha. Através de sinais de luz emitidos de um LED (diodo emissor de luz) infravermelho, é feita a transmissão de dados. Esses sinais são capturados por um sensor presente no aparelho, receptor (WILLIAMS, Stuart; MILLAR, Iain. *The IrDA Platform*, 1999). Apesar de ser uma interface barata, sua velocidade de transmissão de dados é baixa, alcançando até 115 kpbs. A mobilidade dela também é limitada, visto que tem um alcance não superior a apenas 2 metros e necessita de linha de visada, sem obstáculos, ou seja, os aparelhos precisam estar apontados um para o outro para que haja a transmissão de dados. Esse sistema é comumente utilizado nos controles remotos das televisões (JANUSZKIEWICZ, Łukasz et al, 2016):

IrDA-D – Orientada a conexão entre dispositivos para transferência de dados;

IrDA-C – Orientada a comando e controle.

Wifi – (*Wireless Fidelity*) – Licenciada pelo Wi-Fi Alliance para descrever a tecnologia de redes sem fios embarcadas (WLAN) baseada no padrão IEEE 802.11. O WiFi, baseado no padrão IEEE 802.11, opera nas faixas ISM (*Industrial, Scientific and Medical*) com os seguintes intervalos: 902 MHz - 928 MHz; 2,4 GHz - 2,485 GHz e 5,15 GHz - 5,825 GHz (dependendo do país, esses limites podem sofrer variações) (MATOS, 2017). O

ponto de acesso transmite o sinal sem fios numa pequena distância, geralmente de até 100 metros, mas se a rede for do padrão IEEE 802.11n a distância pode chegar até 400 metros.

2.6 Internet das Coisas

Internet das coisas ou I.o.T. (Sigla em inglês) é a interação de vários objetos conectados à mesma rede, a possibilidade de controlar elementos físicos diversos à distância e permitir que esses elementos controlem outros que estejam a eles conectados (FACCIONE FILHO, 2016). O conceito foi criado no final dos anos 90 no MIT, e vem se adaptando conforme o avanço da tecnologia.

O conceito de I.o.T. pode aplicar-se inclusive para comunicação entre veículos, segundo **Oraibi et al. (2017)**, a comunicação veículo a veículo ainda apresenta divergências quanto ao ideal desejado, fazendo-se a desejável um servidor como mediador da comunicação, representado na figura 3, porém os testes são promissores e irão avançar cada dia mais.

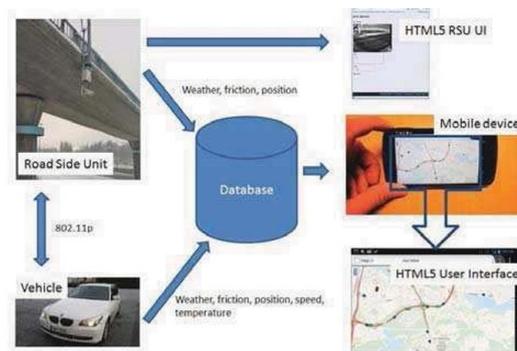


Figura 3 - I.o.T. em veículos

Fonte: Oraibi et al., 2017

2.7 Protocolo SSH

O protocolo SSH (*Secure Shell* ou *Secure Socket Shell*) foi criado para fornecer mais segurança que outros protocolos existentes quando os usuários precisarem controlar e modificar seus servidores remotos na internet. Sendo assim um protocolo que oferece aos usuários, principalmente aos administradores de sistema, uma maneira segura de acessar um computador em uma rede aberta, como a internet (BARRET, 2001).

Utilizando o modelo cliente-servidor, este protocolo conecta um aplicativo cliente *Secure Shell*, que é o onde a sessão é exibida, com um servidor SSH que é onde a sessão é

executada, através da criação de túneis seguros para outros protocolos, ou seja, ele utiliza diferentes combinações de tecnologia para assegurar a comunicação entre dois pontos por meio de um túnel (BUTLER, 2000).

Os administradores de rede utilizam o SSH para gerenciar sistemas de forma remota por confiar em sua criptografia, conhecida como o ponto forte do protocolo. Além disso, também é a criptografia que permite que eles se conectem a outro computador em uma rede, executem comandos e movam arquivos de um computador para outro.

2.8 Protocolo MQTT

Ao se pensar numa estrutura para transmissão de mensagens em uma rede de I.o.T, o padrão é utilizar o MQTT, que consiste no envio e recebimento de mensagens curtas em via dupla (*publisher/subscriber*) para todas as conexões.

Segundo Singh et al. (2015), o MQTT é centrado em dados, com isto é mais simples e possui mais leveza, pois suas mensagens são em formato binário e seu cabeçalho tem no máximo dois bits. Ele segue o modelo de publicação/assinatura, que faz com que mensagem chegue no servidor e não fique esperando a resposta, podendo fazer vários envios sem travas, para isso o protocolo implementa um *middleware* denominado de *broker*. O *broker* é responsável por receber, e disparar as mensagens recebidas dos *publishers* para os *subscribers*.

O *publisher* é responsável por se ligar ao *broker* e publicar as mensagens. Já o *subscriber* é responsável por se ligar ao *broker* e receber as mensagens que ele tiver interesse, essas mensagens são separadas por um “tópico”, que consiste em um caminho determinado pelo programador.

O *publisher* envia a mensagem para o *broker* em um determinado tópico. O *broker* é responsável por receber a mensagem do *publisher* e fazer um pré-filtro das mensagens e enviá-las para os *subscribers* que estiverem registrados em um determinado tópico.

2.9 RFID

A denominação RFID (*Radio Frequency Identification* – identificação por radiofrequência em tradução livre) é utilizada para tecnologias de captação de dados via

rádio frequência. Uma das tecnologias mais utilizadas é a de inserção de um número de série em microchip para identificar determinada pessoa ou objeto.

Para isso, é feita a gravação do número em um elemento ao qual o microchip está alocado, comumente chamado de etiqueta ou “*tag*”, por um equipamento gravador. E na outra ponta, temos um equipamento leitor para identificar o número existente na “*tag*” e confirmar como um dado válido associando-o à pessoa ou objeto cadastrado.

Na prática, um sistema de RFID é composto, basicamente, de uma antena, um transceptor, que faz a leitura do sinal e transfere a informação para um dispositivo leitor, e também uma etiqueta de rádio frequência, que deverá conter o a informação a ser transmitida. Segundo **Domdouzis** et al. (2007) as aplicações para o RFID são abrangentes justamente por identificar a itens, tornando-os rastreáveis.

3. Materiais e Métodos

Para que o projeto se aproximasse da realidade, à um custo acessível para o orçamento limitado de um estudante, escolheu-se o robô seguidor de linha como substituto para um carro autônomo de nível 2. Os kits de robôs seguidores de linha variam de quatrocentos a mil e quinhentos reais, dependendo de quais componentes serão integrados no robô. Adicionando elementos como leitor RFID e conversor DC/DC para energizar os componentes de forma correta, o preço final de cada veículo utilizado no projeto foi menor que mil reais.

3.1 Lista de materiais:

- Carregador de bateria
- Baterias LIPO
- Raspberry Pi4
- Módulo Seguidor De Linha 5 Canais
- Driver de Motor
- Roda C/ Pneu + Motor
- Conversor DC/DC
- Sensor RFID
- Tag RFID
- Maquete da pista em MDF

Os materiais utilizados e suas respectivas figuras estão presentes no apêndice 1.

