

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

JOÃO VITOR ALVARENGA MIOTTO

DESENVOLVIMENTO DE SISTEMA DE TELEMETRIA PARA PROTÓTIPO DE
CARRO ELÉTRICO APLICADO A COMPETIÇÕES DE EFICIÊNCIA ENERGÉTICA

Joinville
2023

JOÃO VITOR ALVARENGA MIOTTO

DESENVOLVIMENTO DE SISTEMA DE TELEMETRIA PARA PROTÓTIPO DE
CARRO ELÉTRICO APLICADO A COMPETIÇÕES DE EFICIÊNCIA ENERGÉTICA

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Prof. Dr. Ricardo José Pfitscher

Joinville
2023

AGRADECIMENTOS

Gostaria de expressar meu agradecimento aos meus pais que nunca deixaram de me apoiar, pela paciência nos períodos mais difíceis e total disponibilidade ao longo dessa jornada. Também aos meus irmãos por todo o apoio e torcida durante todo o período de estudo.

Ao meu professor orientador, Prof. Dr. Ricardo José Pfitscher, por todo o auxílio prestado no desenvolvimento desse projeto, interesse e paciência ao lidar comigo.

Aos meus novos e antigos colegas de equipe por confiarem em mim a responsabilidade desse trabalho de grande importância, em especial ao setor de sistemas embarcados, do qual faço parte, que não mediu esforços para me ajudar.

Aos meus amigos da UFSC que conviveram comigo pelos últimos anos e são uma importante parte da minha vivência nessa universidade. Um agradecimento especial àqueles que estiveram ao meu lado durante a maior parte dos dias em que estive desenvolvendo esse projeto, me acompanhando diariamente por longos períodos de estudo.

Cada um de vocês terá pra sempre um espaço especial na minha memória e no meu coração, como os grandes responsáveis e participantes de um longo período em busca da realização de um sonho.

RESUMO

Os veículos elétricos representam importante parte do mercado de automóveis, ganhando cada vez mais espaço e se tornando cada vez mais atrativo à medida que vão desenvolvendo-se novas tecnologias, seja pela diminuição nos poluentes quando comparado à combustão, da economia, redução no ruído, ou até com a potência gerada, aproximando-se cada vez mais de outras matrizes energéticas. Para que os veículos elétricos alcançassem mais espaço ao longo dos anos, diversos aperfeiçoamentos foram realizados o tornando mais competitivo, tendo em vista que uma importante fonte de conhecimento são as competições e desenvolvimento científico proveniente de universidades. Uma prestigiada competição neste âmbito é a Shell Eco-Marathon, competição universidade com foco em eficiência energética. Este trabalho tem por objetivo o desenvolvimento de um sistema de telemetria que auxilie a equipe Eficem no controle e no desempenho do seu protótipo de carro elétrico durante a competição.

Palavras-chave: Telemetria. Eficiência energética. Shell Eco-Marathon. Protótipo automotivo.

ABSTRACT

Electric vehicles are increasingly gaining prominence in the automobile market, driven by the continuous development of new technologies. These vehicles offer several advantages, such as reduced emissions compared to internal combustion vehicles, energy efficiency, lower noise levels, and performance approaching other energy sources. To ensure the advancement and competitiveness of electric vehicles, ongoing improvements are being made, with competitions and academic research playing a vital role in acquiring knowledge and innovative solutions. The Shell Eco-Marathon, a prestigious university competition focused on energy efficiency, stands out in this context. This work aims to develop a telemetry system that assists the Eficem team in controlling and optimizing the performance of their electric car prototype during the competition. The telemetry system enables real-time monitoring of various vehicle parameters, providing accurate information that contributes to enhancing energy efficiency and the performance of the electric car. Furthermore, this project has the potential to generate new insights and contribute to the continuous evolution of electric vehicles, promoting sustainable development and the adoption of more efficient and environmentally friendly technological solutions.

Keywords: Telemetry. Energy efficiency. Shell Eco-Marathon. Automotive prototype.

