



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA

MATEUS PEREIRA LOURENÇO

**DESENVOLVIMENTO DE UM MÓDULO ELETRÔNICO DE TELEMETRIA DE DADOS PARA MOTOCICLETAS
ELÉTRICAS**

Recife

2023

MATEUS PEREIRA LOURENÇO

**DESENVOLVIMENTO DE UM MÓDULO ELETRÔNICO DE TELEMETRIA DE
DADOS PARA MOTOCICLETAS ELÉTRICAS**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Mecânica da Universidade
Federal de Pernambuco, como requisito
parcial para a obtenção do título de
Bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. José Rodrigues de Oliveira Neto.

Recife
2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Lourenço, Mateus Pereira.

Desenvolvimento de um módulo eletrônico de telemetria de dados para motocicletas elétricas / Mateus Pereira Lourenço. - Recife, 2023.

86 p : il., tab.

Orientador(a): José Rodrigues de Oliveira Neto

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Mecânica - Bacharelado, 2023.

1. Telemetria. 2. Datalogger. 3. Motocicletas elétricas. 4. Unidade eletrônica. 5. Projeto. 6. Aquisição de dados. I. Rodrigues de Oliveira Neto, José. (Orientação). II. Título.

620 CDD (22.ed.)



Universidade Federal de Pernambuco
Departamento de Engenharia Mecânica Centro de
Tecnologia e Geociências- CTG/EEP



ATA DE SESSÃO DE DEFESA DE
TRABALHO DE CONCLUSÃO DE CURSO – TCC2

Ao 29.º dia do mês de setembro do ano de dois mil e vinte e três, às 13:00 horas, reuniu-se a banca examinadora composta pelos avaliadores Prof. **José Rodrigues de Oliveira Neto** (orientador), Prof. **João Paulo Cerquinho Cajueiro** (avaliador) e Prof. **Vítor de Andrade Coutinho** (avaliador) para a sessão pública de defesa do Trabalho de Conclusão de Curso em Engenharia Mecânica da Universidade Federal de Pernambuco, intitulado **Desenvolvimento de um módulo eletrônico de telemetria de dados para motocicletas elétricas**, elaborado pelo aluno **Mateus Pereira Lourenço**, matrícula 20160009227. Após a exposição oral do trabalho, o candidato foi arguido pelos componentes da banca que em seguida reuniram-se e deliberaram pela sua _____, atribuindo-lhe a média _____, julgando-o apto() / inapto() à conclusão do curso de Engenharia Mecânica. Para constar, redigi a presente ata aprovada por todos os presentes, que vai assinada pelos membros da banca.

Orientador: Prof. José Rodrigues de Oliveira Neto

Nota:

Assinatura

Avaliador
Interno:

Prof. João Paulo Cerquinho Cajueiro

Nota:

Assinatura

Avaliador
Externo:

Prof. Vítor de Andrade Coutinho

Nota:

Assinatura

Recife, 29 de setembro de 2023.

Prof. Marcus Costa de Araújo
Coordenador de Trabalho de Conclusão de curso - TCC
Curso de Graduação em Engenharia Mecânica – CTG/EEP-UFPE

Esta monografia é dedicada aos meus pais, pilares da minha formação como ser humano.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, pois todo resultado que consegui conquistar foram fundamentados em seus ensinamentos.

Resguardo agradecimentos também a minha companheira que experienciou e me apoiou em toda a minha jornada acadêmica.

Por fim, agradeço à equipe Mangue Baja da UFPE, neste ambiente pude descobrir e exercitar meus interesses pela engenharia de produto.

“O sucesso é a soma de pequenos esforços repetidos dia após dia.” (Robert Collier)

RESUMO

Com a modernização da tecnologia envolvida nos veículos automotores, carros e motocicletas inteligentes estão cada vez mais populares. Seja pela procura de veículos mais seguros ou que sejam capazes de fornecer retroalimentação do perfil de condução para o usuário, a presença dessa tecnologia busca atender a um mercado cada vez mais exigente na percepção de valor. Nesse cenário, este trabalho tem como objetivo o desenvolvimento de uma unidade eletrônica de telemetria para aquisição de dados durante testes em motocicletas elétricas. Para isso, foram definidos os dados de interesse do projeto a serem observados como: torque, velocidade, tensão e corrente de bateria, posição e valores inerciais. Estes dados, por sua vez, estabeleceram as funcionalidades que a unidade eletrônica embarcaria em sua estrutura como: alarme de erros, mapeamento da posição atual, exibição de parâmetros e armazenamento de dados. Essa definição foi seguida do planejamento do *hardware* e funcionalidades de *software* necessários para que fosse possível o completo funcionamento da unidade de forma robusta. Por fim, o projeto foi validado através de testes de rodagem em uma aplicação real de testes urbanos. A utilização do protótipo em um percurso real possibilitou a verificação da boa integração com o veículo em termos de conexão física e comunicação com os demais módulos.

Palavras-chave: Telemetria. Datalogger. Motocicletas elétricas. Unidade eletrônica. Projeto. Aquisição de dados.

ABSTRACT

With the modernization of the technology involved in motor vehicles, smart cars and motorcycles are becoming increasingly popular. Whether it is the demand for safer vehicles or those that are able to provide feedback on the user's driving profile, the presence of this technology seeks to meet an increasingly demanding market in terms of perceived value. In this scenario, this work aims to develop an electronic telemetry unit to acquire data during tests on electric motorcycles. To do this, the data of interest to the project to be observed was defined, such as: torque, speed, battery voltage and current; position and inertial values. This data, in turn, established the functionalities that the electronic unit would incorporate into its structure, such as: error alarms, current position mapping, parameter display and data storage. This definition was followed by the planning of the hardware and software functionalities needed to enable the unit to function fully and robustly. Finally, the project was validated through road tests in a real urban test application. Using the prototype on a real route made it possible to check that it was well integrated with the vehicle in terms of physical connection and communication with the other modules.

Keywords: Telemetry. Datalogger. Electric motorcycle. Electronic unit. Design. Data acquisition.

LISTA DE FIGURAS

Figura 1 –	Transmissão elétrica.....	18
Figura 2 –	Esquema simplificado de um motor DC sem escovas.....	22
Figura 3 –	Esquema de controle do BMS.....	23
Figura 4 –	Projeto em placa perfurada de prototipação.....	24
Figura 5 –	Esquemático genérico conversor Buck.....	25
Figura 6 –	Sinal UART e camada física.....	29
Figura 7 –	Sinal I ² Ce camada física.....	31
Figura 8 –	Sinal SPI e camada física.....	32
Figura 9 –	Sinal CAN e camada física.....	33
Figura 10 –	XL7015.....	42
Figura 11 –	LM2596.....	43
Figura 12 –	MC7800.....	43
Figura 13 –	Esquemático do XL7015	43
Figura 14 –	MPU6050	44
Figura 15 –	MPU9250	45
Figura 16 –	Esquemático do MPU6050	45
Figura 17 –	SJA1000	46
Figura 18 –	MCP2515.....	46
Figura 19 –	Diagrama de blocos do SJA1000.....	46
Figura 20 –	TJA1050	47
Figura 21 –	MCP2551	47
Figura 22 –	Esquemático TJA1050	48
Figura 23 –	SIM7000G	48
Figura 24 –	Diagrama de blocos do SIM7000G	49
Figura 25 –	LILYGO SIM7000G	50
Figura 26 –	Placas perfuradas	50
Figura 27 –	Esquemático da disposição dos periféricos	51
Figura 28 –	Placa de periféricos montada e soldada	52
Figura 29 –	Esquemático da plataforma de validação	53
Figura 30 –	Arduino MEGA	53
Figura 31 –	Diagrama de blocos do MCP2515	54

Figura 32 –	Diagrama de <i>tasks</i> e recursos	58
Figura 33 –	Diagrama conexão entre a central de processamento e os periféricos	63
Figura 34 –	Diagrama montagem do mecanismo de medição. A) acelerômetro B) giroscópio	63
Figura 35 –	Diagrama atividades de interação entre ESP32 e MPU6050	65
Figura 36 –	Diagrama conexão entre o TJA1050 e a ESP32	65
Figura 37 –	Diagrama atividades exercidas pelo SJA1000	66
Figura 38 –	Interface eletrônica SIM7000G	67
Figura 39 –	Diagrama atividades exercidas pelo SIM7000G	68
Figura 40 –	Diagrama atividades exercidas pelo <i>WiFi</i> SOC	68
Figura 41 –	Interface de <i>login</i> e visualização de dispositivos <i>online</i>	71
Figura 42 –	Interface de visualização dos estados da moto em tempo real.....	71
Figura 43 –	Interface de configuração do dispositivo	72
Figura 44 –	Dimensões e montagem	74
Figura 45 –	Comparativo de características mecânicas entre materiais de impressão 3D	75
Figura 46 –	Testes em campo com o veículo	77
Figura 47 –	Testes de consumo energético	77
Figura 48 –	Protótipo não finalizado de uma PCB	78

LISTA DE TABELAS

Tabela 1 –	<i>Tasks</i> e suas descrições divididas por seus <i>handlers</i>	59
Tabela 2 –	Semáforos e suas descrições	61
Tabela 3 –	Dimensões e montagem	72
Tabela 4 –	Cumprimento dos objetivos específicos	76

LISTA DE SIGLAS

CAD	<i>COMPUTED AIDED DESIGN</i>
2D	<i>BIDIMENSIO</i>
3D	<i>TRIDIMENSIONAL</i>
DC	<i>DIRECT CURRENT</i>
AC	<i>ALTERNATING CURRENT</i>
DC-DC	<i>DIRECT CURRENT TO DIRECT CURRENT</i>
IoT	<i>INTERNET OF THINGS</i>
OTA	<i>OVER-THE-AIR</i>
ADC	<i>ANALOG TO DIGITAL CONVERTER</i>
UART	<i>UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER</i>
SPI	<i>SERIAL PERIPHERAL INTERFACE</i>
I ² C	<i>INTER-INTERGRADED CIRCUIT</i>
SDA	<i>SERIAL DATA</i>
SCL	<i>SERIAL CLOCK</i>
MOSI	<i>MASTER OUT SLAVE IN</i>
MISO	<i>MASTER IN SLAVE OUT</i>
SS	<i>SLAVE SELECT</i>
CAN	<i>CONTROLLER AREA NETWORK</i>
MQTT	<i>MESSAGE QUEUING TELEMETRY TRANSPORT</i>
Wi-Fi	<i>WIRELESS FIDELITY</i>
IEEE	<i>INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS</i>
QoS	<i>QUALITY OF SERVICE</i>
AES	<i>ADVANCED ENCRYPTION STANDARD</i>
GSM	<i>GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS</i>
2G	<i>SECOND GENERATION</i>
ETSI	<i>EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE</i>
MS	<i>MOBILE STATION</i>
BMS	<i>BATTERY MANAGEMENT SYSTEM</i>

IDE	<i>INTEGRATED DEVELOPMENT ENVIRONMENT</i>
BS	<i>BASE STATION</i>
MSC	<i>MOBILE SERVICES SWITCHING CENTER</i>
QAM	<i>QUADRATURE AMPLITUDE MODULATION</i>
GPRS	<i>GENERAL PACKET RADIO SERVICE</i>
NB-IoT	<i>NARROWBAND INTERNET OF THINGS</i>
LTE	<i>LONG-TERM EVOLUTION</i>
3GPP	<i>3RD GENERATION PARTNERSHIP PROJECT</i>
CAT-M1	<i>CATEGORY M1</i>
PCB	<i>PRINTED CIRCUIT BOARD</i>
AVR	<i>ALF AND VEGARD'S</i>
RISC	<i>REDUCED INSTRUCTION SET COMPUTING</i>
ECU	<i>ELECTRONIC CONTROL UNIT</i>
MCU	<i>MAIN CONTROL UNIT</i>
SD	<i>SECURE DIGITAL</i>
TF	<i>TRANSFLASH</i>
DMP	<i>DIGITAL MOTION PROCESSOR</i>
ID	<i>IDENTIFIER</i>
AT	<i>ATTENTION</i>
EDA	<i>ELECTRONIC DESIGN AUTOMATION</i>
TCP	<i>TRANSMISSION CONTROL PROTOCOL</i>
SOC	<i>SYSTEM ON CHIPS</i>
BLDC	<i>BRUSHLESS DIRECT CURRENT</i>
PLA	<i>POLYLACTIC ACID</i>
GNSS	<i>GLOBAL NAVIGATION SATELLITE SYSTEM</i>

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVO GERAL E ESPECÍFICOS.....	15
1.2 METODOLOGIA.....	16
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 MECÂNICA ENVOLVIDA.....	17
2.1.1 Motocicletas Elétricas.....	17
2.1.2 Sistema De Orientação Espacial Tridimensional.....	18
2.1.3 Computed Aided Design.....	19
2.1.4 Impressão 3D.....	20
2.2 HARDWARE ENVOLVIDO.....	21
2.2.1 Sistema De Controle De Motores DC Sem Escovas.....	21
2.2.2 Sistema De Gerenciamento De Bateria De Íons De Lítio.....	23
2.2.3 Utilização De Placas Perfuradas Na Prototipação De Projetos Eletrônicos..	24
2.2.4 Conversão Amplitude De Tensão Dc Utilizando Um Conversor Buck.....	25
2.3 FUNCIONALIDADES.....	26
2.3.1 Internet Das Coisas.....	26
2.3.2 Sistema De Atualização Over-the-air.....	27
2.3.3 Data Logging.....	28
2.4 PROTOCOLOS DE COMUNICAÇÃO.....	29
2.4.1 UART.....	29
2.4.2 I ² C.....	30
2.4.3 SPI.....	31
2.4.4 CAN.....	32
2.4.5 MQTT.....	34
2.4.6 IEEE 802.11 B/G/N/E/I.....	35
2.5 REDES DE COMUNICAÇÃO.....	35
2.5.1 GSM.....	36
2.5.2 GPRS.....	36
2.5.3 NB-IOT.....	37
2.5.4 CAT-M1.....	37
2.6 ARQUITETURAS DE PROCESSADORES.....	38
2.6.1 AVR RISC.....	38
2.6.2 Xtensa LX6.....	39
2.7 SISTEMA OPERACIONAL E FRAMEWORKS.....	39
2.7.1 Sistema Operacional Em Tempo Real.....	39
2.7.2 Framework De Desenvolvimento React.js.....	40
3 DESENVOLVIMENTO DO PROJETO.....	40
3.1 DEFINIÇÃO DO FUNCIONAMENTO DO PROJETO.....	41
3.2 SELEÇÃO DOS COMPONENTES.....	42
3.2.1 Seleção Do Conversor Buck XL7015.....	42

3.2.2 Seleção Do Sensor Inercial MPU6050.....	44
3.2.3 Seleção Do Controlador CAN SJA1000.....	45
3.2.4 Seleção Do Transceptor CAN TJA1050.....	47
3.2.5 Seleção Do Modem SIM7000G.....	48
3.2.6 Seleção Da Unidade Principal De Processamento Lilygo ESP32 Wrover-E SIM7000G.....	49
3.4 INTEGRAÇÃO DOS COMPONENTES ELETRÔNICOS.....	50
3.5 PLATAFORMA DE VALIDAÇÃO.....	52
3.5.1 Seleção Dos Componentes.....	53
3.5.2 Desenvolvimento do Firmware.....	55
3.6 FIRMWARE DO MÓDULO IOT.....	55
3.6.1 Seleção Do Repositório E Sistema De Controle De Versão.....	55
3.6.2 Desenvolvimento Do Firmware.....	56
3.6.2.1 Seleção Do Ambiente De Desenvolvimento Integrado (IDE).....	56
3.6.2.2 Seleção Do Framework De Desenvolvimento Do Firmware.....	56
3.6.2.3 Seleção Do RTOS Utilizado.....	57
3.6.2.4 Seleção Da Plataforma De OTA.....	57
3.6.2.5 Armazenamento Geral De Dados De Estados.....	57
3.6.2.6 Configuração Do FreeRTOS.....	57
3.6.2.6.1 <i>Diagrama Das Tarefas</i>	58
3.6.2.6.2 <i>Semáforos E Gestão Dos Recursos Compartilhados</i>	60
3.6.2.7 Integração Dos Componentes No Firmware.....	61
3.6.2.7.1 <i>MPU6050</i>	62
3.6.2.7.2 <i>TJA1050</i>	65
3.6.2.7.3 <i>SIM 7000G</i>	67
3.6.2.7.4 <i>Módulo WIFI SOC (System-On-Chips)</i>	68
3.7 SOFTWARE DE MONITORAMENTO WEB.....	69
3.7.1 Funcionalidades esperadas.....	69
3.7.2 Desenvolvimento do software web.....	70
3.7.3 Design.....	70
3.8 CAIXA PLÁSTICA DE PROTEÇÃO.....	73
3.8.1 <i>Software De Desenvolvimento CAD</i>	73
3.8.2 Compartimento Principal.....	73
3.6.3 Fabricação e material.....	74
4 RESULTADOS.....	75
4.1 TESTES JUNTO À PLATAFORMA DE VALIDAÇÃO.....	75
4.2 TESTES EM CAMPO.....	76
5 PRÓXIMOS PASSOS.....	78
5.1 Placa de Circuito Impresso.....	78
5.2 Ajustes Automatizados do MPU6050.....	79
5.3 Separação do Hardware de GNSS e GSM.....	79
6 CONCLUSÃO.....	79
REFERÊNCIAS.....	81

1 INTRODUÇÃO

A constante busca por aprimorar a eficiência, segurança e sustentabilidade no setor automotivo tem impulsionado avanços tecnológicos notáveis ao longo das últimas décadas (FIERCE et al, 2022). Entre esses avanços, destaca-se a evolução dos sistemas de telemetria veicular, que revolucionaram a forma como veículos são monitorados, controlados e mantidos, como indica o relatório da Fortune sobre o mercado de telemetria. A telemetria veicular engloba um conjunto de tecnologias que possibilitam a coleta, transmissão e análise remota de dados relacionados ao funcionamento de um veículo, proporcionando insights valiosos para motoristas, fabricantes e empresas de serviços automotivos, Infor (2023).

Desde os primeiros sistemas de telemetria veicular desenvolvidos nas décadas de 1960 e 1970, a tecnologia tem experimentado um crescimento exponencial em termos de capacidade, alcance e aplicabilidade. Inicialmente utilizada em competições automobilísticas para monitorar o desempenho dos carros em tempo real, a telemetria logo encontrou seu caminho para os veículos de uso cotidiano, ampliando suas funcionalidades. Hoje, sistemas de telemetria veicular podem coletar e transmitir uma ampla gama de informações, incluindo temperatura do motor, consumo de combustível, condições de tráfego e até mesmo padrões de condução do motorista, como é explicado no manual do sistema de *Safety Score*¹ da montadora de veículos elétricos Tesla.

Ao longo dos anos, a telemetria veicular se beneficiou significativamente dos avanços na conectividade, sensores e análise de dados. A evolução das redes móveis permitiu a transmissão de dados em tempo real com maior velocidade e confiabilidade, enquanto os sensores automotivos mais sofisticados possibilitaram uma coleta mais precisa e abrangente de informações sobre o veículo e seu ambiente. Além disso, algoritmos avançados de análise de dados têm transformado esses fluxos de informações em *insights* acionáveis, auxiliando motoristas na tomada de decisões mais informadas e fornecendo às empresas automotivas informações valiosas para aprimorar o *design* e a funcionalidade dos veículos.

¹ Disponível em> <<https://www.tesla.com/support/safety-score#version-2.0>> Acesso em: 03 set, 2023

Esses benefícios se tornam especialmente valiosos para o cenário de veículos elétricos. A telemetria veicular, ao aproveitar esses avanços, desempenha um papel crucial na otimização do desempenho e na eficiência desses veículos. Essa tecnologia tem o potencial de maximizar a eficiência energética ao retornar ao usuário os dados importantes referentes ao seu modo de condução para que comportamentos ineficazes possam ser tratados. Portanto, o objetivo do projeto explanado neste trabalho é o desenvolvimento de um dispositivo que realiza a interface entre um veículo elétrico e o condutor de forma remota.

1.1 OBJETIVO GERAL E ESPECÍFICOS

O objetivo geral deste trabalho é o desenvolvimento de um módulo eletrônico capaz de receber as informações de diversos sensores presentes em uma motocicleta elétrica. Esse dispositivo deve atender a necessidade de monitorar testes de engenharia e fomentar tomadas de decisão através de *feedbacks* instantâneos além do registro desses para análises mais detalhadas. Para isso, o equipamento deve ser capaz de comunicar com os engenheiros de teste de forma rápida, além de ser capaz de armazenar os dados sendo trabalhados. Tudo isso deve ser feito de forma confiável e eficiente do ponto de vista energético e de consumo de dados. Para isso, os seguintes objetivos específicos deverão ser alcançados:

- A. Desenvolver um método de comunicação eficiente entre o módulo desenvolvido e os diversos sensores utilizados no veículo;
- B. Desenvolver um método de mensurar a orientação do veículo;
- C. Desenvolver um método de comunicação entre o módulo desenvolvido e algum servidor remoto;
- D. Desenvolver um método de armazenamento de dados em tempo real;
- E. Desenvolver um método de comunicação *peer-to-peer*;
- F. Desenvolver um método de atualização remota para o *firmware* do módulo desenvolvido;
- G. Manter baixos níveis de consumo energético e de dados;

1.2 METODOLOGIA

Nesse trabalho exploratório, deseja-se estudar o desenvolvimento de um equipamento eletrônico capaz de monitorar os estados veiculares e agir como interface entre veículo e internet.

Esse estudo conta com uma abordagem prática, em que as decisões de projeto foram tomadas fortemente embasadas em características físicas dos componentes escolhidos. Além disso, as técnicas utilizadas foram selecionadas de forma a se obter o resultado final com o menor desvio possível do foco do desenvolvimento da camada de aplicação. A abordagem conta, ainda, com o desenvolvimento de um protótipo funcional, validando todas as etapas do trabalho explanado neste documento.

Foram utilizadas técnicas de prototipagem seguindo alguns passos referentes ao desenvolvimento de produtos abordados por empresas automotivas. Nesse contexto, pode-se citar as técnicas de *benchmarking*, versionamento de projeto e validação, que garantiram a seleção de bons componentes, em relação à empacotamento e funcionalidades, evolução contínua do resultado final e a segurança na confiabilidade do projeto.

Com foco na entrega de um protótipo funcional, foram tomadas abordagens de natureza qualitativa e quantitativa. Para a primeira, observação do cumprimento de funcionalidades da aplicação tomadas como objetivo foram registradas de forma comparar com o seu respectivo cumprimento. A abordagem qualitativa, por sua vez, foi caracterizada pela observação de grandezas relacionadas ao consumo de potência pelo protótipo utilizando fontes de bancadas.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é descrito os fundamentos que guiaram o desenvolvimento de projetos de sistemas embarcados. Nesse sentido, serão explicados conceitos relacionados a *software* e *hardware* do projeto e do veículo em que este foi desenvolvido para funcionar.