3.2 Robôs seguidores de Linha

O robô escolhido para o trabalho a ser apresentado perante a banca é composto por controladores Raspberry Pi 4 modelo B, onde dois sensores no mesmo robô (infravermelho e RFID) atuarão de forma separada, mostrando que pode haver interação de dispositivos independente da configuração de hardware, motor em ponte H para melhor controle de velocidade, cinco sensores infravermelho responsáveis pela identificação do trajeto sugerido, um sensor leitor de RFID para localização do robô na maquete, e uso dos protocolos MQTT e SSH para comunicação entre elementos do projeto.

A interação entre os elementos se dará segundo a arquitetura presente na figura 4. Enquanto a montagem foi realizada segundo o esquemático presente no apêndice 2.

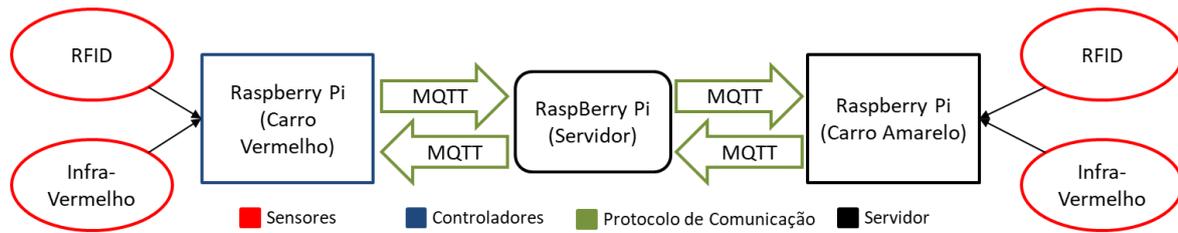


Figura 4 - Arquitetura da interação no projeto

Fonte: Autor

3.3 Ajuste dos motores

Foram utilizados quatro servos em modo contínuo. Esses servos girarão a uma dada velocidade dependendo da largura do pulso que recebam em sua entrada de dados. Para este servo em específico, a largura do pulso varia de 1.0ms a 2.0ms.

Um pulso de 1.5ms comandará o servo para que fique em sua posição neutra, ou “parado”. Um pulso de 1.0ms comandará o servo para que funcione em sua velocidade máxima (cerca de 70 RPM) em uma direção e 2.0ms na direção oposta.

Pulsos entre 1.0 e 1.5ms ou 1.5ms e 2.0ms, farão os servos girarem em velocidades intermediárias e proporcionais.

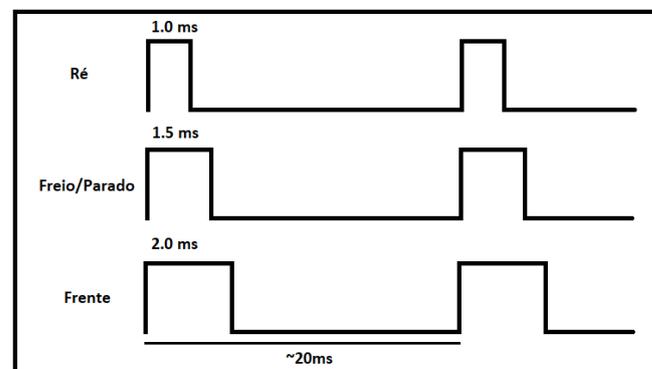


Figura 5 – Pulso para controle do motor

Fonte: Autor

Seguindo o algoritmo abaixo, podemos programar a movimentação das rodas do veículo. No projeto utilizou-se a linguagem C para Raspberry.

Função “Motor Direito Avança”

{

Input positivo no IN1 do motor shield;

Input zero no IN2 do motor shield;

}

Função “Motor Direito Freia”

```
{  
  Input positivo no IN1 do motor shield;  
  Input positivo no IN2 do motor shield;  
}  
Função “Motor esquerdo Avança”  
{  
  Input positivo no IN3 do motor shield;  
  Input zero no IN4 do motor shield;  
}  
Função “Motor Direito Avança”  
{  
  Input positivo no IN3 do motor shield;  
  Input positivo no IN4 do motor shield;  
}
```

3.4 Adicionando os sensores de linha

Ao considerar um maior número de sensores para melhor precisão quanto ao posicionamento do veículo na faixa a ser seguida, optou-se por posicionar cinco sensores para cada veículo, caso fosse utilizado sensores unitários, as ligações com o microprocessador, posicionamento da fiação e acoplagem na miniatura, seria bem mais complexa do que utilizar um módulo completo com o número de sensores desejado, por isso, optamos por utilizar uma placa única que integra cinco sensores infravermelhos.

3.5 Implementando a lógica dos sensores de linha

No caso dos sensores utilizados, um circuito integrado no módulo gera um sinal de saída digital simples (ALTO: escuro; BAIXO: Luz). Para obter um resultado mais aproximado da posição do robô em relação à pista (Fita isolante preta) foi utilizado um módulo integrado com 5 (cinco) sensores, sendo o sensor localizado ao centro, quando em valor “ALTO”, o indicador de posição ideal do carrinho representado na figura 6.

Para melhor ajuste de posição, o espaço entre os sensores permite que dois sensores possam identificar a linha preta simultaneamente, produzindo assim uma entrada dupla de sinal “ALTO”, como representado na figura 7.

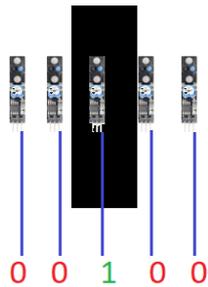


Figura 6 – Reconhecimento em arranjo de sensores infravermelho 1
Fonte: Autor

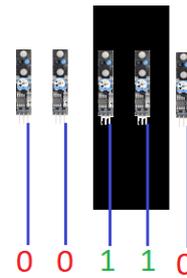


Figura 7 - Reconhecimento em arranjo de sensores infravermelho 2
Fonte: Autor

Trabalhar com cinco sensores, permite a geração de uma “variável de erro” que ajudará a controlar a posição do robô sobre a linha. Adotamos a matriz $[0\ 0\ 1\ 0\ 0]$ como erro zero.

Caso o robô derivar para a esquerda, o erro deverá aumentar e com um sinal positivo. Caso o robô derivar para a direita, da mesma maneira, o erro deverá aumentar, porém com um sinal negativo.

Para identificar cada uma das posições utilizou uma soma das posições de cada sensor em escala logarítmica base 2. Tenso assim um número inteiro relacionado a cada leitura dos sensores infravermelhos como mostrado no algoritmo abaixo.