LISTA DE FIGURAS

Figura 1 – Propaganda de carro elétrico a venda	16
Figura 2 – Diagrama de implantação do sistema proposto	25
Figura 3 – Escopo de projeto 1	26
Figura 4 – Escopo de projeto 2	26
Figura 5 – Dispositivos encontrados na arquitetura de um microcontrolador	27
Figura 6 – Método de leitura de velocidade via sensor óptico	29
Figura 7 – Circuito divisor de tensão	31
Figura 8 – Triangulação de localização do GPS a partir dos satélites	33
Figura 9 – Comunicação MQTT	39
Figura 10 – Diagrama de blocos para alimentação dos controladores	42
Figura 11 – Conector USB tipo C utilizado nos testes	42
Figura 12 – Tela de visualização do piloto	52
Figura 13 – Dashboard de visualização da equipe externa	53
Figura 14 – Sistema de Telemetria para Protótipo de Carro Elétrico	54

LISTA DE QUADROS

Quadro 1 – Medições de Carga do Raspberry Pi e ESP	43
Quadro 2 – Teste de Conexão dos Microcontroladores	44
Quadro 3 – Medições realizadas a partir do circuito divisor de tensão	49
Quadro 4 – Requisitos de Sistema e erros associados	57

LISTA DE TABELAS

Tabela 1 – Medições realizadas para determinação de RPM alto	46
Tabela 2 – Medições realizadas para determinação de RPM baixo	46
Tabela 3 – Medições realizadas pelo sensor ACS712	48
Tabela 4 – Comparativo entre dados coletados por sensor GPS e via Google Maps	50

LISTA DE SIGLAS

ARM	Advanced RISC Machine
CTJ	Centro Tecnológico de Joinville
DARPA	Defense Advanced Research Projects Agency
GPIO	General Purpose Input/Output
GPS	Global Positioning System
HDMI	High-Definition Multimedia Interface
IDE	Integrated Development Environment
IoT	Internet das Coisas
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LIDAR	Light Detection and Ranging
LOG	Histórico de Registros
LPDDR	Low-Power Double Data Rate
MQTT	Message Queuing Telemetry Transport
MUEE	Maratona Universitária da Eficiência Energética
OLED	Organic Light-Emitting Diode
QoS	Quality of Service
RPM	Rotações por minuto
SEM	Shell Eco-marathon
SEMA	Shell Eco-marathon Americas
SEMB	Shell Eco-marathon Brasil
SGBD	Sistema Gerenciador de Banco de Dados
SO	Sistema Operacional
VLTs	Veículos Leves sobre Trilhos
Wi-Fi	Wireless Fidelity

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	VEÍCULOS ELÉTRICOS	15
2.1.1	Tipos de veículos elétricos	16
2.2	COMPETIÇÕES UNIVERSITÁRIAS	18
2.2.1	Shell Echo-Marathon	18
2.2.2	Eficem - UFSC	18
2.3	TELEMETRIA	19
2.3.1	Microcontroladores - hardware embarcado local	20
2.3.2	Comunicação e transmissão de dados	20
2.3.3	Software supervisorio	21
3	DESENVOLVIMENTO DE PROJETO	23
3.1	PROPOSTA DE SISTEMA DE TELEMETRIA	23
3.1.1	Requisitos Funcionais	23
3.1.2	Requisitos Não-Funcionais	24
3.1.3	Arquitetura de Projeto	24
3.2	IMPLEMENTAÇÃO DO PROTÓTIPO	25
3.2.1	Microcontroladores	27
3.2.2	Sensor de velocidade	28
3.2.3	Sensor de potência	30
3.2.3.1	Sensor de corrente	30
3.2.3.2	Sensor de tensão	31
3.2.4	Sensor de carga da Bateria	32
3.2.5	Sensor de localização	32
3.2.6	Sensor de temperatura	33
3.2.7	Banco de dados	34
3.2.8	Interface de Visualização	35
4	METODOLOGIA	36
4.1	MICROCONTROLADORES	36
4.2	SENSOR DE VELOCIDADE	37
4.3	SENSOR DE CORRENTE	37

4.4	SENSOR DE TENSÃO	38
4.5	SENSOR DE TEMPERATURA	38
4.6	COMUNICAÇÃO	39
4.7	BANCO DE DADOS	40
5	RESULTADOS	41
5.1	MICROCONTROLADORES	41
5.1.1	Análise de Carga	41
5.1.2	Teste de conexão	43
5.2	VELOCIDADE	45
5.3	CORRENTE - ACS712	47
5.4	TENSÃO - DIVISOR DE TENSÃO	48
5.5	LOCALIZAÇÃO - GPS-NEO6	49
5.6	TEMPERATURA - DS18B20	50
5.7	COMUNICAÇÃO - MQTT	50
5.8	BANCO DE DADOS - MONGODB ATLAS	51
5.9	INTERFACES DE VISUALIZAÇÃO	52
5.10	MONTAGEM DO SISTEMA DE TELEMETRIA	53
6	CONCLUSÕES	56
	REFERÊNCIAS	58
	APÊNDICE A	63
	APÊNDICE B	66
	APÊNDICE C	69
	APÊNDICE D	71
	APÊNDICE E	73
	APÊNDICE F	75
	APÊNDICE G	76
	APÊNDICE H	79
	APÊNDICE I	81
	APÊNDICE J	83
	APÊNDICE K	85