2.1 MECÂNICA ENVOLVIDA

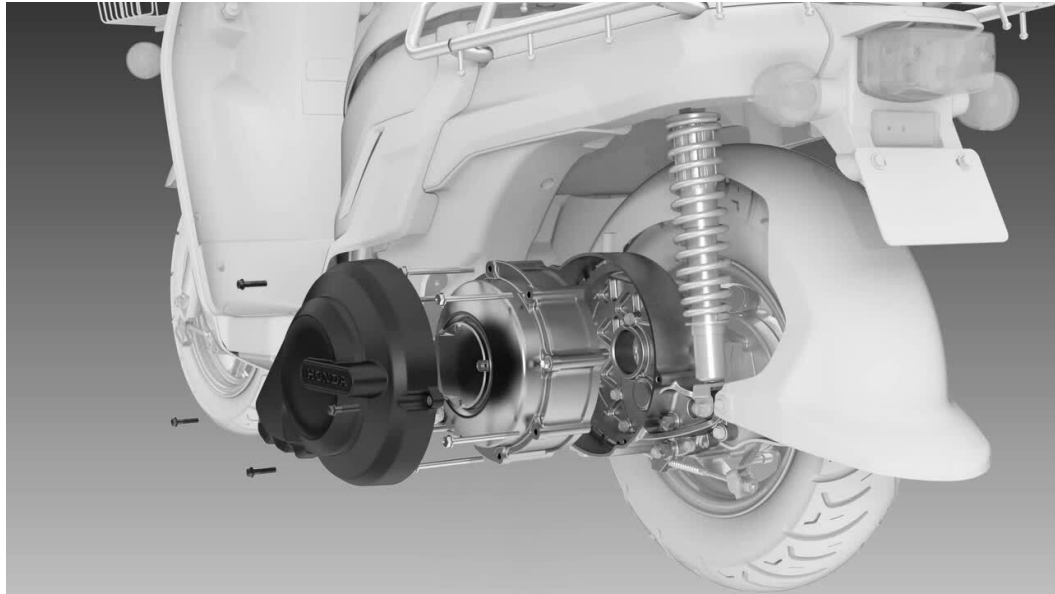
Nesta seção, é explicado os conceitos mecânicos básicos para o desenvolvimento físico do produto estudado. É descrito desde o funcionamento do ambiente de trabalho do dispositivo até as ferramentas utilizadas para o desenvolvimento.

2.1.1 Motocicletas Elétricas

Uma motocicleta elétrica é um veículo que utiliza um motor elétrico para converter a energia armazenada em uma bateria em movimento. O motor elétrico em uma motocicleta elétrica é responsável por transformar a energia elétrica armazenada na bateria em energia mecânica para mover as rodas, (Liu et al, 2020).

De maneira geral, uma motocicleta elétrica consiste em quatro componentes principais: a bateria, o controlador, o motor elétrico e a transmissão. A bateria é a fonte de energia que alimenta o motor elétrico e é responsável por armazenar a energia elétrica (Pereira et al, 2020). Já o controlador é o componente que gerencia a energia da bateria e controla o motor elétrico. O motor elétrico, por sua vez, é o responsável por converter a energia elétrica em energia mecânica, proporcionando a força necessária para mover a motocicleta. A transmissão é responsável por transferir a energia do motor elétrico para as rodas, permitindo que a motocicleta se mova, conforme ilustrado na Figura 1.

Figura 1 – Transmissão elétrica



Fonte: www.shorturl.at/zCTVX

Além disso, uma motocicleta elétrica apresenta algumas vantagens em relação a uma motocicleta convencional. A principal vantagem é a eficiência energética, pois o motor elétrico é capaz de converter uma porcentagem muito maior da energia armazenada na bateria em movimento do que um motor de combustão interna (Rojas et al, 2020). Além disso, a motocicleta elétrica é mais silenciosa, mais limpa e produz menos vibração do que uma motocicleta convencional.

2.1.2 Sistema de Orientação Espacial Tridimensional

Para o controle desse tipo de veículo, o sistema de orientação tridimensional utilizando *pitch*, *yaw* e *roll* é essencial para a navegação, pois permite que a orientação do veículo seja medida e controlada de forma precisa. A medição precisa da orientação é fundamental para a estabilidade e controle do veículo em manobras complexas, como curvas acentuadas e mudanças de direção (Zhang et al, 2018).

O sistema de orientação tridimensional utilizando *pitch*, *yaw* e *roll* é utilizado em navegação e controle de veículos aéreos e submarinos. De acordo com Cheng et al. (2017), este sistema é baseado em três eixos ortogonais que passam pelo centro de massa do veículo: o eixo longitudinal, paralelo à linha formada pelo centro das duas rodas da motocicleta, o eixo lateral, paralelo à linha formada entre os dois espelhos da motocicleta, e o vertical, perpendicular ao plano do chão.

O *pitch*, *yaw* e *roll* representam rotações em torno dos eixos longitudinal, vertical e lateral, respectivamente. O *pitch* é a rotação em torno do eixo longitudinal, que faz com que a extremidade dianteira do veículo se mova para cima ou para baixo (He et al, 2019). O *yaw* é a rotação em torno do eixo vertical, que faz com que o veículo mude a direção em relação ao seu eixo longitudinal. Por fim, o *roll* é a rotação em torno do eixo lateral, que faz com que o veículo incline para um dos lados.

2.1.3 Computed Aided Design

Computed Aided Design (CAD) é uma técnica computacional utilizada em diversos campos da engenharia, arquitetura e design. O CAD é uma tecnologia de software que permite criar, modificar e analisar modelos virtuais em 2D ou 3D de produtos e sistemas (Ahmad Ibrahim, 2021). Dessa forma, essa técnica será utilizada para o desenvolvimento do invólucro plástico que integrará os componentes deste projeto.

O CAD permite aos usuários criar modelos de forma rápida e precisa, permitindo a visualização de projetos antes da construção física. Ele é utilizado em projetos de engenharia para criar desenhos precisos e detalhados de peças mecânicas, sistemas elétricos e eletrônicos, sistemas de tubulação, estruturas e outros componentes (Hossain et al, 2020).

O uso de *CAD* pode trazer benefícios significativos para o processo de design, incluindo redução de erros, economia de tempo e custos, melhoria na qualidade do produto final e a capacidade de simular o desempenho do produto antes da produção física. Essa ferramenta também pode permitir a colaboração entre equipes de projeto distribuídas geograficamente, permitindo que os membros da equipe trabalhem juntos em tempo real em um ambiente virtual.

Além disso, a integração do *CAD* com outras tecnologias de software, como simulação, análise de elementos finitos e manufatura assistida por computador, pode melhorar ainda mais o processo de design e produção. A combinação do *CAD* com simulação permite que os engenheiros simulem e analisem o desempenho de um produto antes da produção física, permitindo a otimização do design (Chowdhury et al, 2020). O uso de *CAD* integrado com manufatura assistida por computador, como impressão 3D, permite que os modelos virtuais sejam transformados em objetos físicos de forma rápida e precisa.

2.1.4 Impressão 3D

A impressão 3D é uma técnica de fabricação aditiva que utiliza uma impressora 3D para criar objetos tridimensionais a partir de um modelo digital. A impressão 3D é uma tecnologia emergente que tem a capacidade de revolucionar a maneira como os produtos são fabricados, permitindo a personalização em massa, a produção sob demanda e a criação de geometrias complexas (Vaezi e Seitz, 2020). Dessa forma, o invólucro desenvolvido utilizando a técnica mencionada anteriormente será fabricado utilizando esta técnica de manufatura aditiva.

A técnica de impressão 3D é realizada em três etapas principais: preparação do modelo, configuração da impressora e impressão do objeto. A preparação do modelo envolve a criação de um modelo digital tridimensional em um software de *CAD* (como descrito anteriormente) ou por meio de uma digitalização tridimensional de um objeto existente (Gao et al, 2015). A configuração da impressora envolve a escolha de materiais (como plástico, metal, cerâmica ou biotintas), a definição da resolução e espessura de camada e a configuração da temperatura e velocidade de impressão. Finalmente, a impressão do objeto envolve a criação do objeto camada por camada, a partir da extrusão do material derretido ou por meio da solidificação de materiais líquidos.

A impressão 3D é utilizada em diversos campos, incluindo engenharia, arquitetura, medicina e arte. A impressão 3D permite a fabricação de próteses personalizadas, implantes cirúrgicos e modelos anatômicos para planejamento cirúrgico, permitindo uma precisão e eficiência na prática médica (Silva et al, 2017). Além disso, a impressão 3D pode ser utilizada na produção de peças de reposição, ferramentas e protótipos de produtos em engenharia, permitindo uma redução nos custos e tempos de produção.

2.2 *HARDWARE* ENVOLVIDO

Neste capítulo, é explanado os conceitos básicos de Hardware elétricos e eletrônicos que guiam o desenvolvimento desta área no trabalho. É explicado conceitos desde o sistema embarcado veicular até os fundamentos de componentes básicos.

2.2.1 Sistema de Controle de Motores *DC* sem Escovas

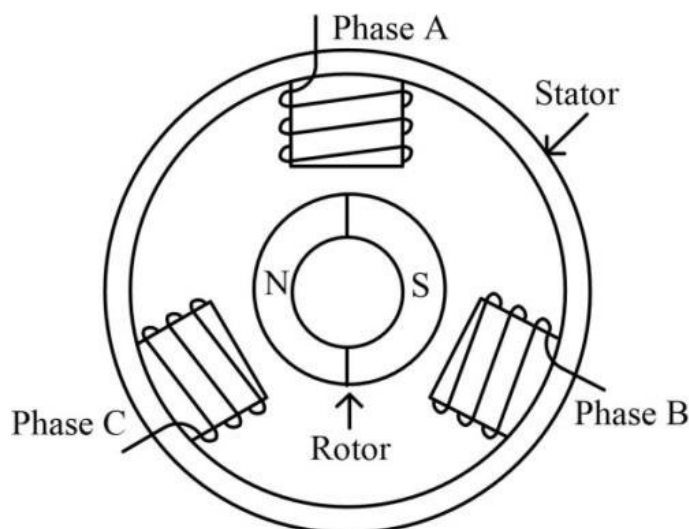
Um sistema de controle de motores de corrente contínua (*DC*) trifásicos sem escovas é um tipo de controle que utiliza um motor *DC* alimentado por um sistema trifásico de corrente alternada (*AC*). Este tipo de sistema é utilizado em aplicações de controle de movimento, como em robótica, sistemas de processos industriais e automação em geral, Automate (2019).

O controle de motores *DC* sem escovas é baseado em um circuito eletrônico que manipula a corrente elétrica que é fornecida ao motor (Karami et al, 2018). Este circuito eletrônico consiste em um inversor, que é responsável por gerenciar a corrente que é fornecida ao motor, e um conjunto de sensores que medem a posição e a velocidade do rotor. Todas essas informações são transmitidas para outros módulos como painel, sistema de gerenciamento de baterias (*BMS*) e módulos de telemetria, como o que é estudado neste trabalho.

Esse controle é realizado através da variação da corrente que é fornecida ao motor. O controle da corrente é realizado através de um conjunto de transistores que são acionados pelo microcontrolador central (Jang et al, 2020). Estes transistores são responsáveis por comutar a corrente que é fornecida ao motor, de modo a produzir um campo magnético rotativo que impulsiona o movimento do rotor do motor.

Um dos principais benefícios do uso de um sistema de controle de motores *DC* sem escovas é a alta eficiência energética, pois o motor, Figura 2, é alimentado por um sistema trifásico de corrente alternada, que é mais eficiente do que um sistema a combustão. Além disso, a falta de escovas no motor elimina o desgaste mecânico associado ao uso de motores *DC* com escovas, (Yang et al, 2020).

Figura 2 – Esquema simplificado de um motor DC sem escovas



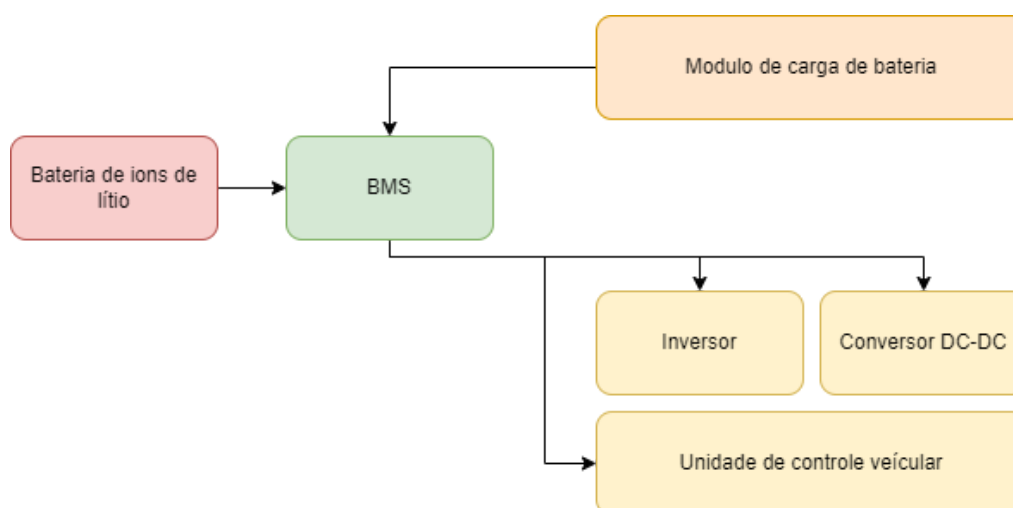
Fonte: www.researchgate.net

2.2.2 Sistema de Gerenciamento de Bateria de Íons de Lítio

Um sistema de gerenciamento de baterias de íons de lítio é um componente essencial em sistemas de armazenamento de energia baseados em baterias, como em veículos elétricos, sistemas de energia solar e aplicações de armazenamento de energia em geral. Este sistema é responsável por monitorar o estado da bateria, controlar o carregamento, a descarga da bateria e proteger contra condições adversas de operação (Ttconsultants, 2022).

Um sistema de gerenciamento de baterias de íons de lítio típico é composto por vários componentes, incluindo sensores de corrente e tensão, um controlador de carga e descarga, um sistema de proteção contra sobrecarga e sobre descarga, e um sistema de comunicação para enviar dados sobre o estado da bateria ao usuário ou sistema de controle (Ouyang, 2017). No cenário estudado neste documento, esse tipo de sistema se apresenta conforme o diagrama da Figura 3.

Figura 3 – Esquema de controle do BMS



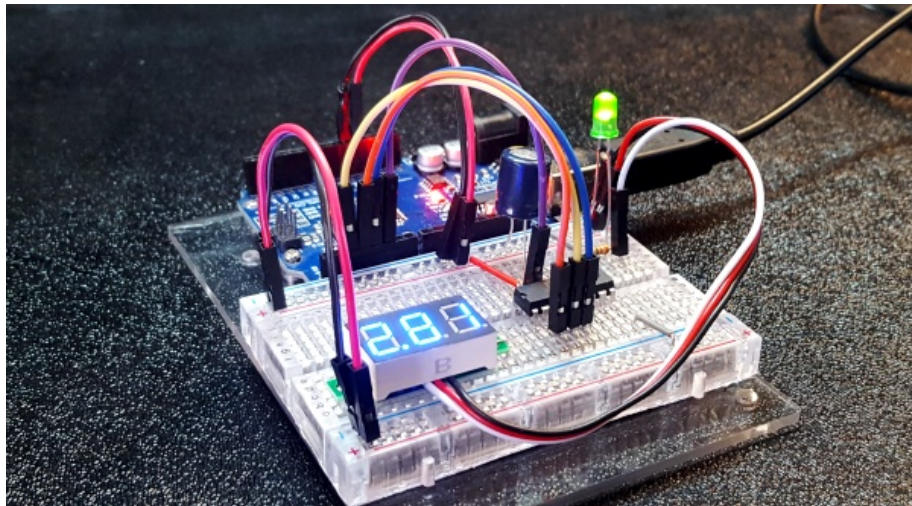
Fonte: O autor, 2023.

Um dos principais desafios no gerenciamento de baterias de íons de lítio é garantir a segurança da bateria durante o carregamento e descarregamento. Isso é realizado através do monitoramento constante da temperatura da bateria, da corrente e da tensão, e do controle preciso da carga e descarga da bateria, (Wang et al, 2018).

2.2.3 Utilização de Placas Perfuradas na Prototipação de Projetos Eletrônicos

As placas perfuradas são frequentemente utilizadas em prototipação de projetos eletrônicos, oferecendo uma plataforma prática e versátil para testes de circuitos. Segundo Zavrel et al (2015), as placas perfuradas, como apresentado na Figura 4, são úteis para prototipagem rápida de circuitos eletrônicos, permitindo que os engenheiros experimentem e modifiquem seus projetos antes de produzirem um circuito finalizado.

Figura 4 – Projeto em placa perfurada de prototipação



Fonte: protosupplies.com

As placas perfuradas consistem em uma placa de material isolante com vários furos perfurados em sua superfície, geralmente dispostos em uma matriz de coordenadas. As placas perfuradas são projetadas para permitir a inserção de componentes eletrônicos através dos furos e para estabelecer conexões entre os componentes através de trilhas condutoras (Sánchez, 2013). Essas trilhas são geralmente criadas por meio da soldagem de fios condutores nos furos adjacentes, criando uma conexão elétrica entre os componentes.

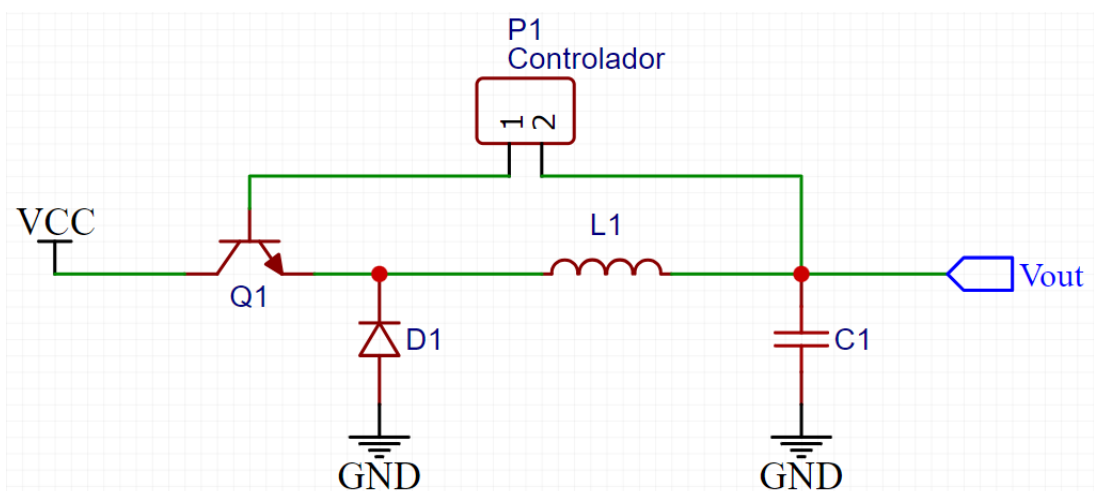
As placas perfuradas têm a vantagem de serem facilmente modificadas e adaptadas aos requisitos do circuito, permitindo a adição ou remoção de componentes eletrônicos e a alteração da disposição das trilhas. As placas perfuradas são uma escolha popular para prototipagem de projetos eletrônicos devido à sua flexibilidade e baixo custo em comparação com outras tecnologias de prototipagem (Raut, 2017).

No entanto, as placas perfuradas também têm algumas limitações, como a dificuldade em manter trilhas confiáveis em circuitos complexos e a possibilidade de curtos-circuitos acidentais entre trilhas próximas. Além disso, as placas perfuradas não são adequadas para produção em massa de circuitos eletrônicos devido à natureza manual do processo de prototipagem.

2.2.4 Conversão Amplitude de Tensão Dc Utilizando um Conversor *Buck*

Um conversor *Buck*, Figura 5, é um tipo de conversor *DC-DC* usado para reduzir uma tensão de entrada *DC* para uma tensão de saída *DC* menor. O conversor Buck consiste em um interruptor controlado, um indutor, um diodo e um capacitor de saída (Erickson; Maksimovic, 2001). O interruptor controlado é usado para alternar rapidamente a tensão através do capacitor, resultando em uma tensão de saída que é menor do que a tensão de entrada.

Figura 5 – Esquemático genérico conversor *Buck*



Fonte: O autor, 2023

Quando o interruptor está fechado, a corrente flui através do indutor e armazena energia magnética. Quando o interruptor é aberto, a energia armazenada no indutor e no capacitor. A tensão de saída é regulada controlando a frequência de chaveamento do interruptor e a largura do pulso de abertura. Segundo Sedra e Smith (2005), a relação entre a tensão de entrada e a tensão de saída em um conversor *Buck* é dado por (1):

$$V_{out} = \frac{D \times V_{in}}{(1-D)} \quad (1)$$

Em que V_{out} é a tensão de saída, V_{in} é a tensão de entrada e D o ciclo de trabalho do interruptor controlado². Esse ciclo é a proporção de tempo que uma

A eficiência do conversor *Buck* depende de vários fatores, incluindo a frequência de chaveamento, o valor do indutor e a resistência do circuito. A eficiência do conversor *Buck* pode ser superior a 90% em algumas aplicações (Erickson; Maksimovic, 2001).

2.3 FUNCIONALIDADES

As funções desempenhadas pelo dispositivo estudado são detalhadas a seguir. Cada uma representa uma atividade acessível através da comunicação remota sob requisição do cliente.

2.3.1 Internet das Coisas

O conceito de IoT se refere a uma rede de dispositivos interconectados que coletam e trocam dados. Segundo Atzori, Iera e Morabito (2010), o IoT é composto por uma infraestrutura de comunicação, dispositivos inteligentes e serviços e aplicativos baseados na nuvem que permitem que esses dispositivos sejam gerenciados e controlados.

² BARRETT, Steven F.; PACK, Daniel J. **Microcontrollers fundamentals for engineers and scientists**. Springer Nature, 2022.

Os dispositivos IoT são equipados com sensores e atuadores que coletam e processam dados em tempo real, permitindo a automação de processos e a tomada de decisões mais inteligentes. De acordo com Gubbi et al. (2013), a *IoT* pode ser aplicada em vários setores, como saúde, transporte, energia, agricultura e meio ambiente, para melhorar a eficiência, a segurança e a qualidade de vida.