Função “ler sensor”

{

Variável inteira “Inteiro”

Vetor 5 posições “Vetor”

Se Sensor1 = “ativado” então

Vetor[0] = 1;

Senão

Vetor[0] = 0;

Se Sensor2 = “ativado” então

Vetor[1] = 1;

Senão

Vetor[1] = 0;

Se Sensor3 = “ativado” então

Vetor[2] = 1;

Senão

Vetor[2] = 0;

```

Se Sensor4 = "ativado" então
    Vetor[3] = 1;
Senão
    Vetor[3] = 0;
Se Sensor5 = "ativado" então
    Vetor[4] = 1;
Senão
    Vetor[4] = 0;
Para i de 0 a 4 faça
    Inteiro = Inteiro + (Vetor[i] * (2 ^ i))
}

```

E para que esse valor do erro possa ser implementado no veículo, é necessário formular a equação que o calcule e atribua o erro a ser corrigido em cada servo para que o veículo retorne ao trajeto ideal. Tendo em vista que utilizamos logaritmo de base 2 na função anterior, podemos formular uma equação em que:

```

Se Resto ( $\log_2(\text{Inteiro}) / 1$ ) = 0 então
    erro = ( $\log_2(\text{Inteiro}) - \text{Quociente}(\text{N}^\circ \text{ de sensores} / 2)$ ) * 2;
Senão
    erro = ( $\log_2(\text{Inteiro}) - \text{Quociente}(\text{N}^\circ \text{ de sensores} / 2) - 1,085$ ) * 2;

```

O valor incrementado de "1,085" corresponde à variação do valor logaritmo do número inteiro, seguindo a tabela 1. Portanto, a relação entre o valor lido pelos sensores, a representação em número inteiro e a variável de erro foi definida como:

Tabela 1 - Relação das representações de posição com variável de erro

Representação Binária	Representação Numérica	Log ₂ (Inteiro)	Variável de Erro
0 0 0 0 1	16	4	4
0 0 0 1 1	24	4,585	3
0 0 0 1 0	8	3	2
0 0 1 1 0	12	3,585	1
0 0 1 0 0	4	2	0
0 1 1 0 0	6	2,585	-1
0 1 0 0 0	2	1	-2
1 1 0 0 0	3	1,585	-3
1 0 0 0 0	1	0	-4

3.6 Controlando a direção (Controle Proporcional – p)

Para fazer com que o robô ande na linha, implementou-se uma lógica de controle que funcionará como o “cérebro” dos carrinhos. Suponha que o robô esteja “andando” sobre uma linha e a saída do *Array* de Sensores é: “0 0 1 0 0”. Neste caso, o erro correspondente é “0” e ambos os motores estarão empurrando o robô para a frente com uma velocidade constante. Um código para essa condição seria:

Potência do motor direito = Potencia inicial + erro

Potência do motor esquerdo = Potencia inicial - erro

Chama função “Motor Direito Avança”

Chama função “Motor Esquerdo Avança”

Supondo que o robô comece a derivar para a esquerda, fazendo com que o sensor 3 também, fique sobre a linha. Neste caso, a saída do *Array* de sensores será: “0 0 1 1 0” e o erro = 1. Nesta situação, diminui-se a velocidade do Servo direito, o que significa diminuir a largura do pulso. Além disso, a velocidade do Servo esquerdo deve aumentar, o que significa diminuir a largura do pulso. Para isso, altera-se a função de controle do motor:

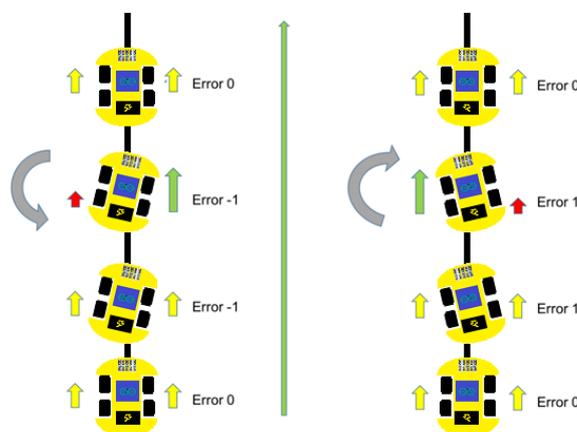


Figura 8 – Correção da trajetória do robô

Fonte: Autor

Sua atuação no código mostrado anteriormente seria:

Potência motor direito = Potencia inicial + erro // erro positivo, potência aumenta)

Potência motor esquerdo = Potencia inicial - erro // erro positivo, potência diminui)

A adição ou subtração de apenas “1” microssegundo na duração dos pulsos não irá gerar a correção necessária, para que a correção se torne eficaz, o número a se adicionar ou

subtrair deve ser maior, por exemplo, 50, 100, etc. Para conseguir isso, o “erro” deve ser multiplicado por uma constante “K”. Uma vez que a influência dessa constante será proporcional ao erro, chamaremos de “Constante proporcional: K_P ”. Inicialmente, com o valor = 50.

A função final para os motores será:

$$K_P = 50;$$

$$\text{Potência do motor direito} = \text{Potencia inicial} + (\text{erro} * K_P)$$

$$\text{Potência do motor esquerdo} = \text{Potencia inicial} - (\text{erro} * K_P)$$

Chama função “Motor Direito Avança”

Chama função “Motor Esquerdo Avança”

Podemos representar o conjunto acima como uma malha fechada segundo a figura 9.

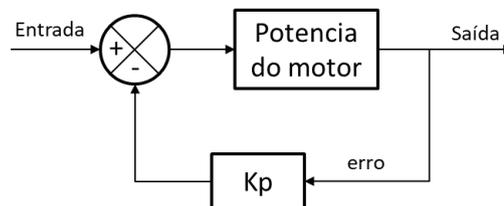


Figura 9 - Malha fechada potência do motor

Fonte: Autor

Neste ponto fica claro que quanto mais o robô “escorregue” para um lado, maior será o erro (1, 2, 3 ou 4) e mais rápido ele deve retornar ao centro (valores a serem adicionados a largura do pulso: 50, 100, 150, 200). A velocidade com que o robô irá reagir ao erro será proporcional ao mesmo, sempre buscando eliminar o erro, ou seja, voltar para a posição ideal “0 0 1 0 0”.

3.7 Implementando identificação por RFID

A fim de sinalizar para os veículos quando há um cruzamento a frente, foram posicionados “tags” de RFID cadastrados com nomes de vias, onde cada uma representa o início ou o final de um cruzamento. Para configurar cada “tag”, é necessário identificar cada número de série, conforme mostra a figura 10.

```
void disp_card_details()
{
    int    tmp;
    char   info[50];
    char   *temp;
    char   *p, sn_str[23];
    char   lido1[] = "[892317B1]";
    char   lido2[] = "[17867B62]";
    char   lido3[] = "[AD555AC4]";
    char   lido4[] = "[4D7591C4]";
    char   lido5[] = "[7DEA59C4]";
    char   lido6[] = "[0D5676C4]";
    char   lido7[] = "[1D8991C4]";
    char   lido8[] = "[4DF88FC4]";
```

Figura 10 – Cadastro dos códigos de cartão RFID

Fonte: Autor

Como explicado por **Vogt** (2002), a leitura das “tags” é feita em ciclos, portanto, para definirmos quando será feita a próxima leitura, atribuímos o valor em milissegundos da última leitura a uma variável, e incrementamos este mesmo valor com o tempo de ciclo desejado, neste caso, utilizamos cem milissegundos. Assim que a leitura de um card cadastrado é feita, enviamos para o veículo, a mensagem via MQTT com a localização do veículo, assim o veículo entende se é necessário parar, ou pode continuar sua rota.