APÊNDICE L	88
APÊNDICE M	89
APÊNDICE N	91
APÊNDICE O	92

1 INTRODUÇÃO

Apesar da história dos carros elétricos começar ainda no século XIX, associada às das baterias (HOYER, 2008), somente com o aumento da preocupação com o meio ambiente, a partir de 1960, os carros elétricos passaram a ser um interesse das empresas automotivas (BARAN; LEGEY, 2010). Com o crescimento do número de automóveis elétricos, estudos vêm sendo desenvolvidos para aumentar a eficiência energética e ampliar as tecnologias, buscando tornar os automóveis elétricos competitivos economicamente e uma alternativa sustentável comparados aos automóveis a combustão (CHRISPIM; TORRES; SIMÕES, 2016).

As tecnologias desenvolvidas para automóveis elétricos, têm início nas competições esportivas de automobilismo e competições relacionadas a protótipos de veículos e veículos motorizados. Uma importante competição universitária de protótipo de veículos e âmbito global, é a Shell Eco-marathon, que “[...]desafia jovens a pensar, desenhar e construir protótipos de carros ultra eficientes.” (SHELL BRASIL, 2022), e ocorre anualmente, a nível nacional e internacional.

Nas competições da Shell Eco-Marathon, a equipe é responsável por desenvolver um projeto e construir um protótipo de carro com capacidade para uma pessoa, considerando a carenagem, sistemas veiculares e sistema de propulsão, que pode ser via motor elétrico ou a combustão, preocupando-se com a segurança e dirigibilidade do piloto. Além disso, é responsável pelo controle das informações fornecidas ao piloto, bem como qualquer outra informação coletada que possa auxiliar a equipe, visando sempre a eficiência energética do protótipo (SHELL BRASIL, 2022).

Essa coleta de informações deve levar em consideração que as competições ocorrem anualmente em diferentes locais, apresentando alterações na pista e nos obstáculos para comunicação, além da necessidade de o sistema estar integrado sem uso de alimentação externa, utilizando apenas uma bateria para todo o automóvel. Dessa forma, um projeto de telemetria que forneça informações do carro e obstáculos físicos no local, tanto ao piloto quanto para áreas externas do carro, pode auxiliar na condução do veículo durante a prova (STERNAL et al., 2012)

Entretanto, a complexidade para implementação de múltiplos sensores no automóvel e a limitação da alimentação do sistema, são fatores que dificultam a criação de projetos complexos para obtenção e envio de informação do protótipo à nuvem. A criação desses sistemas, portanto, exige que sejam realizados testes em circuitos separados, garantindo o funcionamento de cada uma das etapas, bem como garantir a boa comunicação das informações em um sistema único integrado, prevendo possíveis interferências geradas, entre outras limitações enfrentadas por uma equipe universitária,

como o a Equipe de Eficiência Energética (Eficem), do Centro Tecnológico de Joinville (CTJ).

Criada por alunos do curso de Engenharia Automotiva do CTJ , o Eficem surgiu em 2012, com a união entre as equipes Catarina e Feneco, sempre com o foco em eficiência energética, as equipes possuíam protótipos automotivos elétrico e à combustão, respectivamente. Nos dois primeiros anos a equipe recebeu premiações de terceiro melhor projeto na Maratona Universitária da Eficiência Energética (MUEE), com o protótipo elétrico denominado Miliwatt I, em 2012 e com o protótipo a combustão, que recebia o nome de V12, em 2013. O prêmio mais notável obtido pela equipe ocorreu em 2017, obtendo a primeira colocação na Shell Eco-marathon Brasil, batendo o recorde latino-americano, alcançando a marca de 363,6 km/kWh com o protótipo elétrico Setta (EFICEM, 2020).