A interoperabilidade é um dos principais desafios da *IoT*. De acordo com Bandyopadhyay e Sen (2011), a interoperabilidade refere-se à capacidade dos dispositivos *IoT* de se comunicarem e colaborarem uns com os outros, independentemente da plataforma, fabricante ou protocolo de comunicação. A padronização e a normalização dos protocolos de comunicação são essenciais para garantir a interoperabilidade na *IoT*.

2.3.2 Sistema de Atualização *Over-the-air*

O sistema de atualização *OTA* (*Over-The-Air*) é uma tecnologia que permite a atualização de software em dispositivos sem a necessidade de conexão física. Esse sistema é utilizado em dispositivos *IoT* e sistemas embarcados para atualização de *firmware*, correção de bugs e implementação de novos recursos.

De acordo com Gupta e Sharma (2021), o *OTA* é uma tecnologia que permite a atualização de software em dispositivos de forma remota e segura, sendo utilizado em uma ampla gama de dispositivos *IoT*, desde veículos até eletrodomésticos inteligentes. Ele é realizado através da transmissão de atualizações de software pela rede sem fio, permitindo que os dispositivos sejam atualizados automaticamente, sem a necessidade de intervenção do usuário.

O processo de atualização *OTA* envolve o envio de um arquivo de atualização para o dispositivo a ser atualizado, seguido pela verificação da integridade do arquivo e instalação da atualização. Segundo Wang et al. (2020), um dos principais benefícios do sistema *OTA* é a sua capacidade de atualizar dispositivos remotamente, sem a necessidade de interação do usuário. Além disso, ele permite

que as atualizações sejam realizadas de forma segura, evitando a possibilidade de corrupção de dados ou falha no processo de atualização.

2.3.3 Data Logging

O *data logging* é o processo de coleta, armazenamento e análise de dados em tempo real ou posteriormente. Essa técnica é utilizada em várias áreas, como engenharia, ciência, agricultura, ambiental, transporte, entre outras, para monitorar e analisar dados e tomar decisões com base nesses dados, Impac (2023).

O processo de *data logging* envolve a seleção dos sensores adequados, a calibração dos sensores, a configuração do *hardware* de aquisição de dados, a configuração do software de aquisição de dados e o armazenamento e análise dos dados. De acordo com Liu e Zhao (2019), a seleção dos sensores é importante para garantir que os dados coletados sejam precisos e confiáveis. A calibração dos sensores é necessária para minimizar os erros de medição e garantir a precisão dos dados coletados.

A configuração do *hardware* de aquisição de dados envolve a seleção do equipamento adequado, como placas de aquisição de dados, conversores analógico-digitais (ADC) e amplificadores, e a conexão dos sensores ao *hardware* de aquisição de dados. A configuração do software de aquisição de dados envolve a seleção do software adequado para aquisição e armazenamento de dados, bem como a configuração dos parâmetros de aquisição, como taxa de amostragem e tempo de amostragem.

O armazenamento e análise dos dados envolvem a transferência dos dados do *hardware* de aquisição de dados para um dispositivo de armazenamento, como um computador ou servidor, e a análise dos dados para obter informações úteis. A análise de dados pode ser feita usando ferramentas de análise de dados, como planilhas, softwares de processamento de sinais ou de aprendizado de máquina.

2.4 PROTOCOLOS DE COMUNICAÇÃO

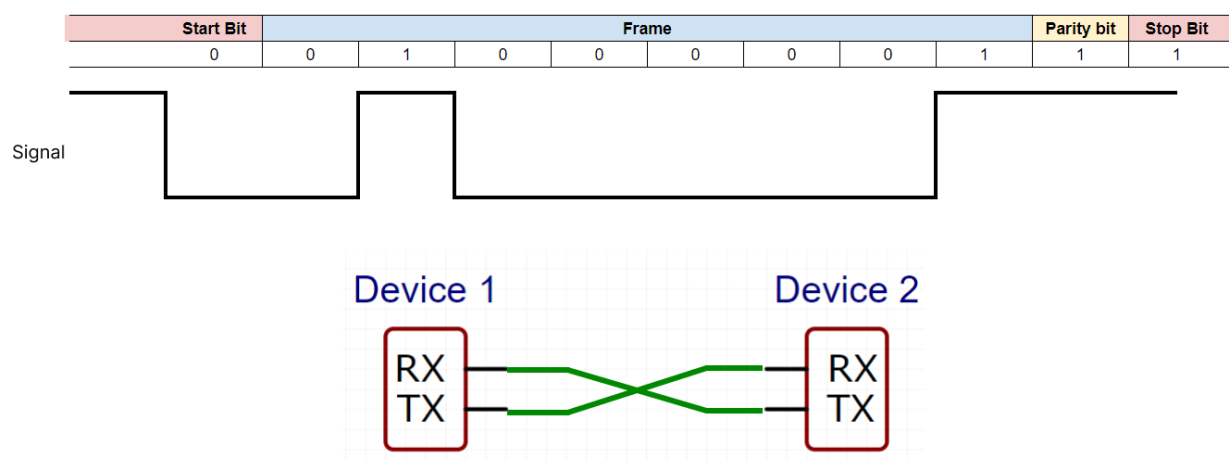
A integração de componentes desse projeto utilizam protocolos específicos para troca de dados entre circuitos integrados internos e módulos eletrônicos externos. Dessa forma, se faz necessário discutir as tecnologias utilizadas para completo entendimento do trabalho.

2.4.1 UART

UART (Universal Asynchronous Receiver/Transmitter) é um protocolo de comunicação serial assíncrono utilizado em sistemas embarcados, dispositivos eletrônicos e comunicações de computador.

Segundo Chappell (1993), o protocolo *UART* é baseado em uma comunicação serial assíncrona em que os dados são transmitidos em série, um bit de cada vez, Figura 6. A comunicação é assíncrona porque não há um sinal de *clock* compartilhado entre os dispositivos em comunicação, ao contrário de outros protocolos como (*Serial Peripheral Interface*) *SPI* e (*Inter-integrated circuit*) *I²C*.

Figura 6 – Sinal UART e camada física



Fonte: O autor, 2023

De acordo com Wakerly (1977), o protocolo *UART* envia os dados em pacotes, chamados de *frames*, que contém um bit de início, um número fixo de bits de dados, um ou dois bits de paridade e um ou mais bits de parada. O bit de início é sempre lógica 0 e é usado para sincronizar o receptor com o transmissor. Os bits de dados contém os dados propriamente ditos, enquanto os bits de paridade são opcionais e são usados para detectar erros de transmissão. Os bits de parada indicam o fim do *frame* de dados.

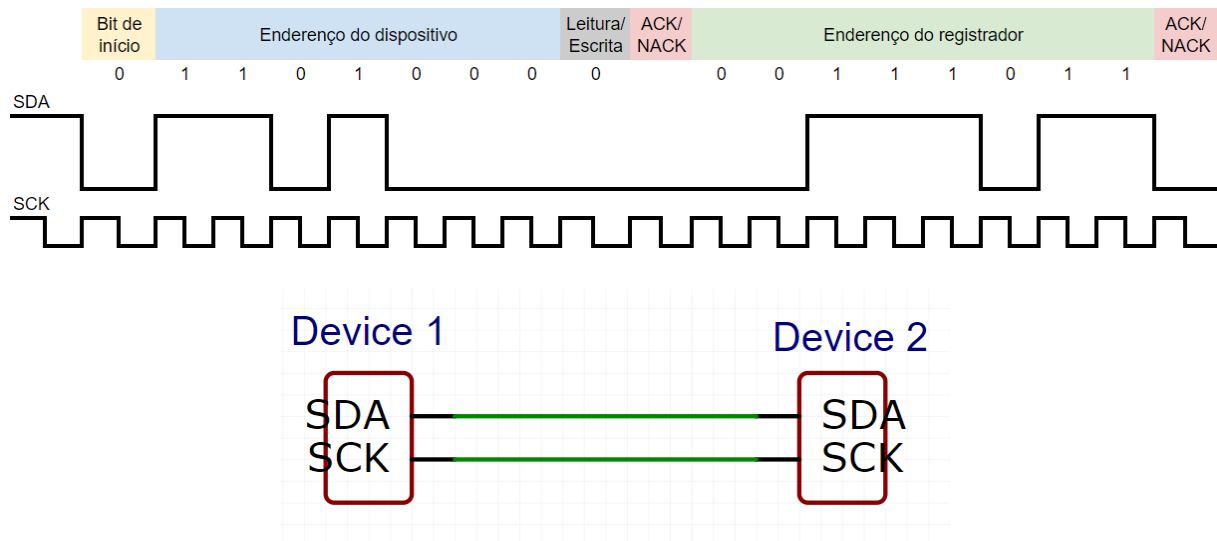
O protocolo *UART* é fácil de implementar em *hardware* e *software*, tornando-o um protocolo popular em aplicações de baixa velocidade e de baixo custo. Ele é utilizado em microcontroladores, sensores, módulos de comunicação sem fio e outros dispositivos eletrônicos.

2.4.2 I²C

O protocolo *I²C* é um protocolo de comunicação serial síncrono usado para conectar dispositivos em sistemas embarcados e em eletrônica em geral.

De acordo com *Datasheet I²C* (2021), o protocolo *I²C* utiliza apenas duas linhas de comunicação: uma linha de dados bidirecional (*SDA*) e uma linha de *clock* (*SCL*). A linha de *clock* é usada para sincronizar a comunicação entre os dispositivos, enquanto a linha de dados é usada para o tráfego de dados, Figura 7.

Figura 7 – Sinal I²Ce camada física



Fonte: O autor, 2023

O protocolo *I²C* utiliza um endereço de 7 bits para identificar os dispositivos conectados à rede. Cada dispositivo tem um endereço único que é usado para direcionar a comunicação para o dispositivo correto. O protocolo *I²C* também permite a comunicação entre mestre e escravo, onde o mestre controla a comunicação e o escravo responde às solicitações do mestre.

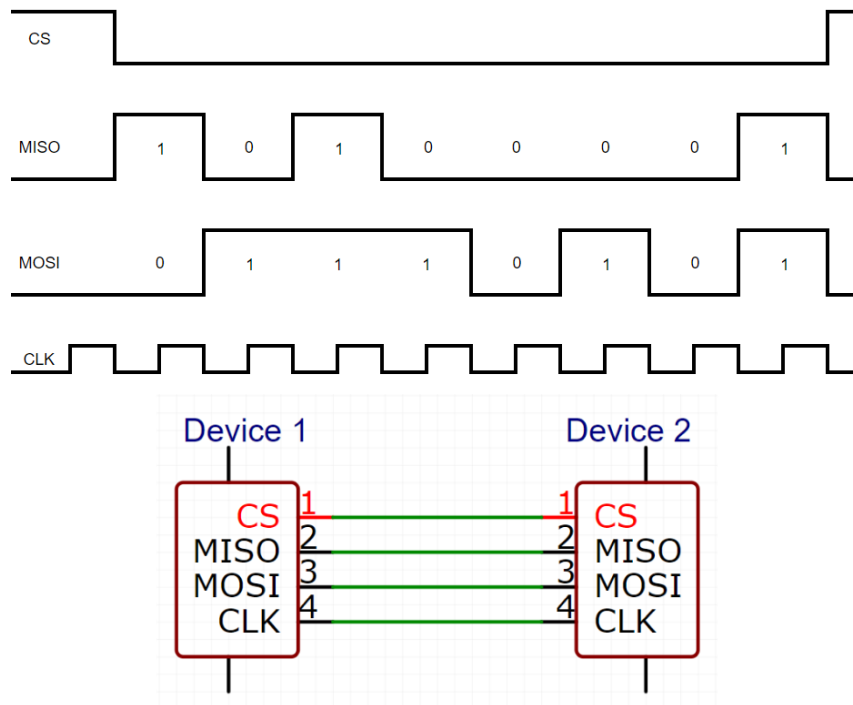
O protocolo *I²C* é utilizado em sistemas embarcados e em eletrônica de consumo devido à sua simplicidade, baixo consumo de energia e suporte para múltiplos dispositivos conectados em uma única rede.

2.4.3 SPI

Assim como o *I²C*, o protocolo *SPI* é um protocolo de comunicação serial síncrono usado para conectar dispositivos em sistemas embarcados em um circuito eletrônico.

De acordo com a documentação oficial do protocolo *SPI* (*SPI Interface Specification*, 2018), o protocolo utiliza quatro linhas de comunicação, Figura 8: uma linha de clock (*SCK*), uma linha de saída do mestre para o escravo (*MOSI*), uma linha de entrada do escravo para o mestre (*MISO*) e uma linha de seleção de escravo (*SS*) para escolher qual dispositivo se deseja se comunicar.

Figura 8 – Sinal SPI e camada física



Fonte: O autor, 2023

A linha de clock é usada para sincronizar a comunicação entre os dispositivos, enquanto as linhas *MOSI* e *MISO* são usadas para enviar e receber dados, respectivamente. O protocolo *SPI* suporta comunicação *full-duplex*, permitindo a transferência de dados em ambas as direções simultaneamente.

O protocolo *SPI* é utilizado em sistemas embarcados e em eletrônica de consumo devido à sua alta velocidade de comunicação, suporte para múltiplos dispositivos conectados em uma única rede e baixo consumo de energia.

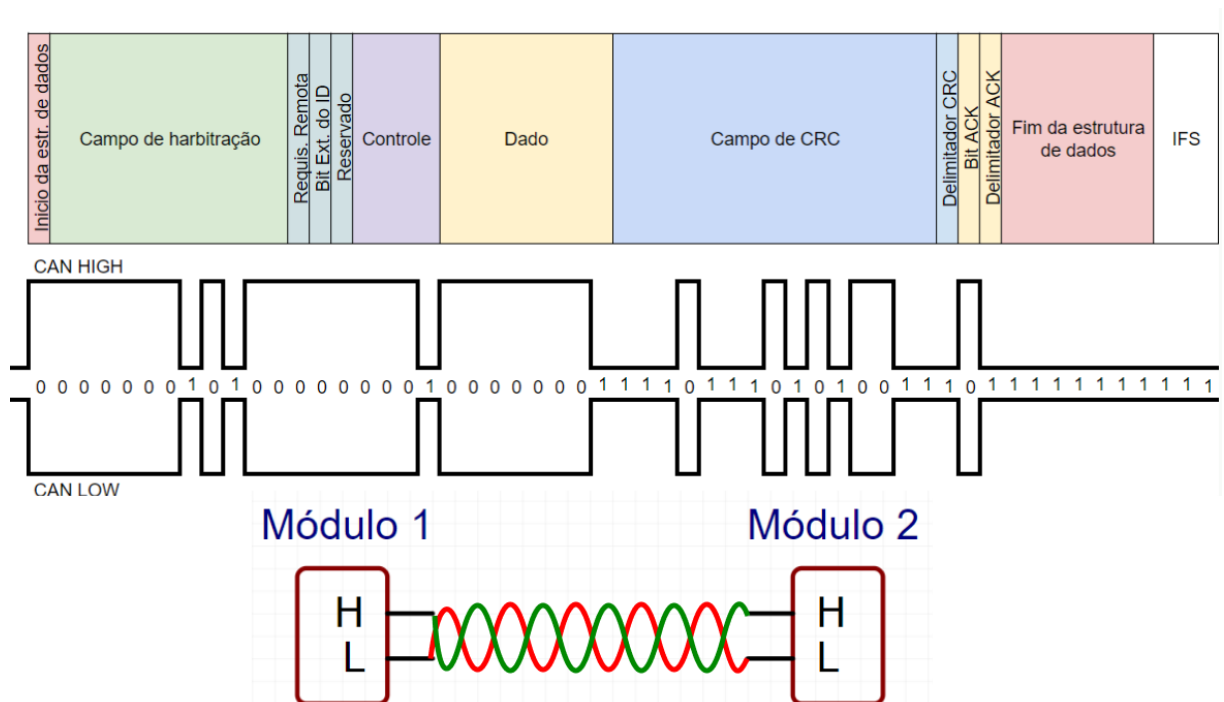
2.4.4 CAN

O protocolo *CAN* (*Controller Area Network*) é um protocolo de comunicação serial usado para comunicação entre dispositivos em rede, comumente utilizado em sistemas embarcados para veículos e outros equipamentos industriais.

De acordo com a documentação oficial do protocolo CAN (CAN Specification, 2018), o protocolo utiliza um esquema de comunicação baseado em mensagens que são transmitidas por nós individualmente na rede. O protocolo permite a comunicação de dados entre dispositivos a uma taxa de transferência de até 1 Mbps, com um máximo de 8 bytes de dados em uma única mensagem.

Sua estrutura física é composto por dois barramentos complementares trançados e blindados para evitar interferências eletromagnéticas. Os sinais complementares atingem amplitudes de 2.5V até 3.75V para o sinal de CAN *HIGH* e 2.5V até 1.25V para o *LOW*. A composição dos dois sinais indicam um estado recessivo (0) quando *HIGH* está em 3.75V e *LOW* esté em 1.25V. Quando esses barramentos se encontram em um estado dominante (1), *HIGH* e *LOW* apresentam valores de 2.5V, Figura 9.

Figura 9 – Sinal CAN e camada física



Fonte: O autor, 2023

O protocolo *CAN* utiliza um esquema de arbitragem distribuída para gerenciar a comunicação entre dispositivos na rede, permitindo que cada dispositivo tenha igualdade de oportunidade para transmitir dados. O protocolo também fornece mecanismos para detecção e correção de erros de transmissão de dados, garantindo uma comunicação confiável entre dispositivos.

A relevante presença³ deste tipo de comunicação nos produtos automotivos de mercado orienta os estudos desse trabalho para desenvolvimento de *software* e *hardware* em torno deste recurso, sendo explicado

2.4.5 MQTT

O protocolo *MQTT* (*Message Queuing Telemetry Transport*) é um protocolo de comunicação de mensagens leves projetado para ser usado em redes de sensores e dispositivos *IoT*.

De acordo com a documentação oficial do protocolo *MQTT* (OASIS, 2014), o protocolo utiliza um modelo de publicação/assinatura (*publish/subscribe*) para permitir que dispositivos publiquem mensagens em tópicos (*topics*) específicos e que outros dispositivos se inscrevam nesses tópicos para receber as mensagens correspondentes. O protocolo é baseado em um modelo cliente-servidor e utiliza uma arquitetura distribuída para garantir a comunicação entre dispositivos.

O protocolo *MQTT* é conhecido por sua eficiência e escalabilidade em redes de dispositivos *IoT*. Ele é projetado para ser leve em termos de recursos de rede e processamento, permitindo que dispositivos *IoT* com recursos limitados possam utilizar o protocolo. Além disso, o protocolo possui recursos de segurança robustos, como autenticação e criptografia, garantindo a privacidade e integridade das mensagens transmitidas na rede.

³ Disponível em:

<can-newsletter.org/engineering/engineering-miscellaneous/170116_market-research-can-and-its-com-petitors-in-cars_mam> Acesso em: 03 set, 2023

O protocolo *MQTT* é utilizado em aplicações de *IoT*, como monitoramento e controle remoto de dispositivos e sensores, automação residencial e industrial, entre outros.

2.4.6 IEEE 802.11 B/G/N/E/I

O padrão *IEEE 802.11* é um conjunto de especificações que define as tecnologias de redes sem fio (*Wi-Fi*). O padrão *IEEE 802.11* possui várias variações, incluindo os padrões b/g/n/e/i.

De acordo com a especificação oficial do padrão *IEEE 802.11* (*IEEE*, 2012), o padrão 802.11b define a tecnologia *Wi-Fi* que opera na faixa de frequência de 2,4 GHz, com velocidades de transmissão de até 11 Mbps. O padrão 802.11g é uma evolução do padrão 802.11b, com melhorias na taxa de transferência de dados, alcançando velocidades de até 54 Mbps. O padrão 802.11n é uma evolução dos padrões anteriores, com melhorias significativas na taxa de transferência de dados, alcançando velocidades de até 600 Mbps. O padrão 802.11e define melhorias na qualidade de serviço (QoS) para aplicações que requerem maior largura de banda e menor latência. Já o padrão 802.11i define melhorias na segurança das redes Wi-Fi, com a introdução do protocolo de criptografia *AES* (*Advanced Encryption Standard*).

Além dos padrões mencionados acima, o padrão *IEEE 802.11* também define outros recursos, como a capacidade de *roaming* (permitindo que dispositivos se conectem a diferentes pontos de acesso *Wi-Fi*), a capacidade de *beamforming* (permitindo a transmissão de sinal direcionado para dispositivos específicos) e a capacidade de agregação de quadros (permitindo a transmissão de múltiplos quadros de dados em uma única transmissão).

2.5 REDES DE COMUNICAÇÃO

Se tratando de um dispositivo de telemetria, explicar os tipos de comunicação e transferência de dados para um servidor remoto é essencial. Nesta seção, é discutido os métodos disponíveis e utilizados para realizar essa troca de dados entre o dispositivo e um servidor remoto.

2.5.1 GSM

A rede de comunicação *GSM* (*Global System for Mobile Communications*) é um padrão para redes de telefonia móvel de segunda geração (2G) que foi desenvolvido pela *European Telecommunications Standards Institute* (ETSI) (Mouly & Pautet, 1992).

O sistema *GSM* é composto por várias partes, incluindo a estação móvel (*MS*), a estação base (*BS*) e a rede de comutação de circuitos (*MSC*). A estação móvel é o telefone celular, a estação base é a torre de celular e a *MSC* é o ponto central da rede que gerencia as conexões de chamadas.

O sistema *GSM* utiliza a modulação de amplitude em quadratura (*QAM*) para transmitir dados digitalmente. Os dados são divididos em pacotes e transmitidos em intervalos de tempo de quadros de 4,6 milissegundos. Cada quadro é composto por 8 slots, e cada slot pode conter um pacote de dados.

O padrão *GSM* também inclui medidas de segurança, como a criptografia dos dados transmitidos entre a estação móvel e a estação base, para garantir a privacidade dos usuários.

2.5.2 GPRS

A Rede *GPRS* (*General Packet Radio Service*) é uma tecnologia de rede de dados sem fio que foi desenvolvida como uma extensão do sistema *GSM* de segunda geração (2G) (Holma & Toskala, 2001).

A *GPRS* permite que os dados sejam transmitidos em pacotes, em vez de circuitos, como é o caso das chamadas de voz, o que torna o uso da rede mais eficiente. A transmissão de dados ocorre em intervalos de tempo, em que cada pacote é transmitido individualmente e, em seguida, reagrupadas no destino.

A rede *GPRS* é capaz de oferecer taxas de transmissão de dados de até 114 kbps, o que é suficiente para serviços básicos de internet móvel, como navegação na *web* e *email*. Além disso, a tecnologia *GPRS* oferece melhorias na segurança e eficiência do uso da rede em comparação com o *GSM*.