3.8 Estabelecendo comunicação por protocolo MQTT

Para que os veículos pudessem se comunicar, e até mesmo os componentes de um único veículo comunicarem entre si, estabelecemos um tópico junto a um Publisher de MQTT utilizando um raspberry pi externo como host. Inicialmente, configuramos o código com o IP do host e definimos o tópico para a comunicação dos veículos. Também definimos uma “placa” do veículo para identificar para quem a mensagem será enviada, como mostrado na figura 11.

```
#define ADDRESS      "192.168.15.15" //
#define CLIENTID    "Veiculo ABC"
#define TOPIC       "UEA"
#define PAYLOAD     "Carro: XYZ-1234"
#define PLACA       "XYZ-1234"
#define QOS         1
#define TIMEOUT     10000L
```

Figura 11 – Configuração cliente MQTT

Fonte: Autor

3.9 Testes em campo

3.9.1 Robô seguidor de Linha

Como primeiro passo, é feita a montagem e testagem de dois robôs seguidores de linha. Munidos de uma bateria LIPO com 3 células de 11V para autonomia dos veículos, sensores infravermelhos para identificar o caminho a ser percorrido, leitor RFID para localização do veículo na maquete, acesso a rede WI-FI, para utilização do protocolo MQTT, e um driver de motor ponte H para melhor controle de velocidade do veículo. Todos os componentes do veículo são controlados por códigos inseridos no raspberry pi4 acoplado na parte superior do robô.

Os códigos são acionados via SSH a partir de um terceiro raspberry pi que funciona como servidor. Por este servidor, é acessado os arquivos dos dois veículos, e então iniciado os respectivos códigos para leitura de RFID e movimentação. Através do protocolo SSH, pode-se acompanhar o registro das leituras dos sensores, como se estivesse acessando diretamente o terminal no veículo em si.

3.9.2 Interação dos robôs em ambiente não controlado

Para simulação de interação veicular não controlado, dois seguidores de linha são postos em posições distintas na pista de teste em modo de movimentação livre, até que, por uma diferença de velocidade causada pelas curvas da maquete, ocorra uma eventual colisão. Mesmo que os veículos estivessem munidos de sensores a evitar uma colisão frontal, acarretaria em uma mudança de trajetória o que faria o veículo sair da pista, impedindo-o de continuar o trajeto, uma vez que o veículo está programado para pausar o código caso não encontre a faixa da pista. Neste caso, faz-se o paralelo com uma situação onde dois motoristas com lapso de atenção em um cruzamento se envolvem em uma colisão.

3.9.3 Rastreamento de posição dos robôs na maquete

Para um controle efetivo de tráfego, é necessário inicialmente definir uma via preferencial e a localização dos veículos na pista. Para uma forma simplificada, identificamos cada cruzamento com uma “tag” RFID, sendo assim, ao momento em que o robô se alinhar com o cartão, saberemos a posição aproximada do veículo. Com cada robô

trafegando em uma velocidade diferente por conta das curvas na maquete, é necessário definir a preferência seguindo critérios diversos, como via principal, ordem de chegada e veículo preferencial.

Nesse primeiro momento, caso um veículo se localize no início de um cruzamento, o mesmo receberá o comando para parar a movimentação e esperar a resposta positiva para retomada.

3.9.4 Controle de tráfego realizado automaticamente

Para que o experimento tivesse validade, era necessário que vários veículos transitassem em uma pista com risco de colisão. Para isso, utilizamos dois modelos diferentes de robôs e com seus componentes agindo em códigos separados, que representariam a pluralidade de marcas ou tecnologias encontradas em uma situação prevista. O modelo da pista foi idealizado para fornecer um trajeto infinito com dois cruzamentos de vias. Durante o experimento, os dois veículos irão transitar indefinidamente pela maquete, e farão uso da comunicação com um servidor externo, via MQTT para identificar cruzamentos e receber ordens de parada ou autorização de retomada do movimento, sem que haja interferência humana na tomada de decisões seguindo o fluxograma da figura 12.

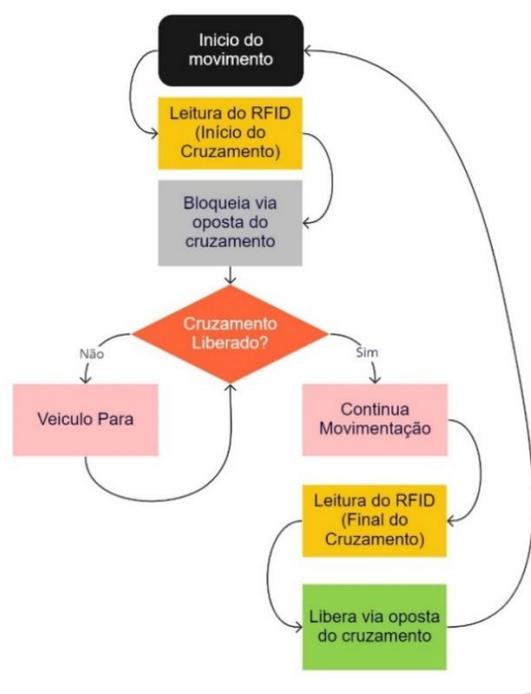


Figura 12 - Fluxograma da interação entre veículos

Fonte: Autor

4. Resultados e Discussão

4.1 Da movimentação

Ao iniciar os testes, percebe-se que a movimentação do veículo é altamente influenciada pela carga da bateria acoplada na miniatura. Por se tratar de um sistema de controle que influencia diretamente na potência do motor, uma bateria sem carga o suficiente, não é capaz de modificar a potência o suficiente para alterar a trajetória do veículo de forma eficiente.

Após alguns ajustes e utilizando as baterias sempre acima de uma carga aceitável, os dois veículos trafegam ininterruptamente pela pista, em um movimento livre sem controle de parada ou evasão de obstáculos. Automaticamente reduzindo a velocidade durante as curvas e atingindo a sua velocidade máxima durante as retas. Durante 50 voltas, foi apontado 43 colisões dos veículos, uma média de 1 colisão a cada 1,16 voltas.

A atualização do sensor é feita a cada 50ms, o que gera uma lista de leitura extensa durante todo o trajeto do veículo, e dá mais precisão quanto ao ajuste da miniatura no trajeto desejado. Abaixo, a figura 13 representa o print de um intervalo de 1 segundo de leitura. Média de 1 colisão a cada 1,16 voltas. Vídeo do experimento da movimentação livre disponível para conferência no link do Youtube: <https://youtu.be/9XSyCHz3tI>.

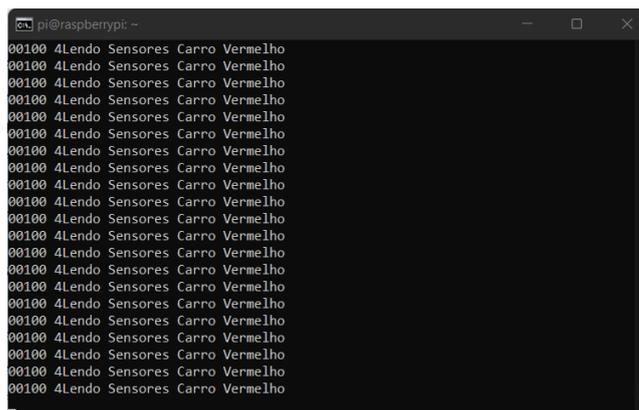
A terminal window titled 'pi@raspberrypi: ~' displays a continuous stream of sensor data. Each line of output consists of a hexadecimal value '00100', the text '4Lendo Sensores', and 'Carro Vermelho'. The text is repeated approximately 25 times, indicating a high-frequency data stream.