No entanto, a carência de informações disponíveis para o piloto, assim como a ausência de informações do protótipo durante as voltas realizadas na competição, geram dificuldades ao longo das competições, como a impossibilidade de definir a origem de problemas oriundos das voltas, controlar a velocidade do protótipo, localização do piloto quanto ao número de voltas e ociosidade da equipe externa. Informações estas, dependentes do retorno do carro à oficina para iniciar as análises e investigar possíveis melhorias (EFICEM, 2020).

A experiência da equipe Eficem, do CTJ, em competições nacionais, tem demonstrado que poucas equipes apresentam projetos para telemetria na Shell Eco-Marathon, normalmente, sendo realizada apenas a verificação de velocidade de forma local, para o piloto. Ou então, quando há comunicação com a equipe externa, são utilizados equipamentos de transmissão restrita a uma área aberta, de curta distância, o que limita o monitoramento a partir de ambientes fechados, como é o caso nos boxes.

Visando resolver essa problemática e oferecer suporte a melhorias futuras, este trabalho de conclusão de curso projetou e desenvolveu um sistema de telemetria para o protótipo de carro elétrico da equipe Eficem. O sistema de telemetria proposto suporta o monitoramento direto e em tempo real, enviando informações ao piloto e a um sistema de armazenamento para manutenção do histórico para comparação dos dados coletados. Com esse sistema é possível verificar possíveis falhas e/ou permitir controle por parte da equipe externa quanto à manutenção e condições elétricas/eletrônicas do veículo, implicando em segurança e confiança para o piloto, abrindo oportunidades de melhoria no desempenho da equipe na competição. Ainda, espera-se permitir que trabalhos futuros aprimorem o sistema para oferecer ferramentas de visualização, apoio a condução e controle automatizado, aumentando assim a competitividade da equipe de competição.

1.1 OBJETIVOS

Para sanar a necessidade de um sistema de aquisição e apresentação de dados para protótipos de carros elétricos, no escopo de competições de eficiência energética, propõe-se os seguintes objetivos.

1.1.1 Objetivo Geral

Desenvolver um sistema de telemetria com apresentação de dados ao piloto e à equipe externa, para o protótipo do carro elétrico da equipe Eficem do CTJ-UFSC.

1.1.2 Objetivos Específicos

- Definir requisitos de monitoramento de um veículo elétrico no escopo da Shell Eco-Marathon;
- Projetar um sistema para coleta dos dados em tempo real;
- Desenvolver uma interface de apresentação dos dados ao piloto e a equipe externa;
- Validar o sistema de telemetria quanto aos requisitos de monitoramento definidos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados conceitos fundamentais para a compreensão de um sistema de telemetria e sua importância no mundo moderno aplicado a carros elétricos no contexto de competições de eficiência energética. Serão explicitados cada uma das partes que formam o sistema, os componentes mais importantes em âmbitos competitivos, como se da sua aplicação em competições de eficiência energética, as possíveis contribuições para o desempenho e um aprofundamento sobre a Shell Eco-marathon (SEM), principal competição para a equipe Eficem.

2.1 VEÍCULOS ELÉTRICOS

Para compreender a história dos veículos elétricos, deve-se primeiro considerar que se trata de uma trajetória que precede automóveis e da própria gasolina como combustível. Em 1837 era construída a primeira locomotiva elétrica, desenvolvida por Robert Davison Aberdeen, propulsionada por um motor elétrico e abastecida por uma bateria rústica de ferro-zinco (BOTTURA; BARRETO, 1989).

A partir de 1859, Gaston Planté apresentou a primeira bateria de chumbo ácido, que após a década de 1880 seria utilizado na França, EUA e Reino Unido para o desenvolvimento de diversos veículos elétricos. Apenas em 1885 foi realizada a primeira demonstração de um motor de combustão interna apresentado por Karl Benz. O interesse no potencial apresentado pelos veículos elétricos fez Thomas Edison desenvolver a bateria níquel-ferro, com capacidade de armazenamento 40% maior que a bateria de chumbo em 1901 (BARAN; LEGEY, 2010).

Ao longo dos anos, novas tecnologias foram desenvolvidas como apostas ao crescimento da utilização de veículos elétricos da época, os deixando mais competitivos e viáveis.