2.5.3 NB-IOT

A rede *Narrowband Internet of Things (NB-IoT)* é uma tecnologia de comunicação sem fio desenvolvida especificamente para dispositivos de Internet das Coisas. O *NB-IoT* é uma variante do *LTE (Long-Term Evolution)*, projetado para operar em frequências mais baixas, usando menos largura de banda e, portanto, permitindo que os dispositivos se comuniquem com uma baixa taxa de dados (Chafii et al., 2018).

A principal vantagem do *NB-IoT* é sua capacidade de operar em locais de difícil alcance, como porões e áreas remotas, tornando-se uma opção atraente para aplicações de *IoT* em que o alcance é um fator crítico. O *NB-IoT* também é projetado para fornecer eficiência energética para dispositivos de baixa potência, com bateria durável e baixa manutenção.

A tecnologia *NB-IoT* foi padronizada pela *3GPP* em 2016 e, desde então, vem sendo adotada por operadoras em todo o mundo como uma alternativa econômica e eficiente para conectar dispositivos *IoT*.

2.5.4 CAT-M1

A rede de comunicação *CAT-M1 (Category M1)* é uma tecnologia de comunicação sem fio de baixa potência desenvolvida para suportar dispositivos de *IoT* que requerem conexão constante e dados em tempo real. Ela usa as redes celulares existentes para fornecer conectividade confiável e segura para uma ampla gama de aplicações *IoT*.

Segundo a Ericsson (2020), a *CAT-M1* suporta comunicação em duas direções, permitindo que dispositivos enviem e recebam dados. Ela também possui recursos de segurança aprimorados em relação às tecnologias de *IoT* mais antigas, como a *GSM* e a *GPRS*, com criptografia de ponta a ponta e autenticação de dispositivos.

A *CAT-M1* é uma alternativa popular para dispositivos *IoT* que exigem baixo consumo de energia e longa duração de bateria, como dispositivos de monitoramento remoto e rastreamento de ativos. Além disso, ela é suportada por muitos provedores de rede em todo o mundo.

2.6 ARQUITETURAS DE PROCESSADORES

Neste projeto, são utilizados dois MCUs com finalidades diferentes, sendo necessário, portanto, entender como funciona a arquitetura de cada um. Neste capítulo, serão explicados os modelos AVR RISC e o Xtensa LX6.

2.6.1 AVR RISC

A arquitetura AVR (*Alf and Vegard's RISC*) é uma arquitetura RISC (Reduced Instruction Set Computing) desenvolvida pela empresa Atmel. Os processadores AVR são baseados em uma arquitetura de 8 bits, mas também possuem modelos de 32 bits. A arquitetura AVR é caracterizada por ser uma arquitetura de baixo consumo de energia, ter uma instrução por ciclo de clock, registradores de propósito geral em grande quantidade e um conjunto de instruções de tamanho reduzido.

Segundo um estudo de A.I-Bakry (2016), a arquitetura AVR possui um pipeline de três estágios, o que permite um desempenho elevado. Além disso, a arquitetura também possui um conjunto de instruções otimizado para suportar operações de bit, o que a torna ideal para aplicações que envolvem controle de dispositivos eletrônicos. Os autores também ressaltam que a arquitetura AVR é utilizada em uma ampla variedade de dispositivos, como microcontroladores, sensores e sistemas embarcados em geral.

Outro estudo de Akashe et al. (2017) aponta que a arquitetura *AVR* também é muito utilizada na indústria de *IoT*, devido ao seu baixo consumo de energia e tamanho reduzido. Os autores mencionam que a arquitetura *AVR* tem sido utilizada em dispositivos *IoT* que possuem requisitos de processamento modestos, como sensores e dispositivos de monitoramento de temperatura.

2.6.2 Xtensa LX6

A arquitetura *XTensa LX6* é um conjunto de instruções de processador de 32 bits desenvolvido pela empresa Tensilica. O LX6 é uma arquitetura altamente configurável que permite a personalização do processador para atender às necessidades específicas de diferentes aplicações.

O *LX6* também apresenta um pipeline de seis estágios para uma execução eficiente de instruções e inclui caches separados para instruções e dados para uma maior eficiência na leitura e escrita de dados. Além disso, a arquitetura *LX6* possui recursos de gerenciamento de energia que permitem um ajuste fino do consumo de energia do processador, tornando-o adequado para aplicações com requisitos rigorosos de energia.

2.7 SISTEMA OPERACIONAL E FRAMEWORKS

Visto os MCUs que são utilizados, neste capítulo, é explicado qual ferramenta é utilizada para gerenciamento das tarefas desempenhadas pelo MCU e qual estratégia de sistema é utilizada para a interface entre o usuário e a ECU desenvolvida.

2.7.1 Sistema Operacional em Tempo Real

Real-time Operating Systems (RTOS) são sistemas operacionais projetados para lidar com requisitos em tempo real em que as tarefas precisam ser executadas dentro de prazos rigorosos, GONZALEZ (2003). Eles são utilizados em sistemas

embarcados, como controladores de processo industrial, sistemas de controle de tráfego e equipamentos médicos.

Um *RTOS* oferece um mecanismo de gerenciamento de tarefas e recursos que pode garantir que as tarefas sejam executadas dentro dos prazos, e é projetado para oferecer baixa latência e alta confiabilidade. O *RTOS* é responsável por determinar qual tarefa deve ser executada em um determinado momento, alocando recursos de forma eficiente e garantindo que as tarefas sejam executadas dentro dos prazos estabelecidos.

Existem vários *RTOS* disponíveis, como *FreeRTOS*, *VxWorks* e *RTLinux*. Cada um desses sistemas tem suas próprias características e recursos que os tornam adequados para diferentes aplicações.

2.7.2 Framework de Desenvolvimento React.js

O *React* é uma biblioteca de *JavaScript* utilizada para construir interfaces de usuário. Ele foi desenvolvido pelo *Facebook* e é utilizado em todo o mundo para a criação de aplicativos da *web*. A principal característica do *React* é a sua capacidade de criar componentes reutilizáveis que podem ser combinados para formar interfaces complexas. Além disso, o *React* utiliza o modelo de programação declarativo, em que os desenvolvedores descrevem como a interface deve parecer e o *React* cuida de atualizar a interface conforme as mudanças de estado.

3 DESENVOLVIMENTO DO PROJETO

Este capítulo é iniciado com a descrição da seleção dos componentes utilizados baseados em suas funcionalidades e a forma de integração necessária para a correta comunicação entre todos eles. É detalhado também, o design voltado para a plataforma de testes utilizada para acelerar as validações realizadas no módulo desenvolvido. A descrição da máquina de estados utilizada pelo *firmware* também é alvo da discussão a seguir, bem como todas as técnicas utilizadas através do sistema operacional de tempo real utilizado (*FreeRTOS*⁴) para gerenciamento de

⁴Disponível em: <www.freertos.org> Acesso em: 03 set, 2023

fluxo de dados. Por fim, o design do invólucro plástico bem como todas as técnicas utilizadas para manter a integridade física do protótipo finaliza a discussão das metodologias implementadas neste projeto.

3.1 DEFINIÇÃO DO FUNCIONAMENTO DO PROJETO

Dentro da primeira macro-etapa, é necessário definir quais funções são desempenhadas pelo protótipo fabricado. A partir dos objetivos específicos, os métodos desenvolvidos de forma a viabilizar total cumprimento das premissas são detalhados da seguinte forma:

- A. Para desenvolver um método de comunicação eficiente entre o módulo desenvolvido e os diversos sensores utilizados no veículo, foi utilizado uma rede *CAN* de comunicação com as demais *ECUs* (*Electronic Control Unit*) do veículo;
- B. Para desenvolver um método de mensurar a orientação do veículo, foi utilizado sensores de orientação e aceleração;
- C. Para desenvolver um método de comunicação entre o módulo desenvolvido e algum servidor remoto, foi utilizado o protocolo *MQTT*;
- D. Para desenvolver um método de armazenamento de dados em tempo real, é utilizado um cartão *SD* para salvar as informações em formato *csv*;
- E. Para desenvolver um método de comunicação *peer-to-peer*, é utilizado o padrão *IEEE 802.11 b/g/n/e/i*;
- F. Para desenvolver um método de atualização remota para o *firmware* do módulo desenvolvido, é utilizado o método de *OTA* através de uma plataforma dedicada;
- G. Para manter baixos níveis de consumo energético e de dados, o fluxo de informação é otimizado de forma a somente manter os *hardwares* periféricos em funcionamento quando for solicitado;

3.2 SELEÇÃO DOS COMPONENTES

Nesta etapa, é descrito a seleção dos componentes que permitiram o correto funcionamento do protótipo. Como o desenvolvimento de uma *PCB* não é o escopo deste trabalho, o fator compacidade foi levado em consideração na seleção de todos os componentes. Isso se deve a necessidade de obter um protótipo funcional que ainda fosse facilmente encaixado nas motocicletas.

3.2.1 Seleção do Conversor *Buck* XL7015

Como este projeto deve estar preparado para lidar com tensões entre 12V e 80V, deve-se selecionar um conversor adequado para essa aplicação. Nesse sentido, foi selecionado o conversor *buck* XL7015, Figura 10, dentre outras opções como o LM2596, Figura 11, e o MC7800, Figura 12. A eficiência do conversor XL7015 pode chegar até 94% enquanto o LM2596 é limitada a 90%. Essa vantagem é ainda maior quando comparado com o MC7800 que é limitada a 70%, tendo a maior parte dissipada por calor. Além disso, o XL7015 tem melhor compacidade devido a sua área menor do que o LM. Finalmente, verifica-se a ampla faixa de tensão de entrada permitida para o XL, variando de 5V até 80V. O esquemático do conversor escolhido pode ser visto na Figura 13.

Figura 10 – XL7015



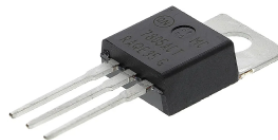
Fonte: www.alldatasheet.com

Figura 11 – LM2596



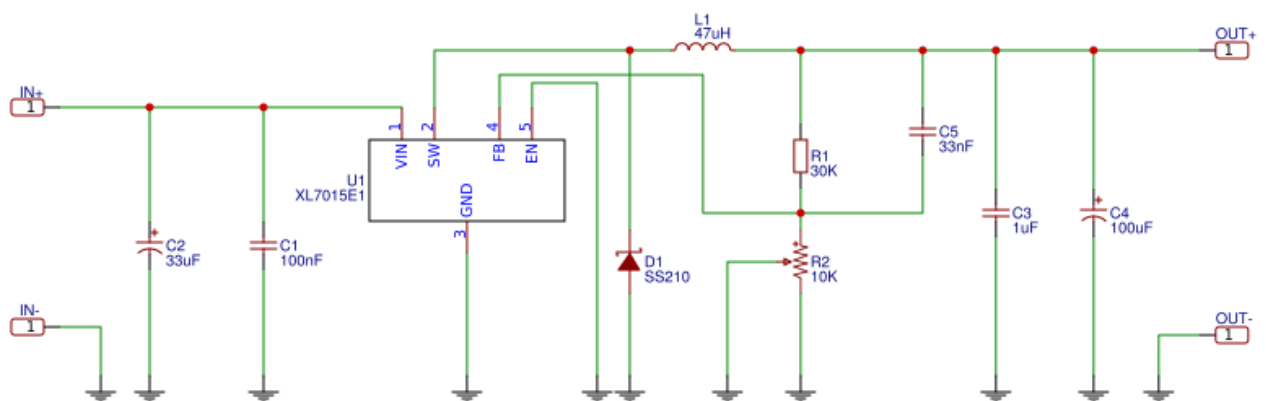
Fonte: www.alldatasheet.com

Figura 12 – MC7800



Fonte: www.alldatasheet.com

Figura 13 – Esquemático do XL7015



Fonte: www.easyeda.com

3.2.2 Seleção do Sensor Inercial MPU6050

Para mensurar a orientação e acelerações do veículo, foi utilizado um GY-521 que incorpora um acelerômetro e giroscópio através do módulo MPU6050, Figura 14. Essa solução foi comparada com a trazida pelo seu sucessor, o MPU9250, Figura 15, que adiciona um magnetômetro ao seu núcleo. Como o magnetômetro somente seria necessário para o cálculo da orientação em Yaw sem acúmulo de erro no tempo, o MPU9250 foi descartado por essa grandeza está fora do escopo do estudo. Além disso, o MPU9250 possui uma melhor compactidade devido a sua menor área. Os controladores desta família ainda possuem um *Digital Motion Processor (DMP)*, responsável pelo processamento dos dados do MPU6050 por fusão de sensores⁵, e entregar essas informações à placa de controle principal. Essa técnica é especialmente útil pois elimina a necessidade do processamento desses dados na placa de controle principal, economizando tempo e recurso do processador. A comunicação entre os dois módulos se dá através do protocolo *I²C*. O esquemático da placa que integra o módulo selecionado pode ser visto na Figura 16.

Figura 14 – MPU6050



Fonte: www.alldatasheet.com

⁵ Disponível em:

<https://ieeexplore.ieee.org/abstract/document/8107038?casa_token=iIFxEsGIAt8AAAAA:i7hmT-tgum9ApIZBvAAKZcKS-FarnbMNRtP6De6qIFa0qKtLOemnhYvWA0FN9aCPCG1f7VP4FZE> Acesso em: 03 out, 2023

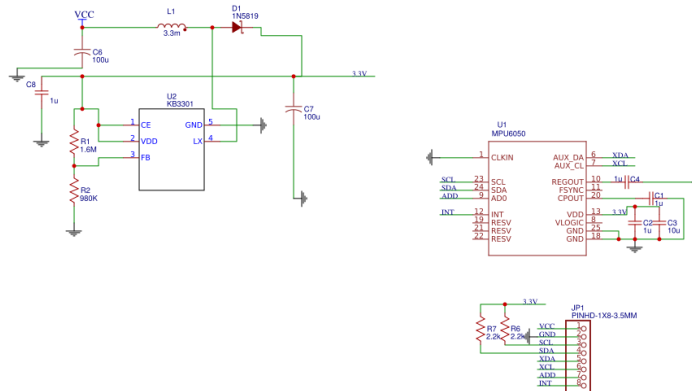
Figura 15 – MPU9250



22 × 17 mm

Fonte: www.alldatasheet.com

Figura 16 – Esquemático do MPU6050



Fonte: www.easyeda.com

3.2.3 Seleção do Controlador CAN SJA1000

Para o módulo de controle CAN, foi selecionado o SJA1000, Figura 17, entre módulos como o MCP2515, Figura 18. Essa escolha se deu devido ao primeiro possuir suporte para identificadores de 11-bits e 29-bits no protocolo CAN 2.0B, ao contrário do segundo que só é compatível com 11-bits. Além disso, o SJA1000 possui um controle de erro mais avançado do que o MCP. Isso porque o primeiro consegue responder de formas diferentes a depender da gravidade do erro (modos

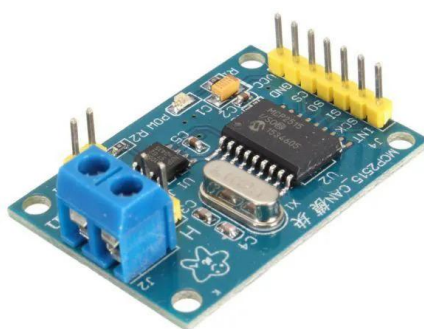
error-passive e *error-active*). A controladora escolhida ainda permite a realização de filtragem de mensagens por ID, otimizando os recursos de processamento da placa de controle principal. O diagrama de blocos do controlador selecionado pode ser visto na Figura 19.

Figura 8 – SJA1000



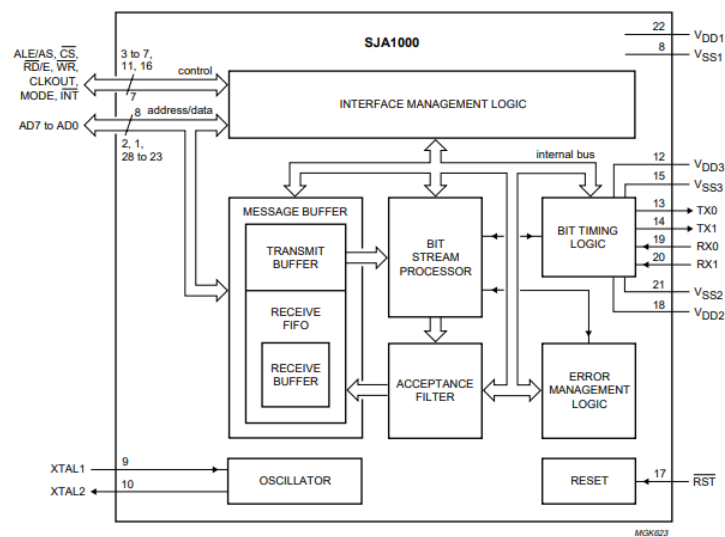
Fonte: www.mouser.com

Figura 9 – MCP2515



Fonte: www.alldatasheet.com

Figura 10 – Diagrama de blocos do SJA1000

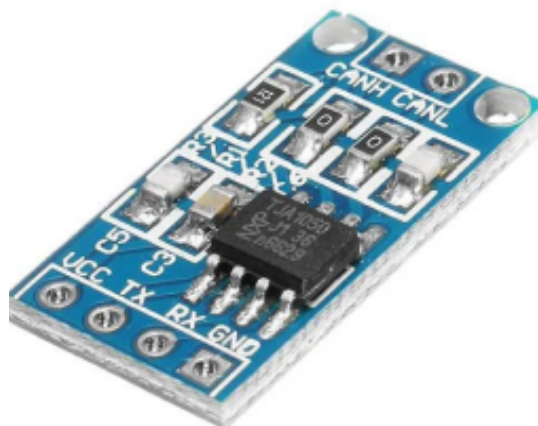


Fonte: www.nxp.com

3.2.4 Seleção do Transceptor CAN TJA1050

Juntamente ao controlador CAN, foi necessário escolher o transceiver que fosse interfacear o protocolo ao hardware. Para esse tópico, foram levados em consideração os módulos TJA1050, Figura 20, e o MCP2551, Figura 21. O primeiro, no entanto, foi selecionado por possuir maior taxa de transferência de dados que o segundo, sem que fosse necessário ajustar o divisor programável. Além disso, o TJA possui aproximadamente 17% do consumo de corrente quando comparado com o MCP. Por fim, o transceiver escolhido possui melhor compacidade devido a sua menor área. A comunicação desse componente com o processador se dá através do protocolo *UART*. O esquemático do transceiver escolhido pode ser visto na Figura 22.

Figura 20 – TJA1050



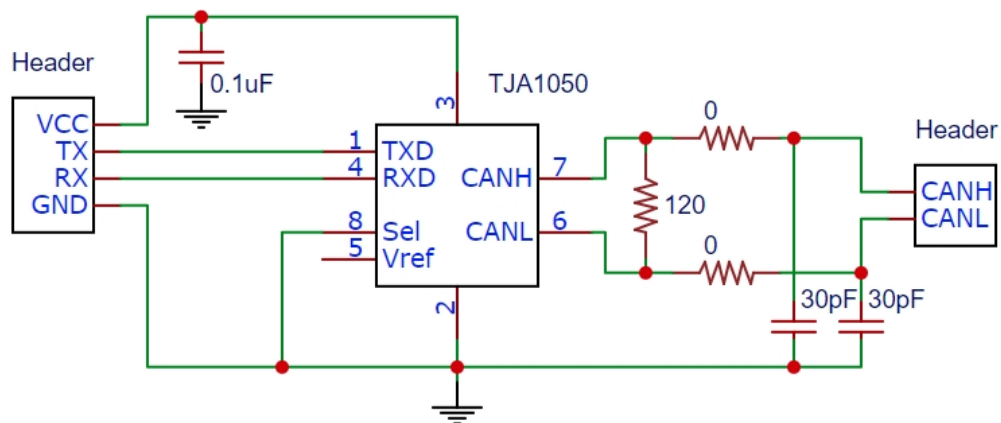
Fonte: www.nxp.com

Figura 21 – MCP2551



Fonte: www.mouser.com

Figura 22 – Esquemático TJA1050



Fonte: www.nxp.com

3.2.5 Seleção do Modem SIM7000G

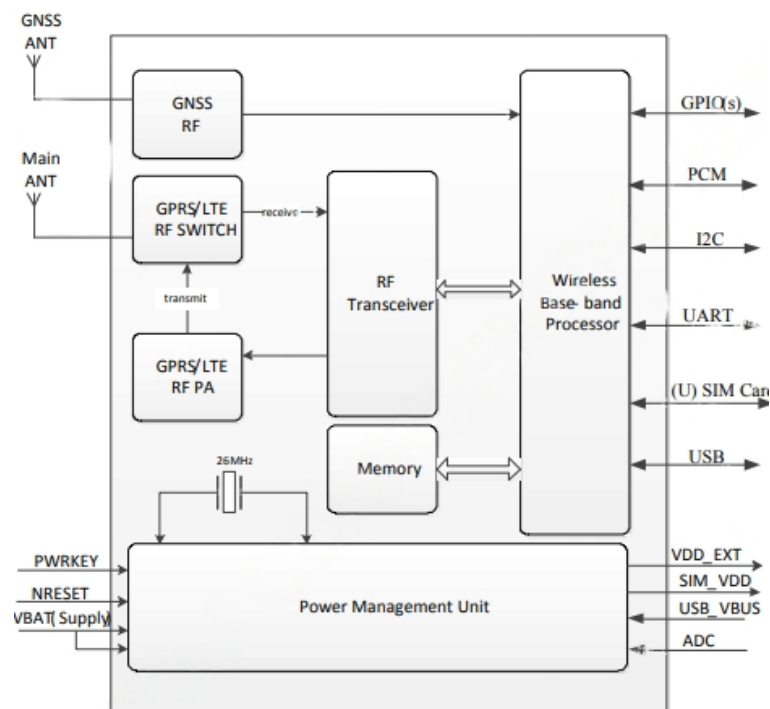
Para a seleção da comunicação móvel de banda larga, foi selecionado o módulo SIM 7000G, Figura 23. Esse chip é capaz de se comunicar em diferentes tipos de tecnologia, como *GPRS*, *LTE* e *NB/IoT*. Esse módulo também é capaz de coletar dados de *GNS* através de conexões com os sistemas *GPS* e *GLONASS*. Esses motivos fizeram com que o SIM7000G fosse escolhido, em detrimento de chips como o SIM800L e o Neo-6M. A comunicação com esse chip se dá através da sua interface *UART* via comandos *AT*, possibilitando uma interface completa para configuração e checagem de status desse módulo. O diagrama de blocos para esse componente pode ser visualizado na Figura 24.