Figura 13 - Leitura dos sensores Infravermelhos no veículo vermelho

Fonte: Autor

4.2 Do trajeto

O trajeto idealizado foi pensado para promover dois cruzamentos sem que fosse necessário reposicionar os veículos para reiniciar a volta na pista. Inicialmente o trajeto

proporcionava cruzamentos em ângulo raso, porém, para otimizar a identificação da pista por parte do sensor infravermelho, a pista foi modificada para dois cruzamentos perpendiculares. Assim, permitindo a simulação de uma avenida principal e duas colaterais. Para fim de validação da pista, a ideia de via principal foi temporariamente suspensa, e adotamos todas as vias com a mesma faixa de preferência a mudança seguiu conforme as figuras:

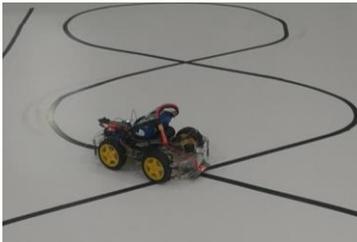


Figura 14 – Pista de testes, 1ª versão
Fonte: Autor

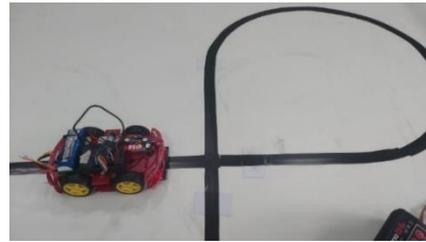


Figura 15 - Pista de testes, 2ª versão
Fonte: Autor

4.3 Da leitura RFID

Cada veículo, ao passar por um cartão RFID, envia uma mensagem para o terminal, confirmando sua localização e enviando uma mensagem ao host do protocolo MQTT, onde identifica a placa do carro e a rua por qual está passando. Caso a localização seja o início de um cruzamento, a mensagem a ser enviada é que, caso o caminho esteja ocupado, o veículo deve parar, e caso seja final de cruzamento, envia a mensagem que é seguro outro veículo atravessar o cruzamento. A figura 16 abaixo mostra as mensagens que um veículo recebe ao atravessar o cruzamento.