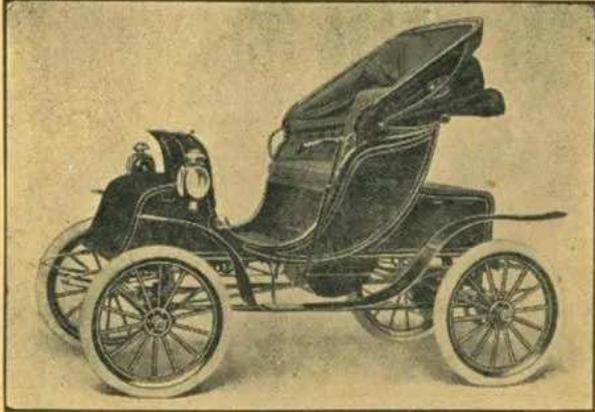
Além das baterias, duas tecnologias desenvolvidas entre 1890 e 1900 contribuíram para melhorar o desempenho dos carros elétricos: a frenagem regenerativa, um equipamento capaz de transformar a energia cinética do automóvel em movimento em energia elétrica durante uma frenagem e o sistema híbrido a gasolina e eletricidade (BARAN; LEGEY, 2010, p. 212).

Um dos grandes marcos do início do século XX foi a implementação de linhas de montagem e produção em massa, dando início a fase moderna da industrialização (Figura 1), reduzindo os valores dos veículos de propulsão a combustão interna, desencadeando um grande crescimento na indústria automobilística e na utilização de

Figura 1 – Propaganda de carro elétrico a venda

**The 100 Mile
Fritchle Electric**

The Only Electric Guaranteed to Go 100 Miles on One Charge,



MODEL "A" VICTORIA PHAETON.

The Victoria Phaeton shown here, is an ideal lady's carriage for city and country use. Its artistic and impressive body design, its superb painting and upholstering make it the most attractive lady's car ever offered to the public.

Harry L. Cort, Sole Agent
Moore Theatre, Phone Main 6103.

Can deliver 10 days after order is placed. Guaranteed against defective parts, material and workmanship for one year from date of delivery.

Fonte: (AMARAL, 2022)

combustíveis a base de petróleo. Dessa forma, os veículos elétricos foram esquecidos pela indústria automotiva e permaneceram ignorados até 1960, quando voltaram à se tornar atrativos devido aos questionamentos realizados à utilização de combustíveis fósseis e crescente preocupação com o meio ambiente (GOLDENSTEIN; AZEVEDO, 2006).

2.1.1 Tipos de veículos elétricos

Com a crescente preocupação com as emissões de gases de efeito estufa, a busca por soluções mais sustentáveis, que garantissem uma maior segurança energética aos países, e a superação de entraves tecnológicos, levaram a indústria automobilística a investir em projetos de veículos elétricos (RIBEIRO; FERREIRA, 2010). Neste contexto, é possível citar diferentes tipos de veículos elétricos disponíveis atualmente, bem como suas aplicações práticas, explorando seus benefícios e desafios na transição para uma mobilidade mais limpa e eficiente.

Veículos elétricos híbridos são aqueles que combinam motores elétricos e motores a combustão em sua construção. Esses veículos oferecem diferentes

configurações, cada uma com suas particularidades e benefícios (DELGADO et al., 2017). Uma configuração comum é aquela em que o motor a combustão é o propulsor principal, nesse caso, o motor elétrico é utilizado principalmente para fornecer tração em baixa potência, aumentando a eficiência do veículo. Nessa configuração, o veículo é denominado híbrido paralelo, e o fornecimento de energia ao motor elétrico é realizada por meio da frenagem regenerativa.

Outra configuração encontrada nos veículos elétricos híbridos é aquela em que o motor a combustão é o propulsor principal, mas o veículo também pode receber energia elétrica por meio de abastecimento externo. Essa configuração permite uma maior autonomia ao veículo, combinando a eficiência do motor elétrico com a disponibilidade do motor a combustão. Ainda de acordo com Delgado et al. (2017), há também a configuração em que o motor elétrico é o propulsor principal, sendo alimentado externamente. Nesse caso, o motor a combustão interna funciona como um gerador, fornecendo energia para recarregar a bateria do veículo. Essa configuração garante uma maior autonomia ao utilizar a energia elétrica como fonte principal de propulsão, e o motor a combustão entra em ação apenas para fornecer energia adicional quando necessário.

Essas diferentes configurações de veículos elétricos híbridos demonstram a flexibilidade e as possibilidades que essa tecnologia oferece. A combinação de motores elétricos e motores a combustão proporciona eficiência energética, redução de emissões e maior autonomia, contribuindo para a transição para um sistema de transporte mais sustentável.