Figura 23 – SIM7000G



Fonte: www.simcom.ee

Figura 24 – Diagrama de blocos do SIM7000G



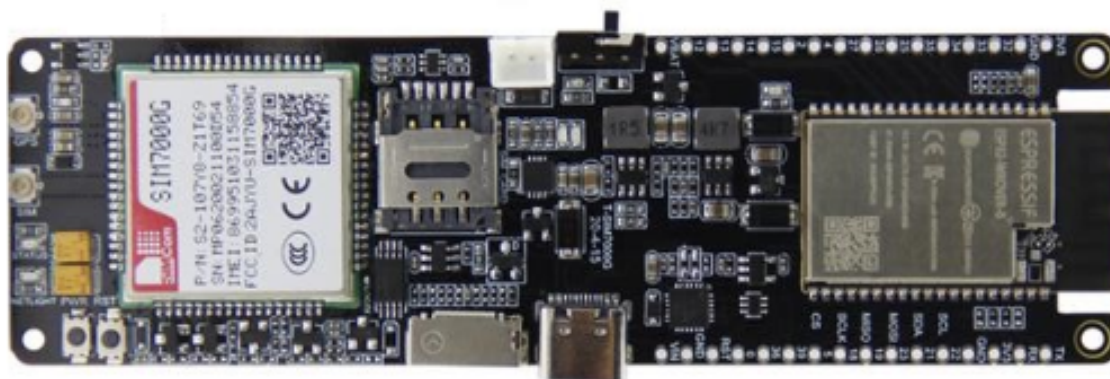
Fonte: www.simcom.ee

3.2.6 Seleção da Unidade Principal de Processamento Lilygo ESP32 Wrover-E SIM7000G

Para a placa de processamento, foi utilizada a ESP32 WROVER-E SIM7000G oferecida pela LILYGO, Figura 25. Essa escolha se deu devido a grande

disponibilidade de periféricos essenciais para o desenvolvimento deste projeto. Esta placa possui um conector para cartão de memória *TF*, uma controladora de comunicação *CAN* SJA1000, conectores para antenas *GPS* e *LTE*, soquete para cartão *SIM* e um módulo de comunicação móvel de banda larga SIM7000G. Essa placa de desenvolvimento apresenta um ótimo encapsulamento dos periféricos necessários para o desenvolvimento do projeto, restando poucos componentes adicionais a serem integrados, como o transceiver da *CAN*, o acelerômetro e a fonte chaveada. Aliado a tudo isso, essa placa de desenvolvimento possui uma controladora ESP32 WROVER-E que possui *clock* de até 240 MHz e dois núcleos de processamento, ideal para a gestão de múltiplas tarefas.

Figura 25 – LILYGO SIM7000G

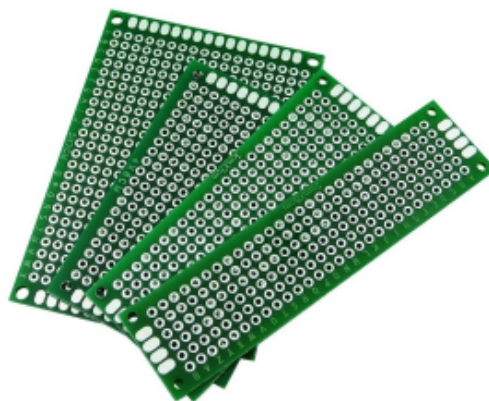


Fonte: www.lilygo.cn

3.4 INTEGRAÇÃO DOS COMPONENTES ELETRÔNICOS

Como foi mencionado anteriormente, a placa de controle principal escolhida, ESP32 SIM 7000G, já encapsula grande parte dos componentes necessários para o funcionamento do projeto, no entanto alguns componentes ainda precisam ser integrados através dos pinos de comunicação e alimentação presentes nessa controladora. Para isso, foi necessário o desenvolvimento de uma placa que agregasse o XL7015, GY-521 e o TJA1050 em um *shield* de fácil manuseio e fixação à placa principal. Para isso, foi utilizado uma placa perfurada de prototipação de 4 cm por 6 cm, Figura 26.

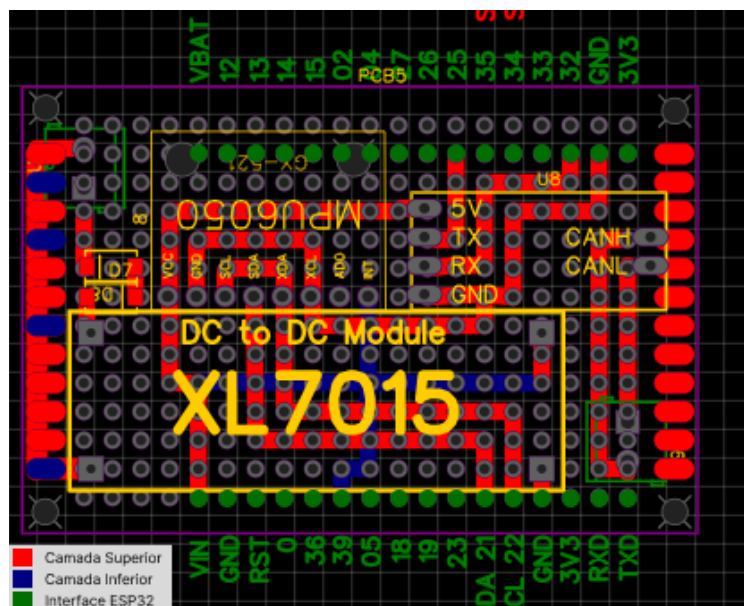
Figura 26 – Placas perfuradas



Fonte: www.mouser.com

Antes da montagem dos componentes, foi utilizado uma ferramenta online de *Electronic Design Automation (EDA)* EasyEDA⁶ para planejar a disposição dos componentes bem como o roteamento dos pinos de alimentação e comunicação, Figura 27.

Figura 27 – Esquemático da disposição dos periféricos



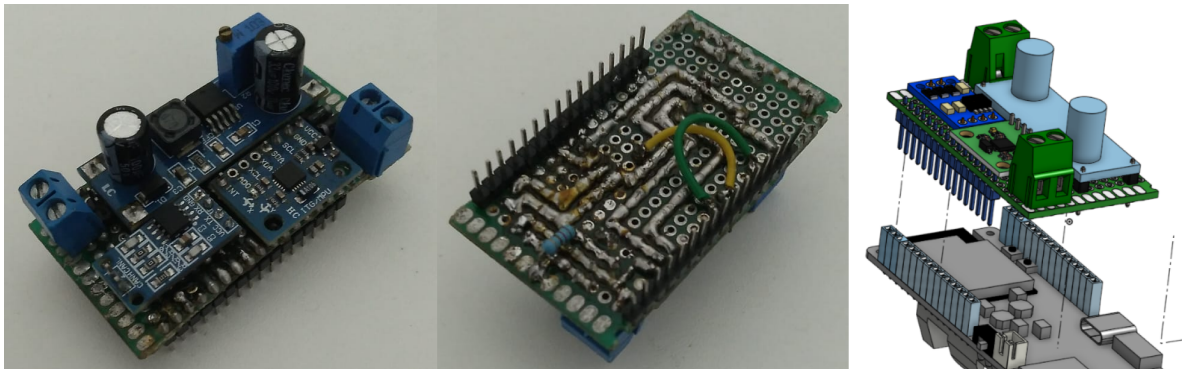
Fonte: O Autor, 2023

Por fim, os pinos macho de interface da placa perfurada à ESP32 foram soldados. Em seguida, o roteamento dos pinos foi realizado através do método de ponte de estanho, comunicando corretamente todos os pinos de comunicação *I²C* e *UART* necessários, além dos pinos de alimentação 5V e *GND*. Ao fim, foi verificado

⁶ Disponível em: <<https://oshwlab.com/mateus.lourenco/tcc-whistleblower>> Acesso em: 03 set, 2023

a existência de curto-circuitos utilizando a verificação da continuidade de trilhas com um multímetro. O resultado foi uma placa robusta e compacta que pode ser facilmente fixada à placa de processamento principal, Figura 28.

Figura 28 – Placa de periféricos montada e soldada

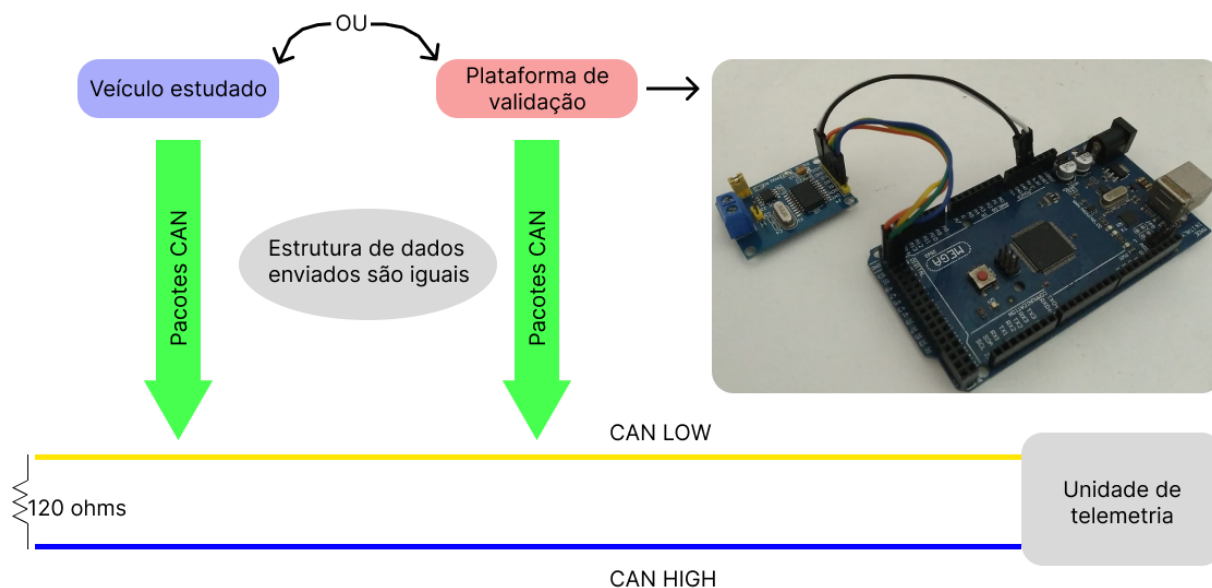


Fonte: O Autor, 2023

3.5 PLATAFORMA DE VALIDAÇÃO

Durante a fase de planejamento do projeto, foi verificado que seria necessário um equipamento que pudesse acelerar o desenvolvimento do software sem que o desenvolvedor precisasse estar próximo do veículo de estudo. Para isso, foi pontuada a necessidade do desenvolvimento de uma plataforma de validação, Figura 29, que pudesse acelerar o processo de criação do software. Assim, foi criado um módulo que pudesse emular o envio no barramento das mensagens CAN típicas do veículo estudado. O desenvolvimento desse item se encaixa no escopo deste projeto na medida em que ele é utilizado tanto durante o desenvolvimento quanto na validação deste trabalho.

Figura 29 – Esquemático da plataforma de validação



Fonte: O Autor, 2023

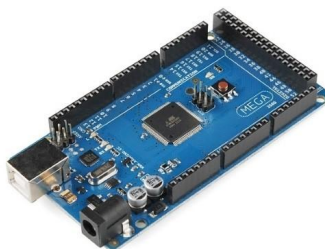
3.5.1 Seleção dos Componentes

Apesar de fazer parte do escopo, a seleção dos componentes deste módulo não tiveram o mesmo rigor utilizado para a seleção do módulo *IoT*. Isso pois este componente somente estaria sendo utilizado em um ambiente controlado de desenvolvimento e testes.

3.5.1.1 ARDUINO MEGA

A placa utilizada como central de processamento foi o Arduino MEGA, Figura 30, equipado de um microcontrolador ATmega 2560. Esse é um controlador que permite um desenvolvimento rápido e robusto para prototipação de projetos simples.

Figura 30 – Arduino MEGA



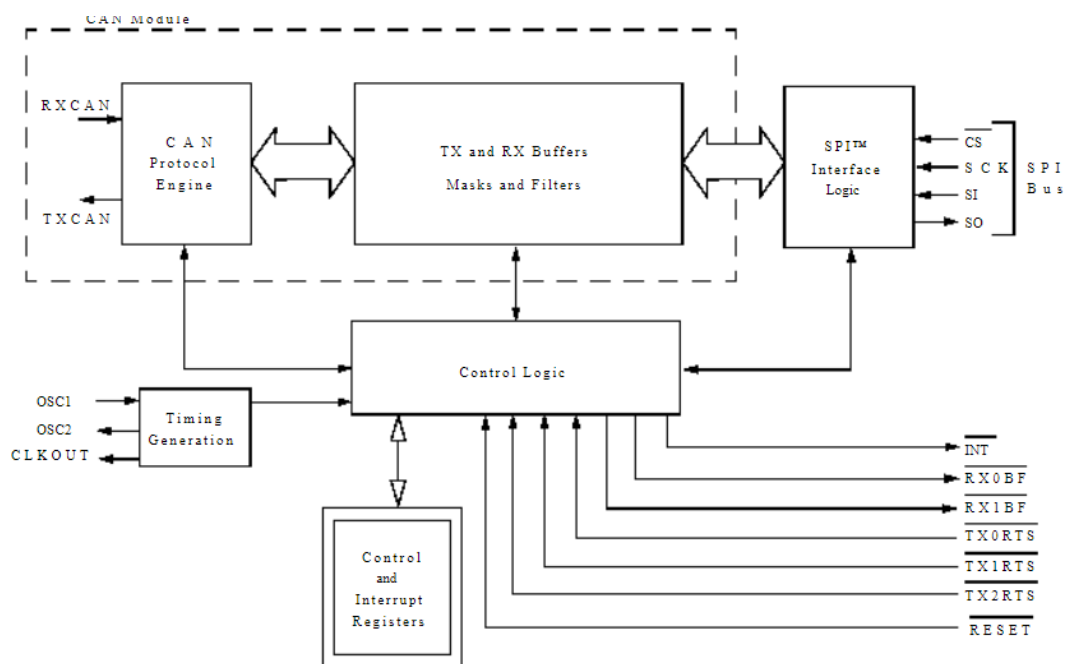
Fonte: www.arduino.cc

Esse microcontrolador foi programado para enviar mensagens típicas recebidas no barramento *CAN* do veículo sendo estudado, como os dados vindos da inversora, *BMS* e *cluster*.

3.5.1.2 MCP2515

Como a placa de processamento principal desse módulo de testes não possui uma controladora *CAN* incluída, foi necessário utilizar o módulo MCP2515, previamente descartado para a utilização no módulo *IoT*. Essa placa já possui a controladora *CAN*, bem como o transceiver e os componentes fundamentais para a integração entre esses componentes através do protocolo de comunicação *SPI*. O diagrama de blocos desse componente pode ser verificado através da Figura 31.

Figura 31 – Diagrama de blocos do MCP2515



Fonte: www.alldatasheet.com

3.5.2 Desenvolvimento do *Firmware*

O firmware embarcado nessa plataforma é mais simples do que a desenvolvida para a central eletrônica de telemetria do ponto de vista de software e hardware. Essa ferramenta utiliza o mesmo padrão de comunicação utilizado pela motocicleta estudada. São mensagens enviadas periodicamente endereçadas a módulos específicos. Essas mensagens seguem o padrão de *broadcasting*, em que todos os módulos têm acesso a todas as mensagens no barramento porém cada dispositivo filtra suas mensagens através do ID. É importante ressaltar que, como a intenção da ferramenta é emular o envio de mensagens, essas, por sua vez, não representavam valores reais, sendo somente utilizada para ajuste de formatação e controle de fluxo de dados.

3.6 FIRMWARE DO MÓDULO IOT

Nesta seção serão descritas as etapas necessárias para o desenvolvimento do software embarcado que coordena as tarefas executadas pelo módulo de telemetria. Também é descrita as ferramentas utilizadas para suportar a evolução dessa fase do projeto.

3.6.1 Seleção do Repositório e Sistema de Controle de Versão

Para sistema de versionamento e repositório, foi utilizado o GitHub⁷. Esta é uma ferramenta popular para o controle de versão e gerenciamento de projetos de desenvolvimento de software. A sua facilidade de uso e ampla gama de recursos o tornam uma escolha adequada para projetos de sistemas embarcados.

⁷ Disponível em: <<https://github.com/Humpkins/Whistleblower>> Acesso em: 03 set, 2023

3.6.2 Desenvolvimento do *Firmware*

O desenvolvimento do software embarcado neste projeto é descrito nesta seção. Serão detalhados as ferramentas de desenvolvimento de código, o sistema operacional selecionado para ser implementado no módulo e as ferramentas de gerenciamento de atualização remota. Além disso, a configuração do sistema operacional e sua integração com os *hardwares* utilizados também serão tratados nos tópicos a seguir.

3.6.2.1 Seleção do Ambiente de Desenvolvimento Integrado (*IDE*)

Para a seleção da *IDE*, foram levados em consideração ferramentas como Arduino *IDE*, Espressif *IDE* e o PlatformIO. O último foi selecionado pois o PlatformIO é uma plataforma de desenvolvimento de software embarcado que oferece uma gama de recursos avançados que o tornam uma escolha superior em relação aos outros avaliados para projetos complexos com diversas bibliotecas e *hardware* periférico. Com a *IDE* escolhida é possível integrar várias plataformas de *hardware* e bibliotecas em um único ambiente de desenvolvimento, facilitando o processo de compilação e *upload* do *firmware*. Além disso, o PlatformIO⁸ oferece uma interface de linha de comando poderosa e suporte para múltiplas linguagens de programação, o que o torna uma ferramenta mais flexível e eficiente para a complexidade do projeto proposto.

3.6.2.2 Seleção do *Framework* de Desenvolvimento do *Firmware*

Para a escolha do *framework* de desenvolvimento, foram considerados o ESP-IDF e o Arduino *framework*. O último foi selecionado pois é uma escolha apropriada para o desenvolvimento do software embarcado complexo exigido pelo projeto. Isso se deve em parte à grande quantidade de bibliotecas disponíveis que facilitaram a escrita de código e permitiu focar em problemas de mais alto nível. Além disso, a facilidade de uso do *framework* do Arduino tornou mais rápido o processo de prototipagem e desenvolvimento em comparação com o ESP-IDF.

⁸ Disponível em: <<https://platformio.org/>> Acesso em: 03 set, 2023

Embora o tamanho do arquivo binário possa ser maior, a velocidade de desenvolvimento e a disponibilidade de documentação compensou isso.

3.6.2.3 Seleção do *RTOS* Utilizado

A seleção do *RTOS* a ser utilizado foi simples, uma vez que os *chipsets* da Espressif já contam com o *FreeRTOS* portados para esta plataforma. Além disso, esse sistema operacional possui bastante documentação disponível além de fóruns para auxílio no desenvolvimento.

3.6.2.4 Seleção da Plataforma de OTA

Para a escolha da plataforma de gerenciamento de versão de OTA, foram considerados a OTADrive e a AsyncElegantOTA. Ambas são facilmente interligadas com o código do projeto, porém, com a primeira, foi possível utilizar o cliente gerado pela conexão GSM para realizar os *updates* remotos, sendo assim a escolhida para o projeto.

3.6.2.5 Armazenamento Geral de Dados de Estados

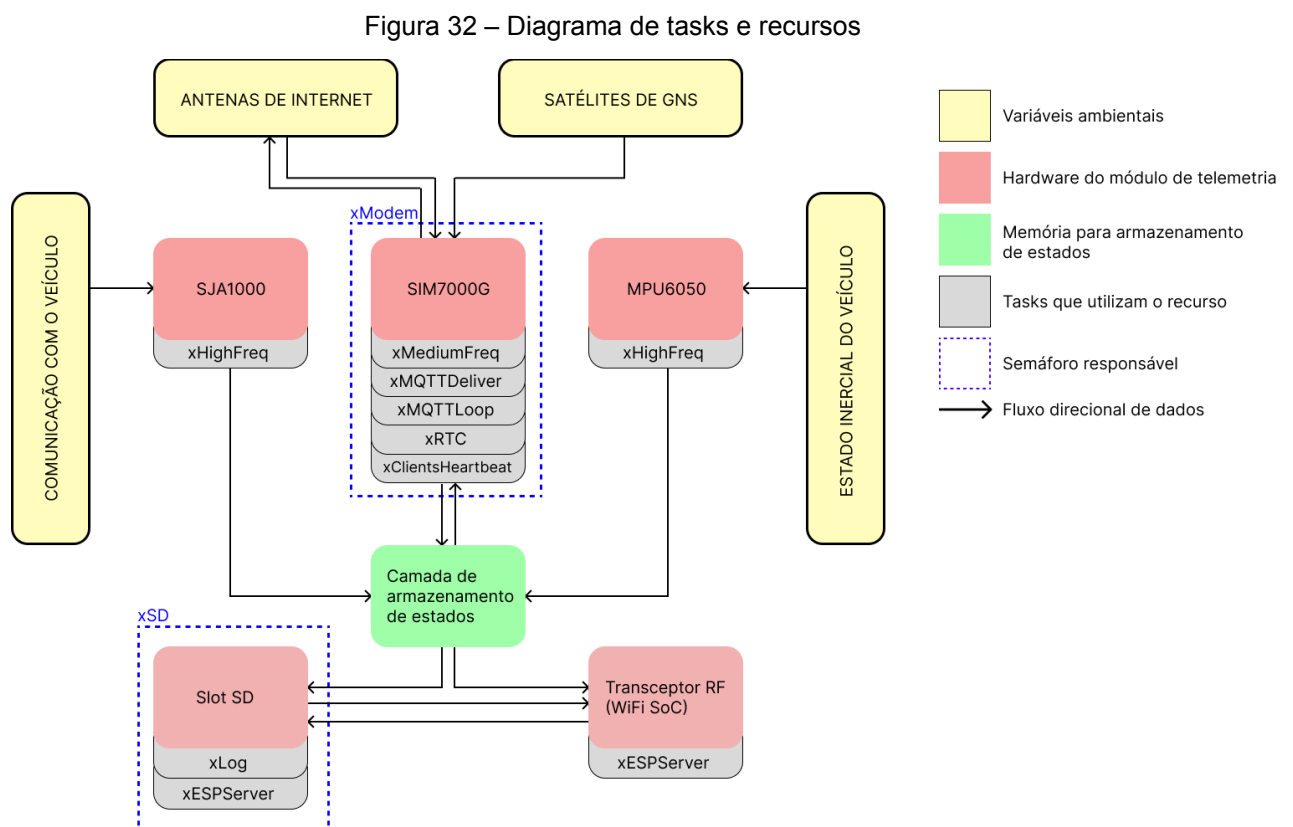
A arquitetura de armazenamento de dados selecionada baseia-se na utilização de classes para armazenar todos os estados capturados pelo dispositivo em atributos de objeto. Essa arquitetura permite o fácil acesso para escrita e leitura de dados mantendo a fácil manutenibilidade do código.