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/sergcar# ./car
Aguardando leitura de TAG RFID
Codigo = [4D7591C4]
*****
Rua Cidade Nova
Veiculo Parado
*****
Mensagem Enviada: XYZ-1234 >Rua Cidade Nova
Codigo = [1D8991C4]
*****
Rua Torquato
Veiculo em Seguranca
*****
Mensagem Enviada: XYZ-1234 >Rua Torquato
Codigo = [AD555AC4]
*****
Rua Djalma Batista
Veiculo Parado
*****
Mensagem Enviada: XYZ-1234 >Rua Djalma Batista
Codigo = [4DF88FC4]
*****
Rua Recife
Veiculo em Seguranca
*****
Mensagem Enviada: XYZ-1234 >Rua Recife
```

Figura 16 - Leitura dos cartões RFID
Fonte: Autor

4.4 Da comunicação via MQTT

Para um veículo mandar a mensagem de sua localização, ele necessita de um host, que recebe a mensagem e a envia a todos os que estão conectados ao tópico relacionado. Para fazer o acompanhamento dessas mensagens, utilizamos um broker no terminal do raspberry pi que serve de host e ativamos o protocolo de leitura para todas as mensagens recebidas pelo host.

Quando as mensagens são repassadas, cada veículo identifica, pela placa no início da mensagem, se a mensagem está direcionada a si, ou a outro veículo. Então, o veículo age de acordo com os parâmetros definidos pelo código que se entende por: Caso o caminho esteja livre, segue sem parada e bloqueia o cruzamento da via concorrente; caso o caminho esteja bloqueado, para e espera confirmação de retomada; caso passe pelo final do cruzamento, libera a via concorrente. A figura 17 mostra as mensagens que o veículo envia ao passar por um cartão RFID.

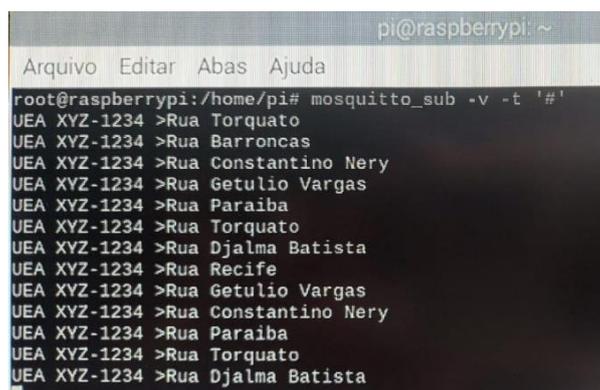
A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows a menu with 'Arquivo', 'Editar', 'Abas', and 'Ajuda'. Below the menu, the command 'root@raspberrypi:/home/pi# mosquitto_sub -v -t '#' is entered. The output consists of a series of messages: 'UEA XYZ-1234 >Rua Torquato', 'UEA XYZ-1234 >Rua Barroncas', 'UEA XYZ-1234 >Rua Constantino Nery', 'UEA XYZ-1234 >Rua Getulio Vargas', 'UEA XYZ-1234 >Rua Paraiba', 'UEA XYZ-1234 >Rua Torquato', 'UEA XYZ-1234 >Rua Djalma Batista', 'UEA XYZ-1234 >Rua Recife', 'UEA XYZ-1234 >Rua Getulio Vargas', 'UEA XYZ-1234 >Rua Constantino Nery', 'UEA XYZ-1234 >Rua Paraiba', 'UEA XYZ-1234 >Rua Torquato', and 'UEA XYZ-1234 >Rua Djalma Batista'.

Figura 17 - Tela subscriber MQTT

Fonte: Autor

Vídeo do experimento de movimentação controlada por MQTT disponível para conferência no link do Youtube: https://youtu.be/ce0kWFfF_b0.

4.5 Da interação entre os veículos

Ao rodar o experimento em modo automático, percebe-se a que a harmonia entre os veículos depende, exclusivamente, de todos os seus componentes estarem em pleno funcionamento. Para que tudo ocorresse dentro do esperado, os ajustes do tempo de leitura dos sensores, velocidade das miniaturas e tamanho da maquete tinham que estar em sintonia.

Ao longo do experimento, houve falha na execução da leitura de alguns componentes, porém, essas falhas foram corrigidas e o experimento seguiu sem mais complicações.

Para que fosse possível observar o comportamento dos elementos, assim como os envios e recebimentos das mensagens via MQTT e, se necessário, realizar ajustes na programação, o reiniciar o código sem ter que retirar os veículos da maquete, fez-se uso do protocolo SSH, dividindo a tela do servidor em 6, permitindo assim o acompanhamento de todo o sistema como mostra a figura 18.



Figura 18 - Tela do Raspberry Pi usado como servidor

Fonte: Autor

Como teste de fadiga, os veículos realizaram o percurso por cerca de 2,5 horas ininterruptas, dentro deste tempo, os veículos completaram 300 voltas na maquete, cada volta durando entre 25 segundos a 35 segundos, aumentando o tempo da volta conforme a bateria era consumida, e tiveram 583 interações, ou seja, se aproximaram um do outro no cruzamento das vias aproximadamente duas vezes por volta. Constatou-se erro de leitura pelos sensores de linha 3 vezes, e pelos sensores de RFID 15 vezes, durante essas falhas, ocorreram duas colisões dos veículos, uma média de 1 colisão a cada 150 voltas. A contagem manual encontra-se no apêndice 3. Devido a diferença de velocidade entre os veículos, fez-se necessário, cerca de oito vezes, o reposicionamento de um dos veículos na maquete, uma vez que não há o conceito de ultrapassagem no projeto.

Tabela 2 - Resultados experimento prático

Voltas dadas	300
Interações entre veículos	583
Erros durante interações	15
Erros no sensor de linha	3
Colisões entre veículos	2
Reposicionamento manual	8

Trecho do vídeo do experimento da movimentação automatizada disponível para conferência no link do Youtube: <https://youtu.be/HuwZUZNcqNw>.

Para demonstrar os resultados práticos, podemos resumi-los em erros e/ou colisões em relação à quantidade de interações durante o tempo de teste, ou seja, a quantidade de vezes que ocorreu um possível acidente, devido a erros de leitura dos sensores. Como mostrado na figura 19, houve um percentual maior de interações bem sucedidas, mostrando a eficiência do sistema para um cruzamento de vias em segurança.

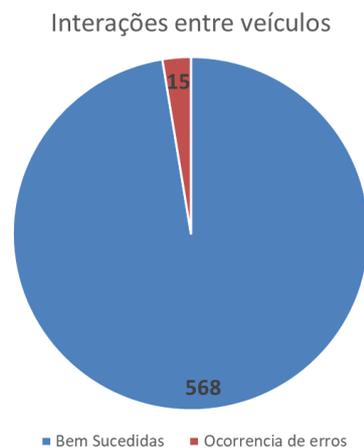


Figura 19 - Gráfico da interação entre veículos

Fonte: Autor

Partindo de uma tensão inicial das baterias entre 12,4V e 12,6V, o primeiro erro de leitura do sensor RFID foi observado após 15 minutos, ao medir a tensão da bateria, obtivemos o valor de 12,1V, a relação do erro com a tensão medida é inconclusiva. Os veículos perderam velocidade e capacidade de locomoção quando a tensão medida na bateria chegou ao valor de 10,4V.

5. Considerações Finais e Trabalhos Futuros