Outro modelo de veículo elétrico, comumente empregado ao transporte público, utilizam motores elétricos com torque elevado, alimentados por cabos externos diretamente conectados, possuindo infraestrutura planejada, ligadas a redes eletrificadas situadas acima, como as utilizadas em *trolley buses*, ou abaixo, como em Veículos Leves sobre Trilhos (VLTs), abastecidos em movimentos (DELGADO et al., 2017).

Os veículos elétricos puros, que possuem como propulsor somente o motor elétrico, são abastecidos exclusivamente por energia elétrica. Essa categoria é também subdividida em veículos elétricos alimentados por baterias (VEB), normalmente abastecido via rede elétrica, esse modelo é o mais comum e difundido na atualidade sendo produzido por indústrias novas e tradicionais, e o modelo movido à célula de combustível (VECC) que possui como base a utilização de um dispositivo eletro-químico capaz de converter a energia liberada pelo tanque pressurizados com hidrogênio em energia elétrica (LEAL; CONSONI, 2021).

2.2 COMPETIÇÕES UNIVERSITÁRIAS

As universidades desempenham um importante papel na construção do conhecimento científico e possuem participação fundamental no desenvolvimento de novas tecnologias e inovações que são utilizadas na sociedade e na indústria. Desde 1970, governos de todo o mundo lançaram iniciativas de incentivo às universidades com o intuito de vincular e estreitar a relação do conhecimento produzidos nas universidades às inovações industriais (MOWERY; SAMPAT, 2005).

Os investimentos em pesquisas nas universidades, tanto de meios públicos quanto nos privados, permanecem sendo um importante pilar para construção da ciência. Uma das vias de incentivo à geração do conhecimento é a partir de competições universitárias que promovem o desenvolvimento de soluções para as mais diversas áreas do conhecimento.

2.2.1 Shell Echo-Marathon

A ideia para a SEM surgiu há mais de 80 anos, a partir de uma disputa entre funcionários da Shell nos Estados Unidos, que competiam para verificar quem conseguia ir mais longe, com carros adaptados para maximizar a eficiência energética, utilizando uma mesma quantidade de combustível. Em 1985, na França, a competição foi oficialmente lançada, com a participação de 25 equipes (SHELL, 2023).

Atualmente, o evento mantém como proposta a construção de protótipos e veículos visando a máxima eficiência, portanto, percorrer a maior distância possível utilizando a menor quantidade de combustível das diferentes matrizes energéticas, considerando para o desenvolvimento, parte mecânica e elétrica dos veículos. Desde sua criação, a competição já contou com a participação de mais de 100.000 alunos de mais de 60 países (SHELL, 2023).

A SEM conta com 3 edições, abrangendo as regiões: Américas, Ásia-Pacífico e Oriente Médio, Europa e África e contemplam duas classes de veículos: Urban Concept, veículos com 4 rodas, mais próximos a carros comuns de estrada, e protótipos de ultra eficiência, mais leves, normalmente com 3 rodas, projetados para garantir a máxima eficiência do combustível. Além disso, as categorias variam entre 3 matrizes energéticas, separadas em motor de combustão interna, bateria elétrica e célula combustível de hidrogênio.

2.2.2 Eficem - UFSC

A equipe de eficiência energética (Eficem), surgiu em 2012 como resultado da fusão de duas equipes: a Catarina e a Feneco. A equipe Catarina focava no desenvolvimento de protótipos elétricos, enquanto a equipe Feneco trabalhava com protótipos à gasolina. Ambas tinham um objetivo em comum: buscar soluções para

maximizar a eficiência energética. A criação da equipe Eficem foi uma iniciativa dos alunos do curso de Engenharia Automotiva do antigo Centro de Engenharias da Mobilidade, atualmente conhecido como Centro Tecnológico de Joinville da Universidade Federal de Santa Catarina (CTJ-UFSC). Na época, o objetivo principal era participar da Maratona Universitária da Eficiência Energética (MUEE) (EFICEM, 2020)

Em sua primeira participação na competição, no mesmo ano de criação da equipe, o Eficem conquistou o prêmio de terceiro melhor projeto da maratona com o protótipo elétrico Milliwatt, concedido pelo canal Discovery Turbo. Repetindo a premiação de terceiro melhor projeto no ano seguinte, mas dessa vez, com o projeto a combustão V12. Desde então, a equipe tem registrado diversos e inovações resultado de novas pesquisas e desenvolvimentos, como a produção de um novo chassi feito em monocoque de fibra de carbono, sendo o primeiro chassi sem liga metálica (EFICEM, 2020).