3.6.2.6 Configuração do *FreeRTOS*

Para iniciar o projeto, as tarefas detalhadas na Definição do funcionamento do projeto foram separadas por funcionalidades que tivessem frequências diferentes de execução. Essas funcionalidades foram então traduzidas em tarefas do *FreeRTOS* de forma a facilitar a gestão de cada uma. Neste tópico, os detalhes de cada atividade bem como a gestão de priorização de cada uma serão discutidas.

3.6.2.6.1 Diagrama das Tarefas

O primeiro passo para a configuração correta do kernel foi organizar as tarefas em diagramas que exibissem suas dependências no ponto de vista de *hardware*, Figura 32, sendo possível organizá-las de acordo com qual recurso estaria utilizando. As responsabilidades de cada tarefa podem ser verificadas na Tabela 1.



Fonte: O Autor, 2023

Tabela 1 – *Tasks* e suas descrições divididas por seus *handlers*

Handler	Descrição	Período (ms)
xMediumFreq	Responsável por coletar todos os dados de média frequência. Todas as informações relacionadas ao <i>GPS</i> e <i>GSM</i> .	1000
xHighFreq	Responsável por coletar os dados de alta frequência. Todas as informações referentes aos dados recebidos no barramento <i>CAN</i> e todos os calculados pelo <i>DMP</i> do MPU6050.	500
xLog	Responsável por salvar os últimos dados coletados em arquivo <i>.csv</i> no cartão <i>SD</i> .	1000
xMQTTDeliver	Responsável por enviar os dados coletados para o servidor remoto utilizando o protocolo <i>MQTT</i> .	10
xMQTTLoop	Responsável por verificar a disponibilidade de mensagens nos tópicos <i>MQTT</i> subscritos. Além disso, também envia mensagens periodicamente para o servidor remoto para evitar desconexões por <i>Timeout</i> .	1000
xESPServer	Responsável por verificar se o servidor local da ESP não recebeu nenhuma requisição dentro de um período de 1 minuto. Desliga o servidor caso esteja ocioso.	5000
xRTC	Periodicamente atualiza o <i>RTC</i> interno da ESP através do servidor <i>NTP</i> .	750

xHeartBeat	Responsável por piscar o <i>LED</i> integrado na placa ESP, indicando o correto funcionamento do software.	50
xClientsHeartbeat	Responsável por observar se os dados coletados pela placa estão sendo utilizados. Desliga o serviço de coleta de dados caso esteja ocioso.	5000

Fonte: O Autor, 2023.

Essa análise durante a configuração do *RTOS* possibilitou a identificação inicial de quais atividades iriam compartilhar recursos de *hardware*, possibilitando então o desenvolvimento de um *handler* que gerenciasse o acesso a essas entidades.

3.6.2.6.2 Semáforos e Gestão dos Recursos Compartilhados

Neste projeto, alguns módulos são utilizados para mais de uma funcionalidade, dessa forma, é necessário aplicar ferramentas que garantam a exclusão mútua para utilização desses componentes. Para isso, foram utilizadas as ferramentas disponibilizadas pelo *kernel FreeRTOS*, como o semáforo. Os semáforos criados e gerenciados em tempo de execução estão detalhados na Tabela 2.

Tabela 2 – Semáforos e suas descrições

Semáforo	Descrição	Recurso
xModem	Semáforo para assegurar a correta comunicação entre a ESP32 e o <i>Modem</i> . Isso se faz necessário pois esse componente comunica-se através de comandos <i>AT</i> para enviar os dados via <i>MQTT</i> , adquirir os dados de <i>GPS</i> e atualizar o <i>RTC</i> interno da ESP.	SIM7000G
xSD	Semáforo para organizar qual atividade tem prioridade na comunicação <i>SPI</i> com o módulo <i>SD</i> . Esse componente é responsável por registrar os dados aqusitados além de disponibilizá-los através do servidor ESP.	Cartão <i>TF</i>

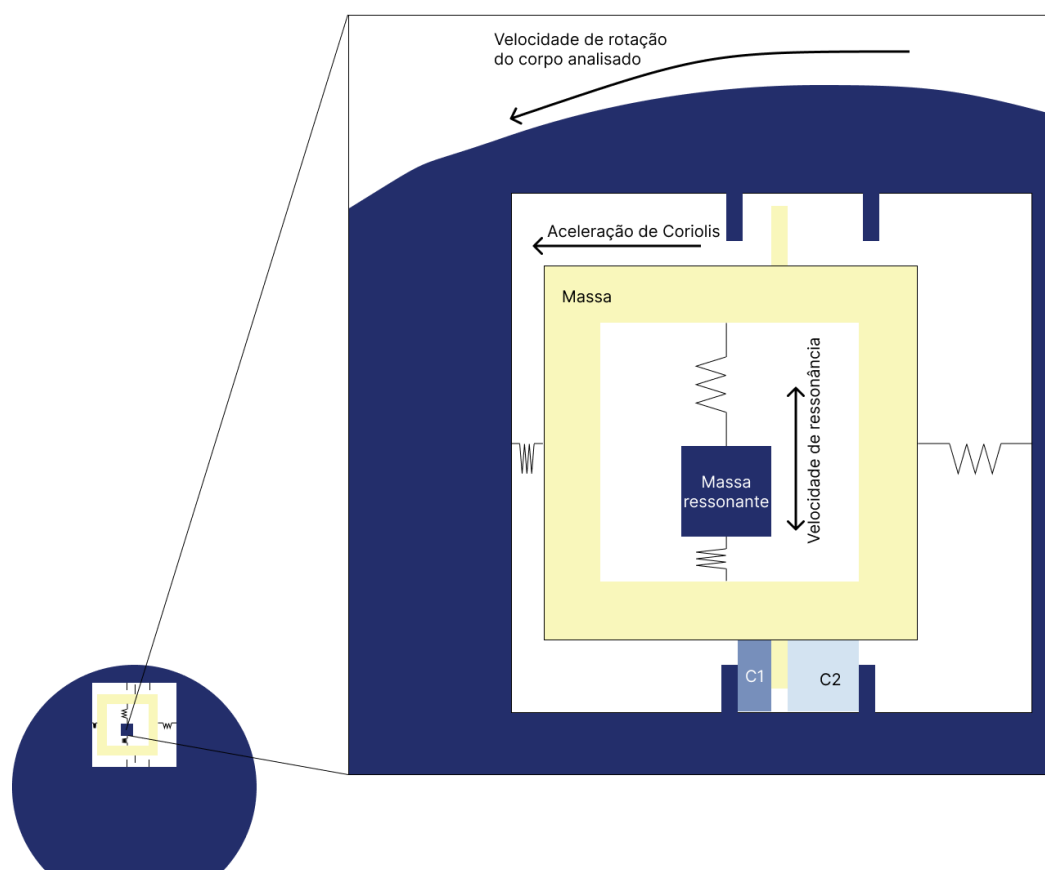
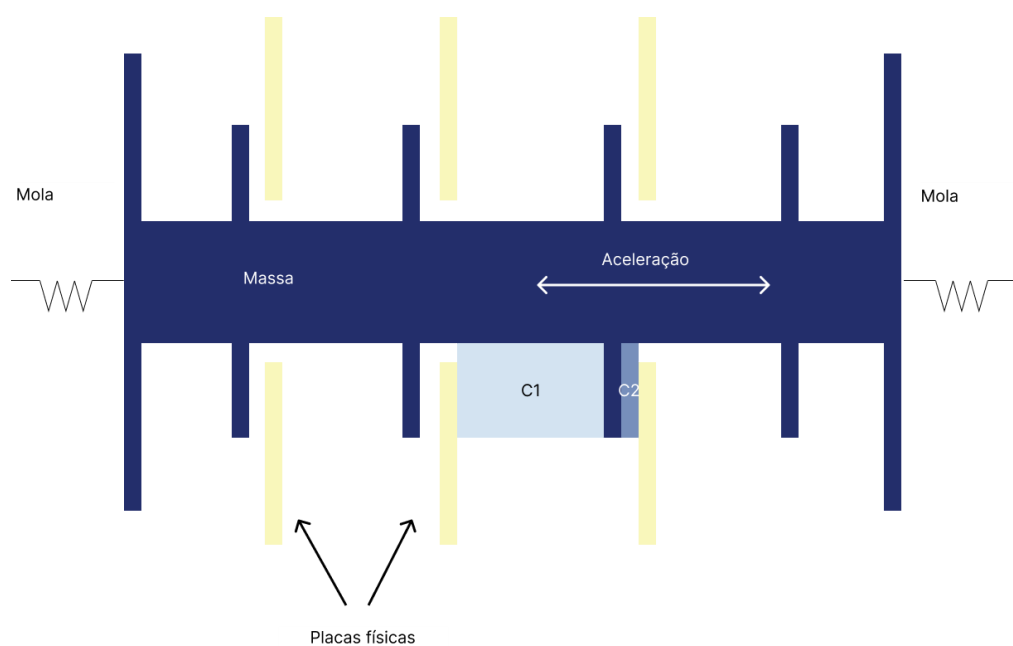
Fonte: O Autor, 2023.

O gerenciamento de exclusão mútua se mostra bastante eficiente no cenário estudado, pois os *hardwares* compartilhados entre várias tarefas possuem uma comunicação síncrona bilateral com o microcontrolador. Essa característica, exige que a comunicação entre o microcontrolador e o periférico dentro de uma tarefa seja finalizada antes da liberação desse recurso para outra.

3.6.2.7 Integração dos Componentes no *Firmware*

Nesta seção, é trabalhado a integração do microcontrolador ESP32 com os módulos eletrônicos adjacentes. Para cada periférico, é descrito a forma de comunicação, as bibliotecas externas utilizadas e o processamento dos dados. A conexão dos periféricos foi realizada conforme a Figura 33.

Figura 34 – Diagrama montagem do mecanismo de medição. A) acelerômetro B) giroscópio



Fonte: O Autor, 2023

O processo básico de extração de dados inicia com a calibração da orientação do MPU com base na orientação do objeto a ser estudado. Para isso, uma série de registros são somados e uma média é tomada para estabelecer um valor dentro de uma certa tolerância. Essas médias são usadas em software para alinhar o eixo do sensor com o eixo do veículo instalado.

Em seguida, os dados mensurados são lidos através dos registradores do sensor. Para a aceleração, os valores para cada eixo inercial são subtraídos das médias calculadas anteriormente. Para a orientação, a velocidade angular registrada pelo sensor é integrada no tempo para obter a orientação absoluta. Além disso, os dados relacionados à orientação são refinados através de um filtro complementar dos valores calculados pelo acelerômetro para remoção de desvio acumulativo.

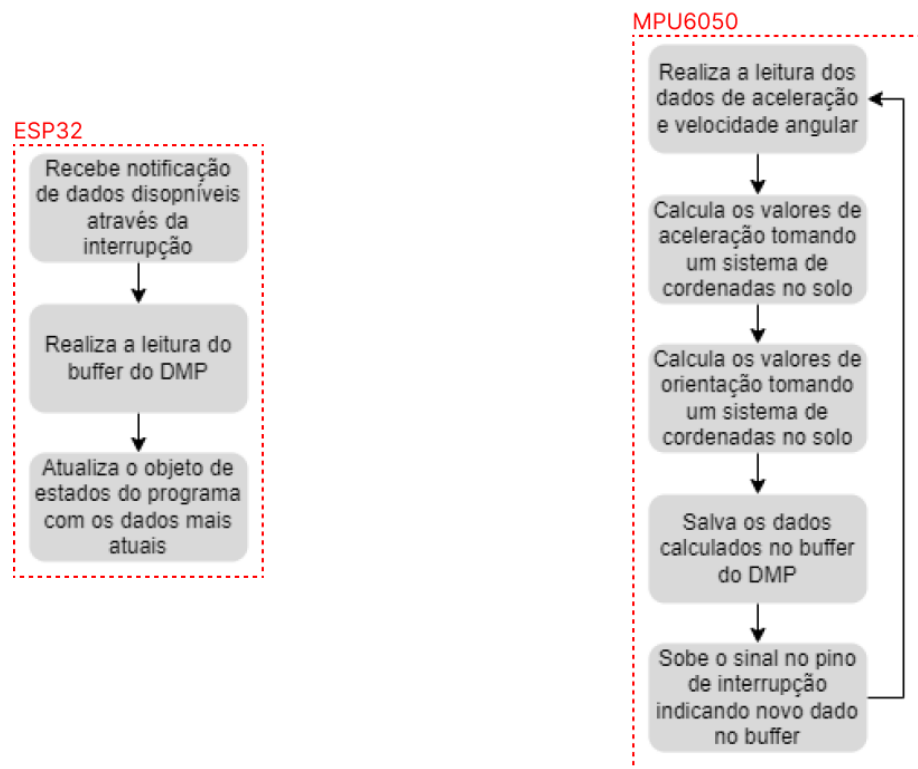
Todo esse processo foi abstraído pelo *DMP* sendo então somente necessário ler os dados já calculados e presentes no seu *buffer*. Dessa forma, somente é necessário ler a orientação dos vetores referentes a orientação da moto e a orientação da gravidade.

Por fim, essas informações são salvas na classe de armazenamento geral dos estados do dispositivo. Nessa estrutura, são gravadas informações relacionadas a orientação, *pitch* e *roll*, não sendo calculados os dados de *yaw*, devido à ausência de um magnetômetro no *chip*. Além disso, os dados de aceleração lateral e longitudinal são salvos para futuras análises de performance de pneu e *powertrain*.

A interface física entre esse dispositivo e o MCU se dá através do protocolo *I²C*, sendo utilizado um pino de interrupção para que o MPU consiga sinalizar ao controlador a disponibilidade de dados calculados com sucesso presentes no *buffer* do *DMP*.

O fluxo de dados dessa etapa é exibido no diagrama de blocos a seguir, Figura 35.

Figura 35 – Diagrama atividades de interação entre ESP32 e MPU6050

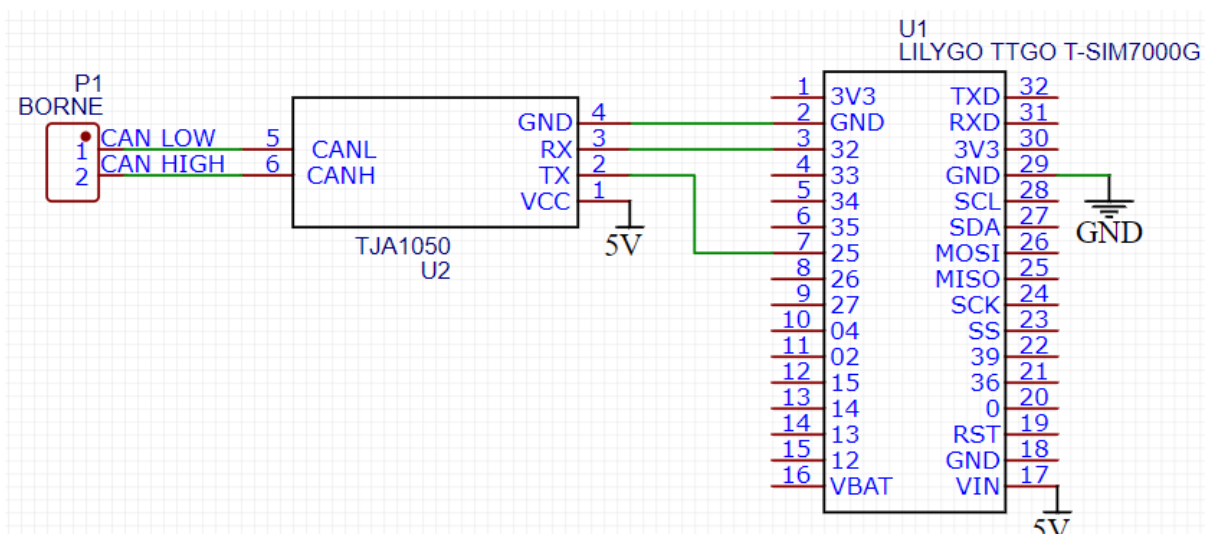


Fonte: O Autor, 2023

3.6.2.7.2 TJA1050

O sistema de comunicação externa desse dispositivo iniciou com a definição da interface entre o transceiver e o *MCU* através da conexão *UART*. Essa conexão é detalhada na Figura 36.

Figura 36 – Diagrama conexão entre a o TJA1050 e a ESP32



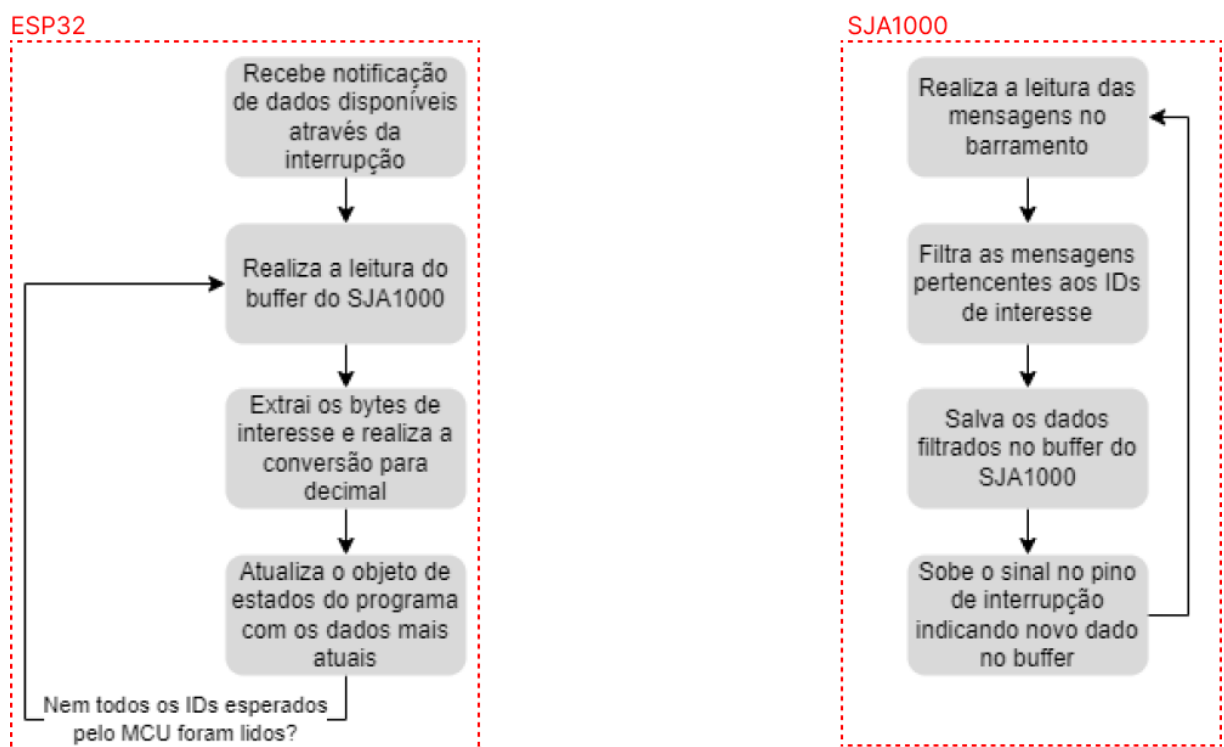
Fonte: O Autor, 2023

Em seguida, foi necessário configurar os filtros de hardware necessários para evitar que o *MCU* gastasse tempo de máquina para processar todas as informações vindas da *CAN*. O módulo controlador da *CAN* utiliza esses filtros para somente adicionar ao buffer de dados acessados pelo *MCU*, as informações de IDs de interesse para o projeto. Isso é necessário pois o barramento no veículo pode receber um volume de até 250 Kbps, e somente é necessário processar 6 dessas mensagens de 8 bytes transmitidas em períodos de 50 ms (96 bytes por segundo).

É checado a presença de dados no *buffer* pelo *MCU* em um período de 500ms e em cada tentativa de leitura, é esperado alguns IDs específicos. A tentativa de leitura termina se acontecer um dos dois eventos: todos os IDs forem lidos com sucesso ou o tempo de tentativa expire.

Com a leitura bem sucedida, os dados são convertidos dos valores hexadecimais transmitidos no barramento para valores de mais fácil leitura. Por fim, esses dados são salvos na classe de armazenamento. O fluxo de leitura é detalhado na Figura 37.

Figura 37 – Diagrama atividades exercidas pelo SJA1000

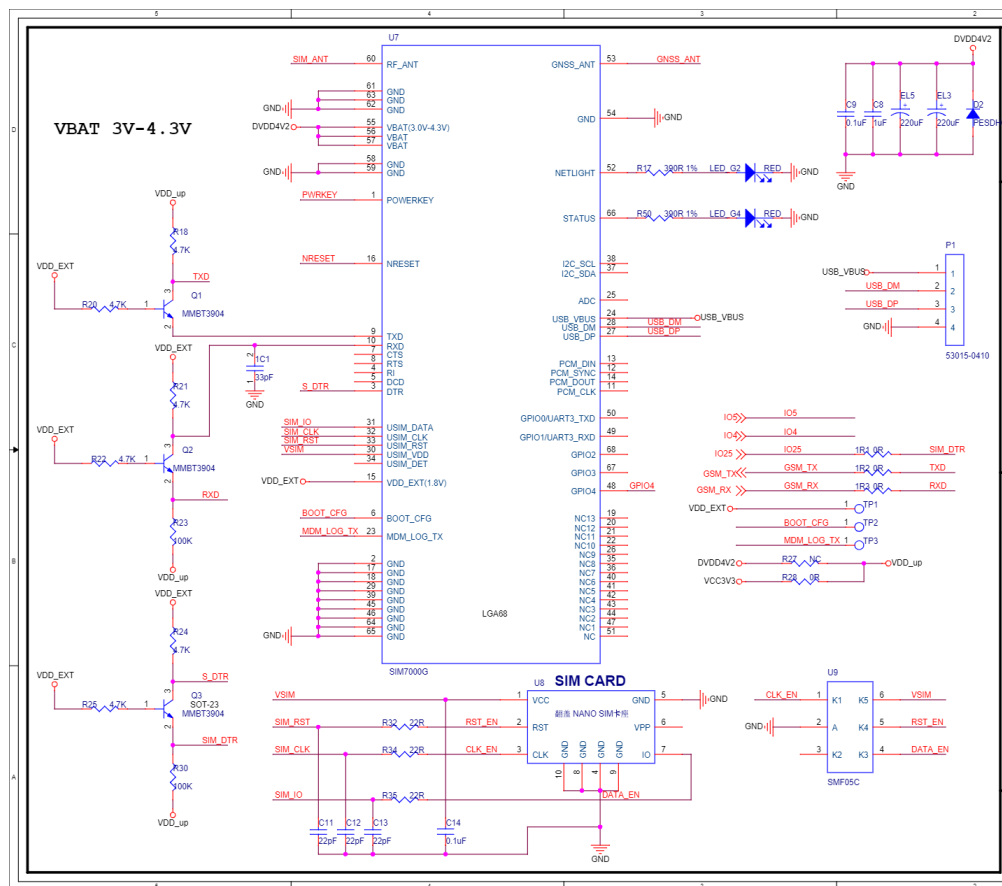


Fonte: O Autor, 2023

3.6.2.7.3 SIM 7000G

Para a placa de desenvolvimento utilizada, o modem já está integrado a *PCB*. A interface desse dispositivo ao *MCU* se dá através de comandos *AT* utilizando um barramento *UART*. Além disso, esse dispositivo foi conectado a uma antena *full band LTE* e uma antena cerâmica ativa de *GNSS*. Como esse dispositivo desempenha múltiplas funções no fluxo do aplicativo, o acesso aos seus recursos é limitado por semáforos do *FreeRTOS*. O diagrama de atividades desempenhadas por esse dispositivo pode ser visto na Figura 38.