Após o teste, pudemos observar que em um ambiente onde todos os componentes realizaram suas devidas funções, os veículos em circulação transitaram de forma harmoniosa, seguindo e parando conforme a situação demandasse. Sem a necessidade de intervenção humana, a interação entre os veículos se tornou mais rápida e efetiva, provando assim a proposta do projeto, uma vez que o sistema automatizado apontou somente 1 colisão a cada 150 voltas. Também por isso, foi evidenciado que erros de funcionamento dos componentes do veículo podem pôr o sistema em risco. Deve-se salientar que o projeto foi realizado com componentes para fins estudantis e de qualidade que não atende os requisitos de um veículo autônomo comercial.

Ao permitir a fluidez do tráfego enquanto assegura o cumprimento das normas de trânsito pré-estabelecidas no código, fica comprovada a eficiência do sistema de controle de tráfego para veículos autônomos. Podemos afirmar que tal sistema é uma das soluções para reduzir ou até mesmo zerar, momentaneamente, o número de acidentes envolvendo veículos em cruzamento de vias, uma vez que, quando bem programado, o sistema impede os veículos de avançarem sem assegurar que o caminho esteja livre para travessia. Não deixando de lado, um eventual erro elétrico ou mecânico nos componentes do veículo que irá levar o sistema de prevenção à falha.

Ainda há um longo caminho para que um carro com nível de automação 3 ou acima esteja disponível no mercado para o público a preço acessível. Por isso, a otimização de tecnologias utilizadas nos veículos, a fim de baratear os custos ou promover maior eficiência durante o trajeto, é essencial para acelerar o processo de migração da direção humana para a direção autônoma.

O projeto realizado neste trabalho, embora simples, mostra que tecnologias de baixo custo que já são utilizadas comumente hoje, podem ser adaptadas e aproveitadas para situações ainda pouco exploradas envolvendo o trânsito veicular tanto de forma individual como considerando as interações entre os veículos. Como continuação, pretendo abordar a comunicação direta entre veículos, eliminando assim a necessidade de totens, ou servidores centralizados para repasse de informações, assim como conceitos de ultrapassagem e predição de trajeto, trazendo o projeto para níveis maiores da automação veicular.

6. Referências Bibliográficas

ABOUT US. Raspberry Foundation, 2022. Disponível em: <https://www.raspberrypi.com/about/>. Acesso em 03 de Fevereiro de 2022.

AHMAD, Adizul et al. Remote control temperature chamber for virtual laboratory. In: **2017 IEEE 9th International Conference on Engineering Education (ICEED)**. IEEE, 2017. p. 206-211.

ANWAR, Md et al. **Performance Enhancement of Solar Energy using PID controller-based DC-DC boost converter**. 2017. Tese de Doutorado. Department of Electrical and Electronic Engineering, Islamic University of Technology.

ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. Revisiting the Ziegler–Nichols step response method for PID control. **Journal of process control**, v. 14, n. 6, p. 635-650, 2004.

BARRETT, Daniel J. et al. **SSH, the Secure Shell: the definitive guide**. " O'Reilly Media, Inc.", 2001.

BUTLER, Randy et al. A national-scale authentication infrastructure. **Computer**, v. 33, n. 12, p. 60-66, 2000.

CAMPBELL, Mark et al. Autonomous driving in urban environments: approaches, lessons and challenges. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 368, n. 1928, p. 4649-4672, 2010.

CNI & IBOPE. (2019). Retratos da Sociedade Brasileira: segurança pública. **Gerência Executiva de Pesquisa e Competitividade - GPC**, 65. <http://www.cni.org.br/>

DAVIS, Harold. **Absolute beginner's guide to Wi-Fi wireless networking**. Que Publishing, 2004.

DE SOUSA PISSARDINI, Rodrigo; WEI, Daniel Chin Min; DA FONSECA JÚNIOR, Edvaldo Simões. Veículos Autônomos: conceitos, histórico e estado-da-arte. In: **Anais do XXVII Congresso de Pesquisa e Ensino em Transportes–ANPET**. 2013.

DEVI, Y. Usha; RUKMINI, M. S. S. IoT in connected vehicles: Challenges and issues—A review. In: **2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE)**. IEEE, 2016. p. 1864-1867.

- DOMDOUZIS, Konstantinos; KUMAR, Bimal; ANUMBA, Chimay. Radio-Frequency Identification (RFID) applications: A brief introduction. **Advanced Engineering Informatics**, v. 21, n. 4, p. 350-355, 2007.
- ENGST, Adam; FLEISHMAN, Glenn. The Wireless Networking Starter Kit: The Practical Guide to Wi-Fi Networks for Windows and Macintosh p. 55-77, 2003
- FACCIONI FILHO, Mauro. Internet das coisas. **Unisul Virtual**, 2016.
- FONG-CHWEE, Teng; SIRISENA, H. R. Self-tuning PID controllers for dead time processes. *IEEE Transactions on industrial electronics*, v. 35, n. 1, p. 119-125, 1988.
- FOTSCH, Paul Mason. The building of a superhighway future at the New York World's Fair. *Cultural critique*, n. 48, p. 65-97, 2001.
- GAST, Matthew. **802.11 wireless networks: the definitive guide**. " O'Reilly Media, Inc.", 2005.
- GILLARD, George. An introduction and tutorial for PID controllers. **Kättesaadav: [http://smithsrobot. weebly. com/uploads/6/0/9/5/60954939/pid_control_document. pdf](http://smithsrobot.weebly.com/uploads/6/0/9/5/60954939/pid_control_document.pdf) (20.01. 2019)**, 2018.
- GRAHAM C. GOODWIN, S. F. G. M. E. S. *Control System Desig*. Prentice Hall PTR, 2020.
- HARDER, Patrick. A guide to wireless technologies. **ASHRAE Journal**, v. 53, n. 2, p. 44-48, 2011.
- JANTZEN, Jan; JAKOBSEN, Carl. Turning PID controller tuning into a simple consideration of settling time. In: 2016 **European Control Conference (ECC)**. **IEEE**, 2016. p. 370-375.
- JANUSZKIEWICZ, Łukasz et al. Wireless indoor positioning system with inertial sensors and infrared beacons. In: **2016 10th European Conference on Antennas and Propagation (EuCAP)**. **IEEE**, 2016. p. 1-3.
- JUNG, Cláudio Rosito et al. Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. **Anais do CSBC**, v. 5, p. 1358-1406, 2005.