Em 2014 foi finalizada a construção de um novo veículo elétrico, o Setta Energy, com menos de 24 kg recebeu o título de protótipo mais leve das Américas na Shell Eco-marathon Americas (SEMA), repetindo o feito em 2016 e 2017. Em 2017 a equipe participou novamente da SEMA e conquistou o 3º lugar com o protótipo elétrico, realizando 284 km/kWh, novo recorde latino-americano, mantendo um protótipo ainda mais leve, com 21,5 kg. Nesse mesmo ano a quipe participou da Shell Eco-marathon Brasil (SEMB) e quebrou seu próprio recorde de eficiência, realizando um percurso com 363,6 km/kWh. Desde então, a equipe permanece com foco nessas duas competições (EFICEM, 2020).

2.3 TELEMETRIA

A palavra telemetria tem origem no grego com a junção de duas palavras, sendo *Tele* que significa longe e *Meter* que significa medir. Dessa forma, podemos compreender a telemetria como a medição realizada à distância, frequentemente utilizada para coleta e transmissão de dados em locais remotos e de difícil acesso (TELEMETRIA, 2023).

O primeiro sistema de telemetria para transmissão de dados surgiu no ano de 1845, sendo utilizado em meio militar para a transmissão de dados entre Winter Palace e o quartel da armada russa. Outros usos marcantes na história, ocorreram em 1874, com a construção de um rede de sensores atmosféricos em Mont Blanc, França, com transmissão em tempo real para Paris. Em 1906 foi criado um sistema de telemetria para a transmissão de informações entre uma série de estações sísmicas para o Pulkovo Observatory, na Rússia (QUEIROS, 2011).

Mas a popularidade de sistemas de telemetria, assim como o conhecimento do

público geral sobre o tema, teve início na década de 90 com sua utilização na Fórmula 1 e em outras categorias do automobilismo, impactando e elevando competitivamente o cenário (BEZERRA, 2022).

Para compreender melhor a maneira como um sistema de telemetria pode auxiliar no monitoramento, não só de veículos, como de diversos sistemas integrados da atualidade, é possível descrever e compreendê-lo a partir de três partes principais: Hardware embarcado local, Transmissão de dados e Software para supervisão.

2.3.1 Microcontroladores - hardware embarcado local

Hardware embarcado local é a parte responsável pela obtenção dos dados no objeto a ser monitorado, considerando desde a obtenção dos dados por meio de sensores, como o tratamento dos dados, no hardware em questão, e o planejamento do local a ser implementado. Sua construção deve ser robusta, com atenção a parte mecânica empregada para se adequar aos ambientes que estes equipamentos ficarão expostos e aos tipos de forças que estarão submetidos, para manter a integridade dos circuitos e equipamentos empregados.

Os sensores e transdutores são dispositivos capazes de utilizar fenômenos naturais para quantificar grandezas física e transforma-las em sinais que podem ser interpretados digitalmente. Estes dispositivos são fruto de diversos avanços tecnológicos, capazes de medir uma ampla gama de parâmetros físicos, químicos e ambientais, permitindo monitorar o desempenho, as condições operacionais e o estado dos sistemas monitorados. Esses sensores são fundamentais para a coleta de informações, detecção de anomalias, prevenção de falhas e otimização de processos, contribuindo para a eficiência e o sucesso dos sistemas de telemetria nas mais diversas áreas, no entanto, para que seja possível realizar o tratamento dos dados obtidos por eles, é necessário a utilização de um hardware processado (WENDLING, 2010).

Os microcontroladores têm ganhado espaço na construção das mais variadas origens de projetos, e desempenham um papel crucial em sistemas de telemetria, sendo responsáveis pelo processamento, comunicação e controle dos dados coletados pelos sensores. Eles adquirem informações, realizam tratamentos necessários, como filtragem e calibração, e transmitem os dados para o sistema supervisor. Além disso, os microcontroladores gerenciam a integridade dos circuitos e dispositivos, garantindo a confiabilidade e o desempenho adequado do sistema de telemetria (OKI; MANTOVANI, 2013)