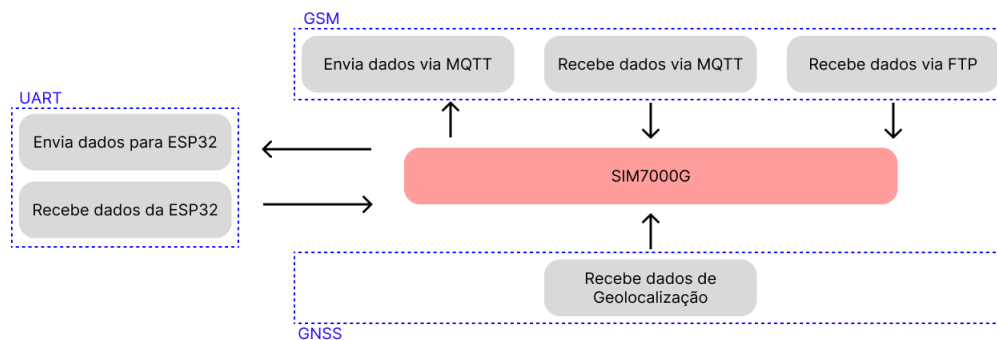
Figura 38– Interface eletrônica SIM7000G



Fonte: www.github.com/Xinyuan-LilyGO

A interface de comunicação via comandos *AT* é encapsulada através da biblioteca TinyGSM sendo possível abstrair a criação de um cliente *TCP*, a aquisição de dados de *GNSS* e o envio de mensagens a um servidor remoto. Esses fluxos podem ser vistos através da Figura 39.

Figura 39 – Diagrama atividades exercidas pelo SIM7000G

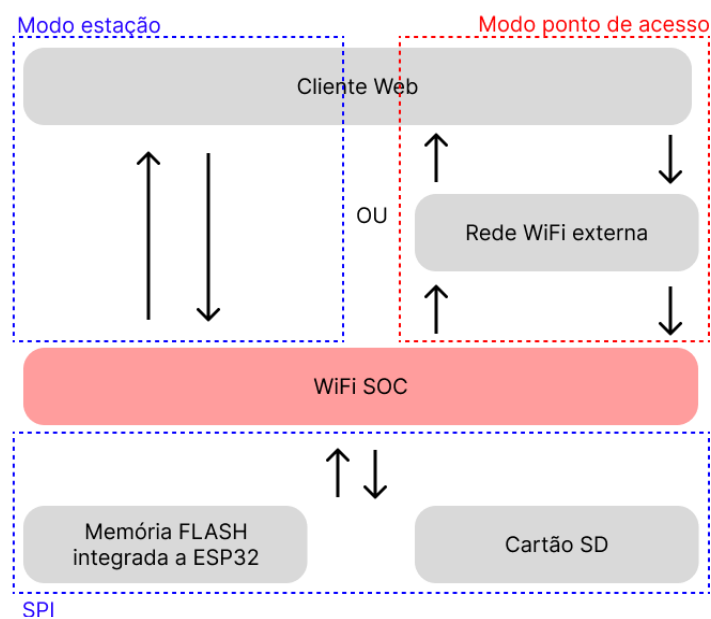


Fonte: O Autor, 2023

3.6.2.7.4 Módulo WIFI SOC (System-On-Chips)

A comunicação *WiFi* da placa é gerenciada de forma a permitir a conexão direta com o controlador através de um servidor *web*. Para isso, é utilizado a memória *flash* interna ao *MCU* para salvar as páginas *html* que serão servidas através de uma comunicação *HTTP*. Além disso, os logs de dados realizados pelo *MCU* e salvos no cartão *TF*, são consultados para serem disponibilizados nessa interface. O diagrama de acesso de dados e tarefas executadas por esse módulo pode ser consultado na Figura 40.

Figura 40 – Diagrama atividades exercidas pelo WiFi SOC



Fonte: O Autor, 2023

3.7 SOFTWARE DE MONITORAMENTO *WEB*

O desenvolvimento da aplicação responsável pela exibição dos dados enviados pelo módulo de telemetria é descrito neste tópico. Nesse sentido, serão abordadas quais as funcionalidades projetadas para essa etapa do projeto, quais as ferramentas utilizadas para o desenvolvimento e o *design* atribuído às páginas de acesso.

3.7.1 Funcionalidades Esperadas

Como o objetivo desta ferramenta é o acompanhamento em tempo real dos parâmetros e estados do veículo durante testes de rodagem, essa ferramenta teve como funcionalidades as seguintes:

- A. Exibição de quais motos estão *online* em tempo real
- B. Acompanhamento de um veículo em tempo real
 - a. Estado Inercial
 - b. *Powertrain* (motor e inversor *BLDC*)
 - c. Baterias (*BMS*)
 - d. Geolocalização (*GPS* e/ou *GLONASS*)
 - e. Rede de móvel (*GSM*)
 - f. Alarmes e erros nos componentes
- C. *Datalogger*
- D. Acesso à memória de um equipamento
 - a. Alterar as configurações internas do equipamento
 - b. Baixar os dados salvos
 - c. Deletar os dados salvos
 - d. Modificar os parâmetros salvos

3.7.2 Desenvolvimento do Software Web

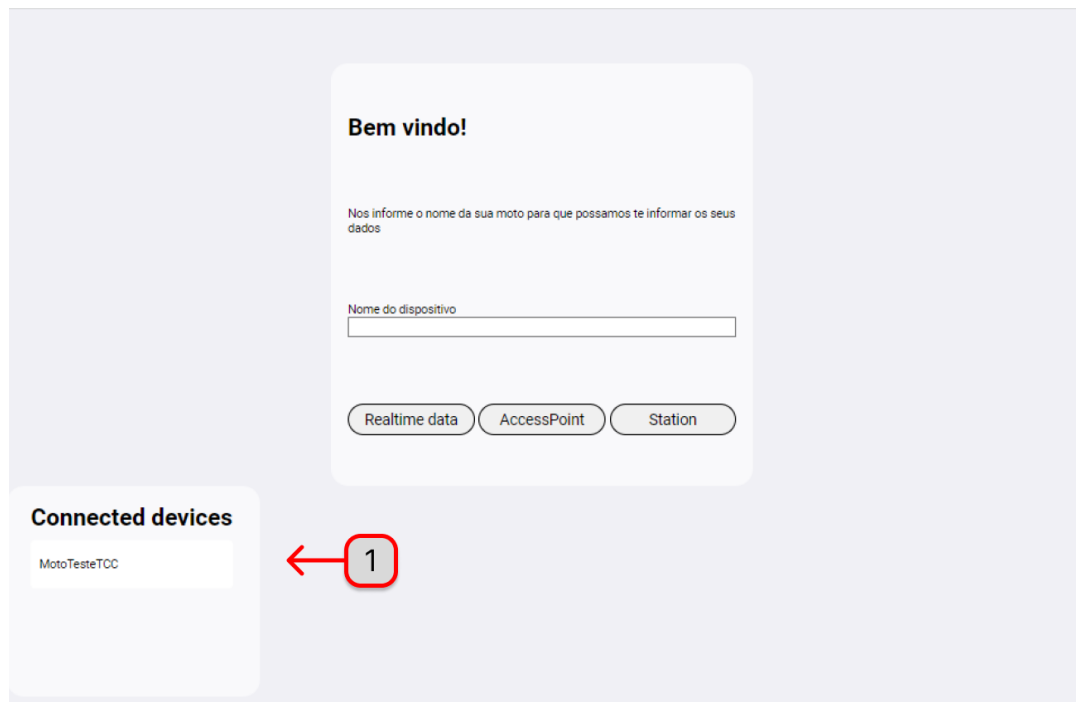
Para permitir um rápido desenvolvimento da plataforma de visualização, foram consideradas 2 ferramentas de produção de software *frontend*: Grafana e React.js. Ambas ferramentas são utilizadas no mercado para a finalidade similar a deste trabalho. Para este projeto, foi escolhido o *framework* React.js devido ao grande ecossistema ao redor dessa ferramenta, além de permitir uma maior ajuste dos componentes utilizados de acordo com a necessidade do trabalho.

A produção do código foi realizada de forma compartimentada, sendo possível reutilizar um mesmo componente em diversos contextos. Esse é o caso dos gráficos de *gauge* utilizados para exibir dados como, velocidade, torque, corrente e eficiências.

3.7.3 Design

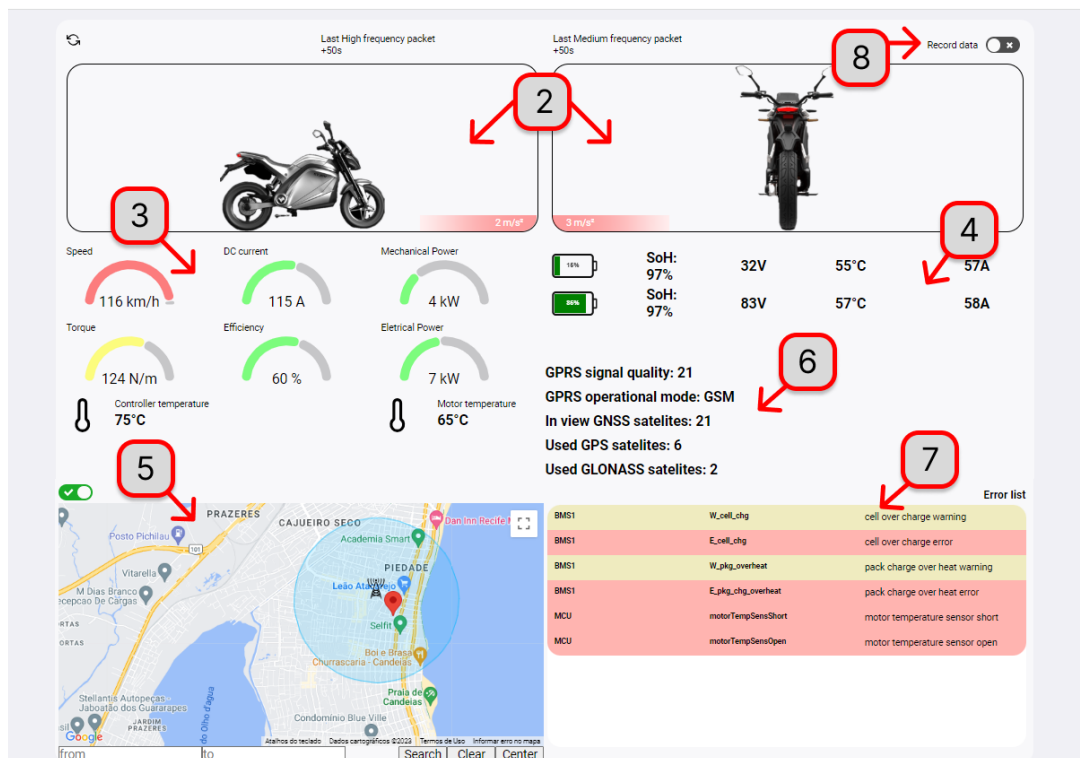
O design das páginas de visualização foram realizadas de forma minimalista, tendo como objetivo a fácil e rápida utilização. Para isso, poucos botões e opções de customização foram utilizados. O design final das páginas bem como a verificação do cumprimento das funcionalidades pode ser visualizado na Figura 41, Figura 42, Figura 43 e explicados na Tabela 3 .

Figura 41 – Interface de login e visualização de dispositivos *online*



Fonte: O Autor, 2023

Figura 42 – Interface de visualização dos estados da moto em tempo real



Fonte: O Autor, 2023

Figura 43 – Interface de configuração do dispositivo

whistleblower

APN Host
claro.com.br

APN User
claro

APN Password
claro

MQTT Host
broker.hivemq.com

MQTT User

MQTT Password

MQTT Port
1883

MQTT ID
MotoTesteTCC

MQTT Project
Whistleblower

WiFi STA SSID
Rede_WiFi

WiFi STA Password
123123

WiFi AP SSID
TCU_TCC_Mateus

WiFi AP Password
tcc_mateus

Read Submit

Datalogger

Log File	Size	Action
log_file_17_10_00_06_08_2023.txt	1882	[Delete]
log_file_17_23_15_06_08_2023.txt	2096	[Delete]
log_file_18_05_00_06_08_2023.txt	2096	[Delete]

Fonte: O Autor, 2023

Tabela 3 – Dimensões e montagem

Funcionalidade	Identificação
motos estão online em tempo real	1
Estado Inercial	2
Powertrain	3
Baterias	4
Geolocalização	5
Rede de móvel	6
Alarmes e erros nos componentes	7
Datalogger	8
Alterar as configurações internas do equipamento	9
Baixar os dados salvos	10
Deletar os dados salvos	11

Fonte: O Autor, 2023

3.8 CAIXA PLÁSTICA DE PROTEÇÃO

Nesta seção, são discutidas as etapas de desenvolvimento da proteção de plástico que protege todo o circuito eletrônico. Esse componente possui a função de proteger a *PCB* contra impactos, vibrações e umidade, além de oferecer uma aparência agradável ao projeto. Foi importante levar em consideração fatores como o tamanho e a forma do circuito, a resistência do material, a facilidade de montagem e desmontagem, e a acessibilidade a componentes e conexões. É abordado o processo de *design* e fabricação do case, incluindo a modelagem 3D, a seleção de materiais e a montagem final do produto.

3.8.1 Software de Desenvolvimento CAD

Para o desenvolvimento do case, foi necessário utilizar ferramentas virtuais que permitissem o desenvolvimento do projeto com as tolerâncias dimensionais adequadas. Para isso foi utilizado o ⁹OnShape para cumprir com esse propósito.

O OnShape é uma plataforma de CAD baseada em nuvem que permite o desenvolvimento de projetos de modelagem 3D de forma colaborativa. Os principais fatores que guiaram a escolha dessa ferramenta em detrimento às demais tradicionais no mercado, foi a disponibilidade de acesso via *Web*, sem necessidade de *hardwares* dedicados para se obter um bom desenvolvimento. Além disso, a ferramenta de versionamento possibilita a fácil experimentação de diversas versões antes da definição do produto final.

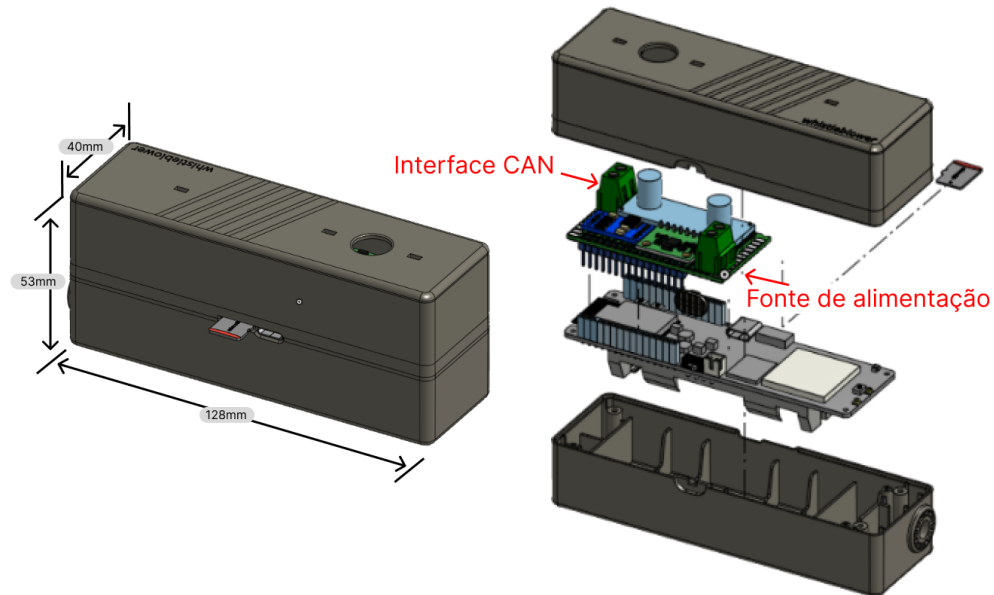
3.8.2 Compartimento Principal

O *design* do compartimento principal baseou-se no modelo já disponibilizado pela fabricante. A partir desse modelo base, foram realizadas as modificações com relação a altura do compartimento, de forma a encaixar os componentes eletrônicos adicionais. O resultado final pode ser verificado através da Figura 44.

⁹ Disponível em:

<<https://cad.onshape.com/documents/b0dd05d124ece4ecf311db5e/w/90852089983174c081e1eab1/e/7486465ff2abd4cc2fab7364?renderMode=0&uiState=64f4108dc4b95858722cda1d>> Acesso em: 03 set, 2023

Figura 44 – Dimensões e montagem



Fonte: O Autor, 2023

3.6.3 Fabricação e material

Para a fabricação deste protótipo, foi utilizado o método da impressão 3D, por ser uma forma rápida, barata e fácil de prototipação de produtos com uma geometria complexa, como a encontrada na caixa estudada. Esses atributos permitem o rápido desenvolvimento e correção de quaisquer erros de geometria que não foram trabalhados durante o modelamento tridimensional. Em um cenário de produção em massa, esse processo deve ser mudado para o método de injeção plástica, devido a rápida manufatura e boa qualidade no produto final, apesar de possuir um custo maior para início de produção, devido ao custo de moldes.

O material selecionado para o projeto foi o PLA, sobretudo devido a sua facilidade de utilização, Figura 45, importante para futuras modificações de geometria utilizando ferramentas manuais. Além disso, esse é um material com ampla disponibilidade no mercado, podendo-se baratear os custos devido ao alto poder de barganha que essa característica traz.

Figura 45 – Comparativo de características mecânicas entre materiais de impressão 3D

	PLA	ABS	HIPS	PETG
Strenght	★★	★★★★	★★★★★	★★
Hardness	★★★	★★★★★	★★★★★	★★★★
Durability	★★	★★★★★	★★★★★	★★★★
Moisture Sensitive	★	★	★	★
Soluble	★	★★★★	★★★★	★
Flexibililty	★	★	★	★
Easy to 3D- print	★★★★	★★	★★	★★★★
Warping Sensitive	★	★★★★	★★★★	★

Fonte: Adaptado de www.3devo.com/, 2023

4 RESULTADOS

Esta seção é reservada para a discussão dos testes, resultados, validação do protótipo e os próximos passos para evolução do sistema.

4.1 TESTES JUNTO À PLATAFORMA DE VALIDAÇÃO

Esta etapa se trata de um teste estático das funcionalidades utilizando a plataforma desenvolvida para esta finalidade. Foi realizado um teste de verificação de funcionalidade, Tabela 4, para assegurar . Com esses testes, foi possível validar as funcionalidades estabelecidas nos objetivos específicos e qualificá-lo para os testes em campo.

Tabela 4 – Cumprimento dos objetivos específicos

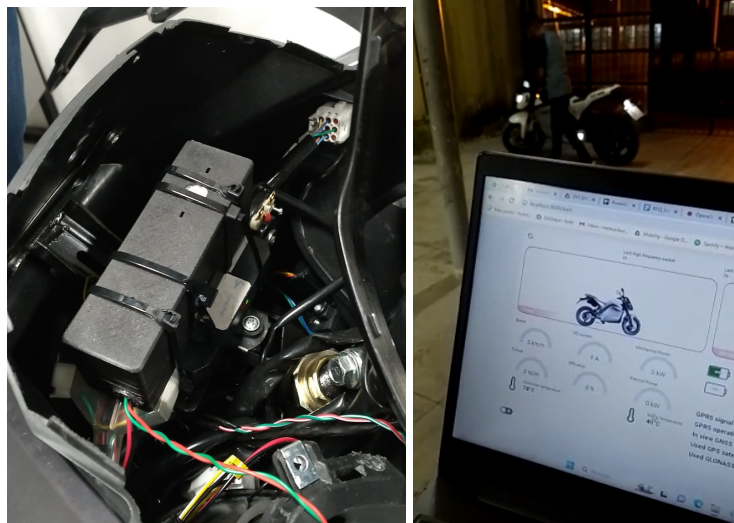
Funcionalidade	Implementada	Descrição
Método de comunicação eficiente entre sensores veiculares e módulo desenvolvido	Sim	Utilização do protocolo <i>CAN</i> para comunicação com outros módulos
Mensurar orientação veicular	Sim	Utilizado um módulo Acelerômetro/giroscópio integrado ao projeto
Método de comunicação eficiente entre o módulo desenvolvido e um servidor remoto	Sim	Utilizado o protocolo <i>MQTT</i> para comunicação em tempo real com um servidor remoto
Armazenamento dos dados em tempo real	Sim	Utilização de um leitor de cartões <i>TF</i> para salvar os dados adquiridos pelo projeto
Comunicação <i>peer-to-peer</i>	Sim	Utilizado módulo <i>WiFi</i> da <i>ESP32</i> para comunicação direta com a placa
Atualização remota do firmware do dispositivo	Sim	Implementação do método de atualização <i>OTA</i> para dispositivos remotos
Mínimos níveis de consumo energético e de dados	Não	Não foram implementadas rotinas de hibernação dos dispositivos devido ao curto cronograma

Fonte: O Autor, 2023.

4.2 TESTES EM CAMPO

Para validação em campo deste dispositivo, o equipamento foi montado em uma moto, Figura 46, e percorreu um percurso padrão urbano. Durante o teste, foi acompanhado os dados em tempo real e foi enviado o comando para iniciar o *log* de dados. Ao todo, foram realizadas 5 horas de teste contínuo em campo, sendo suficiente para validação do protótipo.