- KADER, Mohammed Abdul et al. Line Following Autonomous Office Assistant Robot with PID Algorithm. In: **2018 International Conference on Innovations in Science, Engineering and Technology (ICISSET)**. **IEEE**, 2018. p. 109-114.

- KAR, Aniket K. et al. Automated guided vehicle navigation with obstacle avoidance in normal and guided environments. In: **2016 11th International Conference on Industrial and Information Systems (ICIIS)**. IEEE, 2016. p. 77-82.
- KOOPMAN, Philip; WAGNER, Michael. Autonomous vehicle safety: An interdisciplinary challenge. **IEEE Intelligent Transportation Systems Magazine**, v. 9, n. 1, p. 90-96, 2017.
- LOUREIRO, Antonio AF et al. Redes de sensores sem fio. In: **Simpósio Brasileiro de Redes de Computadores (SBRC)**. sn, 2003. p. 179-226.
- MATOS, Edemir Marcus Carvalho de et al. Qualidade de vídeo baseada em perda de pacotes utilizando o padrão IEEE 802.11 AC. 2017. MAZARI, Funda Buyuk et al. Novel portable device to analyze the moisture permeability of car seat. **Industria Textila**, v. 69, n. 3, p. 183-189, 2018.
- NEIS, Luan Felipe et al. Mobilidade Urbana e sustentabilidade: em busca de soluções internacionais para os problemas locais. 2019.
- ORAIBI, Ibrahim; OTERO, Carlos E.; OLASUPO, Tajudeen O. Empirical path loss model for vehicle-to-vehicle IoT device communication in fleet management. In: **2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)**. IEEE, 2017. p. 1-4.
- PARK, Byeong-Yong; JEONG, Myung-Hun; PARK, Seong-Ook. A magneto-dielectric handset antenna for LTE/WWAN/GPS applications. **IEEE Antennas and Wireless Propagation Letters**, v. 13, p. 1482-1485, 2014.
- PEARL, Tracy Hresko. Compensation at the Crossroads: Autonomous Vehicles and Alternative Victim Compensation Schemes. In: **Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society**. 2019. p. 187-193.
- PEREIRA, Sandor Banyai; BOTELHO, Rober. Design de Interação: fatores humanos e os carros autônomos. **Design e Tecnologia**, v. 8, n. 16, p. 69-86, 2018.
- REIS, J. W. W. R. A. PID Control of Continuous Processes. **Programmable Logic Controllers, Fourth Edition, Prentice Hall PTR**, 2020.

RYAN, Mark. “Tesla Autopilot 2.0 Full Self Driving Hardware - Neighborhood Long”. Youtube, 2018. **Disponível em:** <https://www.youtube.com/watch?v=wuhbqcMzOaw>. Acesso em 26/05/2021.

SALAZAR, J. Wireless networks. Czech Technical University of Prague, Faculty of Electrical Engineering, p. 7, 2017.

SHAMSUZZOHA, Mohammad; SKOGESTAD, Sigurd. The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning. **Journal of Process control**, v. 20, n. 10, p. 1220-1234, 2010.

SINGH, Meena et al. Secure mqtt for internet of things (iot). In: **2015 fifth international conference on communication systems and network technologies**. IEEE, 2015. p. 746-751.

TORRES, G. F. Explicando a Teoria PID. ni.com, 2020.

UZAIR, Muhammad. Who Is Liable When a Driverless Car Crashes? **World Electric Vehicle Journal**, v. 12, n. 2, p. 62, 2021.

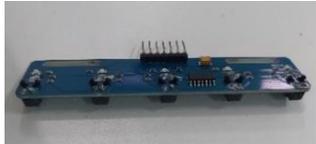
VOGT, Harald. Efficient object identification with passive RFID tags. In: **International Conference on Pervasive Computing**. Springer, Berlin, Heidelberg, 2002. p. 98-113.

WANG, Shulong et al. A novel IoT access architecture for vehicle monitoring system. In: **2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)**. IEEE, 2016. p. 639-642.

WILLIAMS, Stuart; MILLAR, Iain. The IrDA Platform. In: **Insights into mobile multimedia communications**. Academic Press, 1999. p. 593-608.

Apêndice 1 – Lista de Materiais

Tabela 3 - Materiais Utilizados

Item	Quantidade	Descrição	Foto
Carregador de bateria LIPO (Lipo/Life Balance Charger)	1	Carregador para baterias de Lítio	 <p>Figura 20 - Carregador de Bateria Fonte: Autor</p>
Baterias LIPO Turnigy 2200mAh	3	Bateria de lítio de 2200mAh/11.1V	 <p>Figura 21 - Bateria LIPO Fonte: Autor</p>
Raspberry Pi4	3	Computador de placa única utilizada para controlar as miniaturas de veículo	 <p>Figura 22 - Raspberry Pi 4 Fonte: Autor</p>
Módulo Seguidor De Linha 5 Canais Tcrt5000	3	Sensor infravermelho capaz de detectar variação de cor	 <p>Figura 23 - Sensor Infravermelho Fonte: Autor</p>
Driver de Motor: Driver Ponte H Dupla L298n	3	Driver para acionamento de motores em sentido horário ou anti-horário	 <p>Figura 24 - Motor Ponte H Fonte: Autor</p>
Roda C/ Pneu + Motor Dc 3 A 6v C/ Redução Para Arduino	12	Conjunto de roda e motor para movimentação das miniaturas	 <p>Figura 25 - Pneu+Motor DC Fonte: Autor</p>

<p>Conversor DC/DC Buck Boost 2 Em 1 Step Up E Down X16019</p>	<p>3</p>	<p>Regulador de tensão, para alimentação correta do circuito</p>	 <p>Figura 26 - Conversor DC/DC Fonte: Autor</p>
<p>Sensor RF ID: RC522</p>	<p>3</p>	<p>Sensor leitor de código RFID para identificação de cartões “tag”</p>	 <p>Figura 27 - Leitor RFID Fonte: Autor</p>
<p>Cartão “tag” RFID</p>	<p>8</p>	<p>Cartões com criptografia RFID com identificação única</p>	 <p>Figura 28 – “Tags” RFID Fonte: Autor</p>
<p>Placa MDF (1.90x2.80m)</p>	<p>1</p>	<p>Estrutura para simulação e validação do projeto de pesquisa</p>	 <p>Figura 29 - Pista de Testes Fonte: Autor</p>

Apêndice 2 – Conexões dos Equipamentos

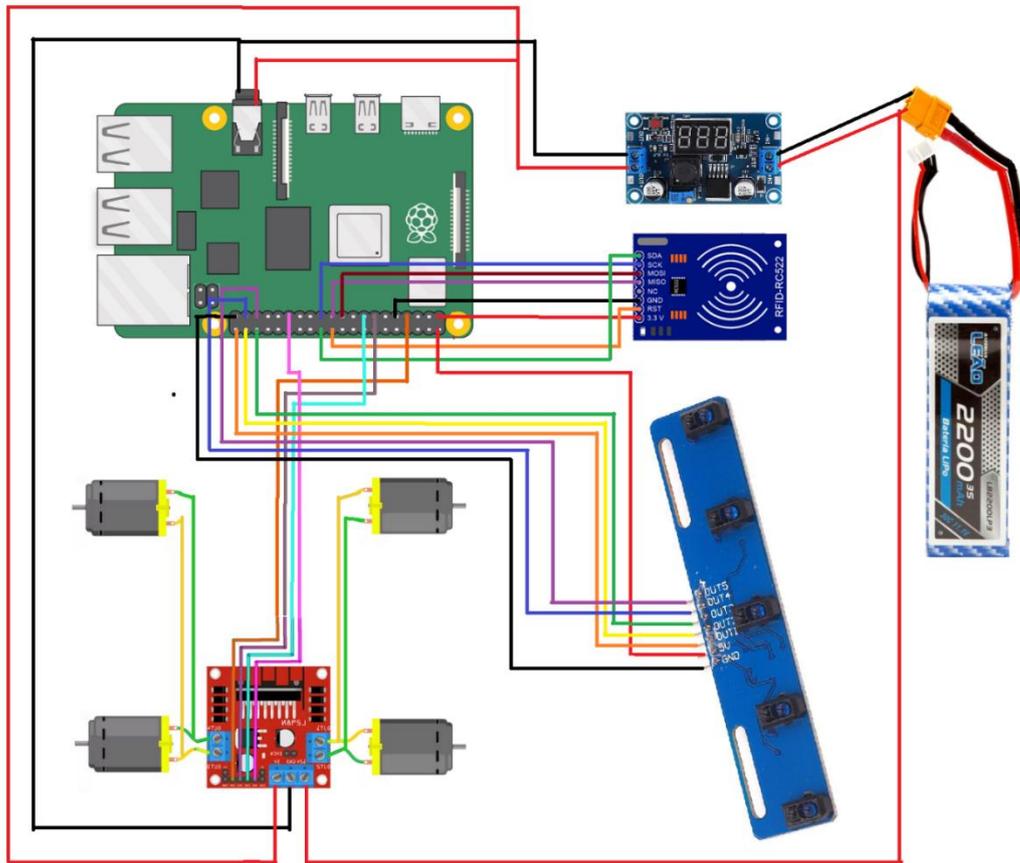


Figura 30 – Esquemático do robô seguidor de linha

Fonte: Autor

Tabela 4 - Conexões dos componentes às portas GPIO

Componente	Pino de Saída	Porta GPIO
Módulo Seguidor De Linha 5 Canais Tcrt5000	Out 1	21
	Out 2	20
	Out 3	16
	Out 4	26
	Out 5	19
Driver de Motor: Driver Ponte H Dupla L298n	In 1	04
	In 2	27
	In 3	22
	In 4	05
Sensor RF ID: RC522	RST	25
	MISO	09
	MOSI	10
	SCK	11
	SDA	08

Apêndice 3 – Contagem de voltas e interações do veículo

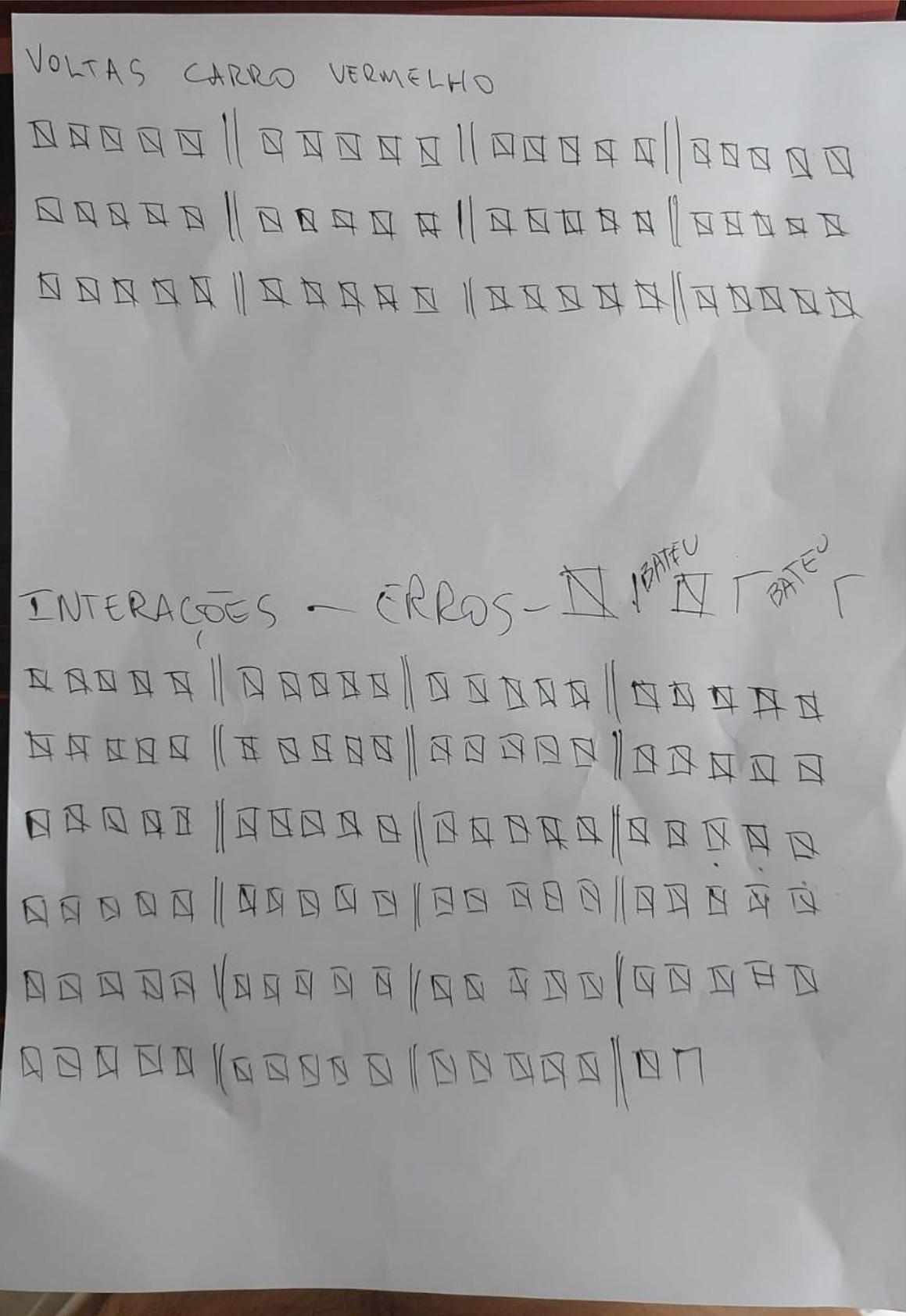


Figura 31 - Contagem manual do teste prático
 Fonte: Autor