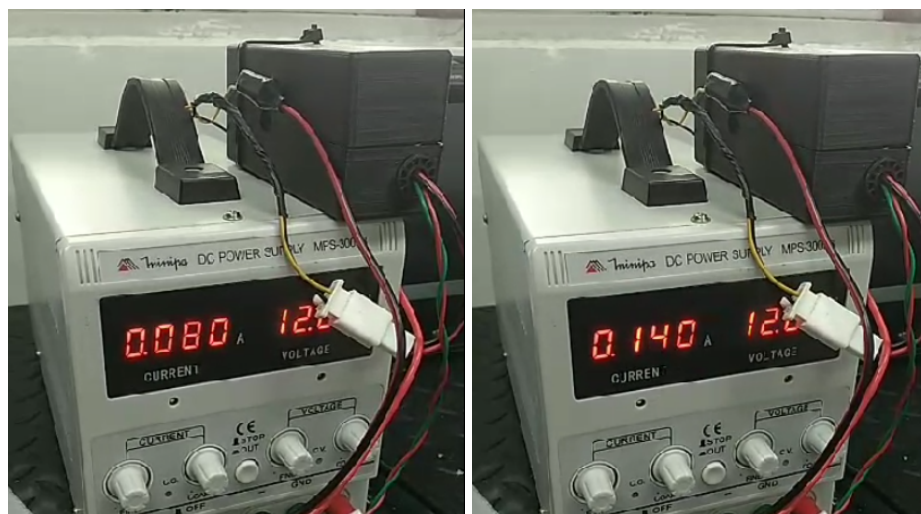
Figura 46 – Testes em campo com o veículo



Fonte: O Autor, 2023

Como o dispositivo se encontra instalado em uma infraestrutura elétrica compartilhada com diversos outros equipamentos que competem para a utilização da bateria do veículo, foi necessário avaliar a potência que este item consome. Para isso, o equipamento foi montado em bancada e seu consumo foi avaliado utilizando uma fonte de bancada, Figura 37. O resultado mostrou um pico de consumo de 1.68W (0.140A) em picos de 0.28 segundos, 0.96W (0.08A) no estado ocioso e 1.23W (0.103A) RMS. Esses resultados estão de acordo com modelos comerciais de equipamentos similares.

Figura 47 – Testes de consumo energético



Fonte: O Autor, 2023

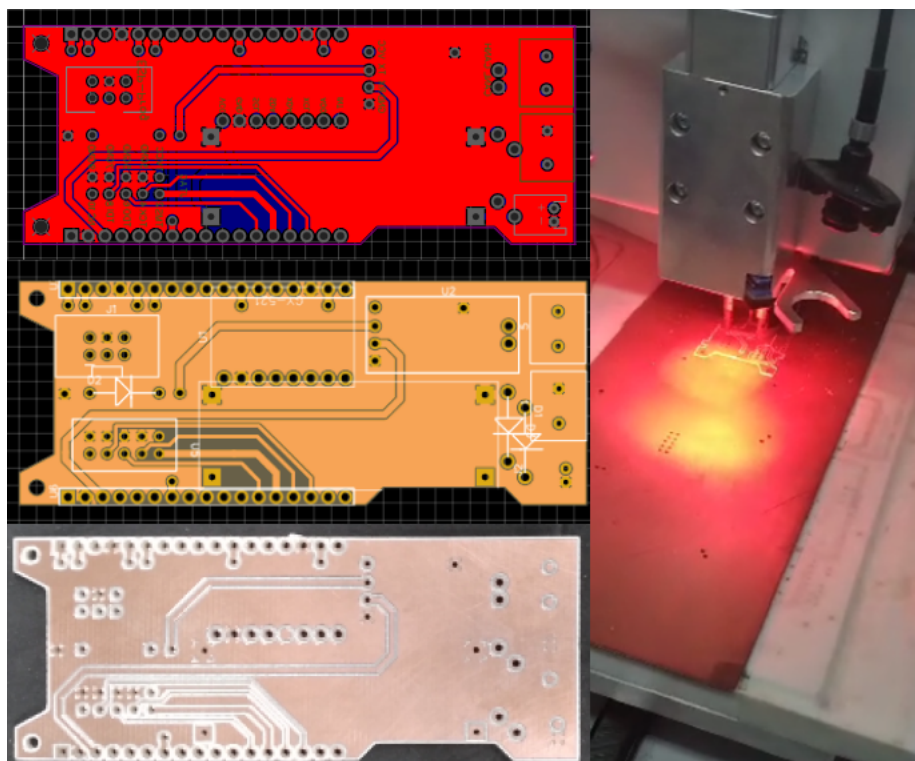
5 PRÓXIMOS PASSOS

Durante a produção deste protótipo, foram observadas algumas oportunidades de melhoria que ainda não foram implementadas no escopo deste trabalho. Neste tópico, serão descritos quais são e os desafios que as descontinuarão durante a elaboração do código.

5.1 Placa de Circuito Impresso

Como explicado anteriormente, foi utilizado uma placa perfurada e pontes de estanho para unir os componentes à placa principal ESP32. Essa decisão se mostrou útil durante este trabalho, pois permitiu rápidos ajustes que foram necessários para garantir o bom aproveitamento do espaço disponível e remoção de itens que não seriam utilizados mais. No entanto, esta placa de fibra de vidro não é a ideal para continuar com futuros desenvolvimentos, sendo necessário o desenvolvimento de uma *PCB* que consiga encapsular todos os componentes utilizados e um circuito mais compacto e robusto, Figura 48.

Figura 48 – Protótipo não finalizado de uma PCB



Fonte: O Autor, 2023

5.2 Ajustes Automatizados do MPU6050

Para o bom funcionamento da unidade de medição inercial, esta precisa ser ajustada para cada veículo e situação que ela venha a ser implementada. Neste trabalho somente um veículo foi alvo de estudo, sendo possível somente realizar a calibração manualmente e escrevendo os resultados diretamente no *firmware* da aplicação. Esta situação impede a escalabilidade deste projeto para outros veículos. Uma solução que deve ser trabalhada futuramente é a implementação de uma funcionalidade de calibração na interface *web*, sendo possível o ajuste para cada tipo de veículo sem a necessidade de gerar novos *firmwares*.

5.3 Separação do *Hardware* de *GNSS* e *GSM*

Como explicado durante o texto, esse projeto utilizou uma placa de desenvolvimento que incorporava um modem capaz de adquirir dados de *GNSS* e *GSM*. Embora este tenha sido o suficiente para o completo funcionamento das funções planejadas, a comunicação com esse item se mostrou bastante ineficiente. Este problema deriva do fato de que os dados de *GNSS* são valores que demoram até 1 segundo para serem adquiridos pelo módulo e, durante o tempo de aquisição, o modem é bloqueado para qualquer outra atividade, relacionada ou não com a conexão com a internet. Sendo assim, este trabalho ainda possui a oportunidade da separação do *hardware* responsável pelo *GNSS* dos demais, sendo possível comunicar com esse item de forma assíncrona e mais eficiente.

6 CONCLUSÃO

Com todas essas informações geradas durante este projeto, pôde-se entender o potencial que ferramentas IoT estão tomando na tecnologia atualmente. Com poucos recursos de hardware, foi possível criar uma ferramenta que possui alto valor agregado para engenheiros automotivos na medida que este viabiliza testes controlados e acompanhados a cada segundo.

O desenvolvimento deste item iniciou com a definição das funcionalidades que o equipamento iria assumir e as atividades que ele deveria realizar de forma a garantir o funcionamento. Com isso, foi possível estabelecer a proposta de valor do equipamento.

Com o funcionamento definido, foram selecionados os componentes eletrônicos que iriam compor o sistema, dado suas vantagens referentes a itens similares. Nesta etapa, foram selecionados componentes responsáveis pela alimentação energética, sensores inerciais, transceptor de dados, central de processamento e modem de comunicação em rede.

Esses itens foram, em seguida, integrados através da fabricação de uma PCB. Nesta etapa, foram prototipados placas perfuradas de fibra de vidro que integravam os componentes eletrônicos através de pontes de estanho.

Para viabilizar as etapas seguintes, foi necessário fabricar um módulo eletrônico de testes que emulasse o comportamento do veículo alvo. Assim, foi utilizado um Arduino Mega e um controlador de CAN MCP2515 para criar uma bancada que pudesse enviar todos os dados que o veículo alvo é capaz de mandar.

Com a plataforma de validação pronta, foi possível continuar o trabalho ao desenvolver a interface que exibe as informações enviadas pelo dispositivo eletrônico. Para isso, foi utilizado o *Framework* React.js para a criação das janelas e comunicação com o servidor de dados.

Com a lógica pronta, a etapa de desenvolvimento foi finalizada com a modelagem do invólucro em software CAD. O produto dessa etapa pode ser enviado para uma máquina de impressão 3D para fabricar o protótipo.

Por fim, o protótipo passou por testes em campo, de forma a validar todas as funcionalidades desenvolvidas em um ambiente que simula situações reais. Além disso, o consumo energético do dispositivo foi mensurado utilizando uma fonte de bancada.

REFERÊNCIAS

- Fierce Electronics. Recent Chip Scare Tempered IDC's Long-Term Growth Outlook. Fierce Electronics. Disponível em: <https://www.fiercееlectronics.com/sensors/recent-chip-scare-tempered-idcs-long-term-growth-outlook#:~:text=A%20recent%20IDC%20forecast%20shows,from%202022%2C%20according%20to%20IDC>. Acesso em: 30 de junho de 2023.
- Fortune Business Insights. Telemetry Market. Fortune Business Insights. Disponível em: <https://www.fortunebusinessinsights.com/telemetry-market-102598>. Acesso em: 30 de junho de 2023.
- Recursos de software exclusivos ajudam os fabricantes de automóveis a superar o desempenho dos concorrentes. Infor. Disponível em: https://dam.infor.com/api/public/content/782c517274834c54a78942aa358c6e76?v=5cc8141d&_ga=2.243496397.971195524.1694440002-958391742.1694440002&_gac=1.48682324.1694440021.Cj0KCQjw9fqBhDSARIsAHlcQYS6-4WXN08Q8Fd576kQpINunjRehnVjuUFHk28Z7ASHfZss7lev4JAaAhoMEALw_wcB. Acesso em: 30 de junho de 2023.
- Página de Suporte do Safety Score. Tesla. Disponível em: [\[https://www.tesla.com/support/safety-score#version-2.0\]](https://www.tesla.com/support/safety-score#version-2.0). Acesso em: 30 de junho de 2023.
- SOARES, F. et al. Exploratory study on the effects of passive solar design strategies on the energy performance of school buildings in a temperate climate. *Energy Procedia*, [S.l.], v. 158, p. 2959-2964, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352152X19309429>. Acesso em: 30 jun. 2023.
- CARRARA, W. C. Sistema de monitoramento e controle de nível de tanque de água utilizando Arduino. 2019. 92 f. Monografia (Graduação em Engenharia Elétrica) - Anima Educação, Faculdades Integradas de Bauru, Bauru, 2019. Disponível em: <https://repositorio.animaeducacao.com.br/handle/ANIMA/4202>. Acesso em: 30 jun. 2023.
- MORAES, R. et al. Assessing the Impact of Electric Vehicle Penetration on Distribution Networks Considering User Daily Mobility. In: *INTERNATIONAL UNIVERSITIES POWER ENGINEERING CONFERENCE (UPEC)*, 2019, Bucharest. Disponível em: <https://repositori.upf.edu/handle/10230/52903>. Acesso em: 30 jun. 2023.
- SANTOS, R. et al. An Integrated IoT Framework for Energy Management in Smart Buildings. *Sensors*, [S.l.], v. 17, n. 3, p. 582, 2017. Disponível em: <https://www.mdpi.com/1424-8220/17/3/582>. Acesso em: 30 jun. 2023.
- BHARDWAJ, R. et al. Optimal placement of phasor measurement units for state estimation in electric power systems. In: *IEEE POWER AND ENERGY SOCIETY*

GENERAL MEETING (PESGM), 2018, Portland. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8654628>. Acesso em: 30 jun. 2023.

OJO, J. et al. An Improved Load Balancing Strategy for Large-Scale IoT Applications. *Energies*, [S.l.], v. 11, n. 1, p. 120, 2018. Disponível em: <https://www.mdpi.com/1996-1073/11/1/120>. Acesso em: 30 jun. 2023.

GONZALEZ, M. et al. Dynamic Tariff and Storage Allocation in Multi-User Smart Grids: An Approach Based on Joint Time and Price Differentiation. In: *IEEE TRANSMISSION AND DISTRIBUTION CONFERENCE AND EXPOSITION (T&D)*, 2020, Chicago. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10072669>. Acesso em: 30 jun. 2023.

HINTON, E. N.; WESTON, P. Baroclinic and Barotropic Instabilities on a Rotating Sphere. In: *AIAA AVIATION 2020 FORUM*, 2020, Reno, NV. Disponível em: <https://arc.aiaa.org/doi/abs/10.2514/6.2020-1732>.

YAN, X. et al. Energy Efficient Task Offloading in Edge Computing-Enabled UAV Networks. In: *26th ACM INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING (MobiCom '20)*, 2020, London, United Kingdom. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3380446.3430643>.

SUSIK, M. et al. Investigating the Effect of Process Parameters on Dimensional Accuracy and Material Hardness in Laser Powder Bed Fusion of Ti-6Al-4V. *Materials*, [S.l.], v. 13, n. 4, p. 922, 2020. Disponível em: <https://www.mdpi.com/1996-1944/13/4/922>.

ALIZADEH, R.; PALOMAR, D. P. MIMO Beamforming for the Cognitive Radio Relay Channel with Direct Links. *Communications in Information and Systems*, [S.l.], v. 15, n. 3, p. 283-312, 2015. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010448515000469>. Acesso em: 30 jun. 2023.

SILVA, G. N. et al. Contribuição do Exame de Tomografia de Coerência Óptica no Diagnóstico de Ceratite Infecciosa por Fusarium. *Revista Interdisciplinar de Ciências Médicas*, [S.l.], v. 2, n. 2, p. 13-20, 2015. Disponível em: <http://revista.fcmmg.br/index.php/RICM/article/view/63>. Acesso em: 30 jun. 2023.

Benefits of DC Motors and Why They're Great for Robotics. *Automate*. Disponível em: <https://www.automate.org/blogs/benefits-of-dc-motors-and-why-they-re-great-for-robotics>. Acesso em: 30 de junho de 2023.

Energy-efficient resource allocation in virtualized cloud data centers. *Energy*, Volume 198, 117126, 2020. Disponível em: [\[https://www.sciencedirect.com/science/article/abs/pii/S0360544220319666\]](https://www.sciencedirect.com/science/article/abs/pii/S0360544220319666). Acesso em: 30 de junho de 2023.

Role of BMS (Battery Management System) in Electric Vehicles. *TT Consultants*,. Disponível em:

<https://ttconsultants.com/role-of-bms-battery-management-system-in-electric-vehicles/>. Acesso em: 30 de junho de 2023.

YANG, C. et al. Coded-Caching-Aided Ultra-Dense Networks: Delay Analysis and Optimization. *IEEE Transactions on Wireless Communications*, [S.l.], v. 17, n. 12, p. 8162-8177, 2018. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8395376>. Acesso em: 30 jun. 2023.

ZHOU, L. et al. Artificial Intelligence-Enabled Secure and Trustworthy Vehicular Edge Computing. *IEEE Transactions on Vehicular Technology*, [S.l.], v. 69, n. 10, p. 11488-11499, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9193968>. Acesso em: 30 jun. 2023.

GONG, Y. et al. Network Structure Constrained Multi-Objective Evolutionary Algorithm for Green Virtual Machine Placement in Cloud Computing. In: 2017 IEEE 37th INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (ICDCS), 2017, Atlanta, GA, USA. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8036285>. Acesso em: 30 jun. 2023.

YANG, L. et al. Compressive Sensing Based Efficient User Selection Scheme for Device-to-Device Communication Underlying Cellular Networks. *Journal of Communications and Networks*, [S.l.], v. 20, n. 4, p. 359-369, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1002007118307536>. Acesso em: 30 jun. 2023.

LOUCKS, J. R. et al. Introducing Filters and Amplifiers Using a Two-Resistor Circuit. *The Physics Teacher*, [S.l.], v. 53, n. 8, p. 478-481, 2015. Disponível em: <https://pubs.aip.org/aapt/pte/article-abstract/53/8/478/277592/Introducing-Filters-and-Amplifiers-Using-a-Two>. Acesso em: 30 jun. 2023.

HAMID, S. M.; MURAT, H. Fault location and section identification on overhead transmission lines using current travelling waves. In: 2013 IEEE POWER & ENERGY SOCIETY GENERAL MEETING, 2013, Vancouver, BC. Disponível em: <https://ieeexplore.ieee.org/document/6744191>. Acesso em: 30 jun. 2023.

SILVA, R. S. et al. Proposta de uma metodologia de desenvolvimento de software utilizando a técnica de Gamificação. In: IETE INTERNATIONAL CONFERENCE ON FRONTIERS OF ELECTRICAL, ELECTRONICS AND COMPUTER ENGINEERING (IETE 2017), 2017, Jabalpur, India. Disponível em: <https://ijecscse.org/papers/IETE2017/6.pdf>. Acesso em: 30 jun. 2023.

LIU, B.; TIAN, F. Adaptive fault section estimation for double-circuit transmission lines based on parallel unscented particle filter. In: 2014 CHINA INTERNATIONAL CONFERENCE ON ELECTRICITY DISTRIBUTION (CICED), 2014, Shanghai, China. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6724721>. Acesso em: 30 jun. 2023.

JOSHI, R. et al. Performance analysis of energy efficient shortest path algorithm for underwater sensor networks. In: 2013 INTERNATIONAL CONFERENCE ON

INFORMATION PROCESSING (ICIP), 2013, Shimla, India. Disponível em: <https://ieeexplore.ieee.org/abstract/document/954206>. Acesso em: 30 jun. 2023.

WANG, X. et al. A state-of-the-art survey on computational intelligence in wireless body area networks: Towards the era of human-in-the-loop healthcare. *Information Sciences*, [S.l.], v. 258, p. 1-13, 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128610001568>. Acesso em: 30 jun. 2023.

CUI, L.; CHEN, X.; LU, W. Data fusion based weighted association rule mining algorithm and its application on fraud detection in telecom. *Future Generation Computer Systems*, [S.l.], v. 29, n. 1, p. 416-423, 2013. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X13000241>. Acesso em: 30 jun. 2023.

FENG, H.; WANG, X.; ZHU, W. Capacity Analysis for Cognitive Radio Networks with Primary User Cooperation. *Wireless Personal Communications*, [S.l.], v. 61, n. 4, p. 751-769, 2011. Disponível em: <https://link.springer.com/article/10.1007/s11277-011-0288-5>. Acesso em: 30 jun. 2023.

Data Loggers ou Registrador de Dados de Temperatura: O Que Você Precisa Saber Antes de Comprar. IMPAC Blog. Disponível em: <https://impac.com.br/blog/artigos/data-loggers-ou-registrador-de-dados-de-temperatura-o-que-voce-precisa-saber-antes-de-comprar>. Acesso em: 30 de junho de 2023.

BAZGHALI, M. R. et al. Efficient User Association and Power Allocation in Non-orthogonal Multiple Access Networks. In: 2020 INTERNATIONAL CONFERENCE ON COMPUTING, NETWORKING and COMMUNICATIONS (ICNC), 2020, Big Island, HI, USA. Disponível em: https://link.springer.com/chapter/10.1007/978-981-16-0733-2_48. Acesso em: 30 jun. 2023.

WANG, Z. et al. Reinforcement Learning Based Resource Allocation for Energy Harvesting Cognitive Radio Networks. In: 2020 18th INTERNATIONAL SYMPOSIUM ON MODELING AND OPTIMIZATION IN MOBILE, AD HOC, AND WIRELESS NETWORKS (WiOpt), 2020, Volterra, Italy. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9151268>. Acesso em: 30 jun. 2023.

SANTOS, T. R. B. et al. Q-Learning Based Dynamic Spectrum Allocation for Cognitive Radio Networks. In: 2017 INTERNATIONAL CONFERENCE ON COMMUNICATIONS (COMM), 2017, Bucharest, Romania. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8319221>. Acesso em: 30 jun. 2023.

YANG, K. et al. Cognitive Radio Spectrum Sensing with Missing Samples and Noise Uncertainty. In: 2021 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS (ICC), 2021, Montreal, QC, Canada. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9507521>. Acesso em: 30 jun. 2023.

HAN, Z. et al. Cooperative Spectrum Sensing in Cognitive Radio Systems with Multiple Antennas. In: 2006 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2006, Istanbul, Turkey. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1646376>. Acesso em: 30 jun. 2023.

NXP SEMICONDUCTORS. PCF8584 I²C and SMBus clock/calendar. NXP Semiconductors, 2010. Disponível em: <https://www.nxp.com/docs/en/data-sheet/PCF8584.pdf>. Acesso em: 30 jun. 2023.

Mouser Electronics. SPI Interface Specification. Disponível em: https://www.mouser.com/pdfdocs/tn15_spi_interface_specification.PDF. Acesso em: 06 de Agosto de 2023

BOSCH. Controller Area Network (CAN) Version 2.0B. Disponível em: <http://esd.cs.ucr.edu/webres/can20.pdf>. Acesso em: 06 de Agosto de 2023.

OASIS (Organization for the Advancement of Structured Information Standards). MQTT Version 3.1.1 Specification. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>. Acesso em: 06 de Agosto de 2023.

MOULY, Michel; PAUTET, Marie-Bernadette. The GSM system for mobile communications. Telecom publishing, 1992.

HOLMA, Harri; TOSKALA, Antti (Ed.). LTE for UMTS: OFDMA and SC-FDMA based radio access. John Wiley & Sons, 2009.

LI, J. et al. Real-Time Data Management Scheme Based on Deep Learning for Industrial Internet of Things. In: 2018 IEEE 16th INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 2018, Porto, Portugal. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8377263>. Acesso em: 30 jun. 2023.

MEDIA PUBLISHER. Network analysis of air transportation in Southeast Asia. International Journal of Innovation, Management and Technology, [S.l.], v. 11, n. 2, p. 97-101, 2020. Disponível em: <http://media-publisher.eu/wp-content/uploads/2020/11/7-2-2020.pdf>. Acesso em: 30 jun. 2023.

BAKRY, Abbas M. Al; RAJAA, D. Resan. Smart phone-Arduino based of smart door lock/unlock using RC4 stream cipher implemented in smart home. International Journal of Advanced Computer Technology, v. 5, n. 5, p. 14-18, 2016. Acesso em: 30 ago. 2023

SAAD, W. et al. An Auction Mechanism for Resource Allocation in Cloud Computing. International Journal of Computer Networks and Information Security, [S.l.], v. 9, n. 6, p. 10-16, 2017. Disponível em: <https://www.mecs-press.org/ijcnis/ijcnis-v9-n6/IJCNIS-V9-N6-2.pdf>. Acesso em: 30 jun. 2023.

ZHENG, X. et al. Secure Node Recruitment Algorithm Based on Trust Model in Wireless Sensor Networks. In: 2003 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2003, Anchorage, AK, USA. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1253273>. Acesso em: 30 jun. 2023.