2.3.2 Comunicação e transmissão de dados

No contexto de telemetria, a transmissão de dados é responsável pelo envio dos dados obtidos no objeto monitorado para o usuário desejado. Essa transmissão pode

ocorrer por meio de protocolos de comunicação entre os equipamentos compatíveis, por meio de redes cabeadas e até mesmo, realizando a comunicação via Internet. Existem diversas opções de comunicação e transmissão de dados disponíveis, dependendo das necessidades e requisitos específicos do sistema de telemetria. Alguns dos métodos mais comuns incluem (LACERDA, 2015):

- Comunicação por cabo: Nesse método, os dados são transmitidos por meio de cabos físicos, como cabos Ethernet, cabos USB ou cabos seriais. Essa abordagem é comumente usada em sistemas com distâncias curtas e controladas, dando preferência a redes de comunicação ponto a ponto, como a conexão direta entre um sensor e um dispositivo de monitoramento.
- Comunicação sem fio: A transmissão sem fio é amplamente utilizada em sistemas de telemetria, pois oferece maior flexibilidade e mobilidade. Existem várias tecnologias sem fio disponíveis, como Wi-Fi, Bluetooth, Zigbee e LoRaWAN. Essas tecnologias permitem a transmissão de dados em distâncias maiores e a comunicação entre vários dispositivos.
- Comunicação via satélite: Em casos em que é necessária uma comunicação em áreas remotas ou em movimento, a comunicação via satélite é uma opção viável. Essa tecnologia utiliza satélites de comunicação para transmitir dados entre o objeto monitorado e o sistema de monitoramento central. É uma solução útil em aplicações como monitoramento de veículos ou navios em áreas onde a infraestrutura de comunicação terrestre é limitada (DURAND, 2008).

Além da escolha do método de comunicação, é importante considerar a segurança dos dados transmitidos. O uso de criptografia e autenticação adequadas é essencial para proteger a integridade e a confidencialidade dos dados durante a transmissão.

Cada método de comunicação possui suas vantagens e limitações, e a escolha adequada dependerá dos requisitos específicos do sistema de telemetria, como distância de transmissão, velocidade de transmissão, ambiente operacional e requisitos de segurança. No contexto deste trabalho, conforme será discutido no Capítulo 3, será utilizada a comunicação sem fio, uma vez que o veículo monitorado estará em movimento e com uma distância de até 1 Km, de acordo com o tamanho da pista.

2.3.3 Software supervisorio

Software para supervisão é o sistema de interface com o usuário que monitorará o sistema, podendo ser desenvolvido das mais diversas formas, para os mais diversos equipamentos, sempre considerando qual será o meio de transmissão de dados utilizado e o local da ponta de recebimento. Ele permite a visualização, análise e controle dos dados coletados pelos sensores, fornecendo informações em tempo real

sobre o desempenho, as condições operacionais e o estado dos sistemas monitorados.

O software supervisorio pode ser desenvolvido de diferentes maneiras, dependendo das necessidades específicas do sistema de telemetria e dos equipamentos envolvidos. Algumas características comuns de um software supervisorio incluem a interface de visualização do usuário, monitoramento em tempo real, análise de dados, acionamento, controle, conectividade, integração de sistemas e gerenciamento de alarmes.

É importante ressaltar que o software supervisorio deve ser projetado levando em consideração o meio de transmissão de dados utilizado e o local da ponta de recebimento. Ele precisa ser compatível com o método de comunicação escolhido, seja por cabo, sem fio, via satélite ou pela Internet, para garantir uma transmissão eficiente e confiável dos dados.

3 DESENVOLVIMENTO DE PROJETO

Neste capítulo é apresentado o caminho seguido na concepção do projeto para a conclusão dos objetivos determinados, partindo da análise e determinação dos parâmetros a serem estudados, requisitos de projeto e determinação de hardware e software admissíveis.

3.1 PROPOSTA DE SISTEMA DE TELEMETRIA

As definições de projeto devem levar em consideração a estrutura física do protótipo desenvolvido pela equipe Eficem, os dados a serem coletados visando o desempenho em eficiência energética e o regulamento da SEM (SHELL, 2023), além de detalhar as definições para as três áreas que a telemetria abrange.

O hardware embarcado, bem como os sensores selecionados, devem ser definidos com a intenção de coletar e transmitir os dados, respeitando as limitações de alimentação da bateria e faixa de trabalho das propriedades a serem coletadas do motor, estabelecidas pelo regulamento, capacidade do motor e da bateria. A comunicação deve ocorrer em tempo real, considerando os bloqueios físicos como paredes e localização do protótipo durante a realização das voltas. As interfaces devem apresentar os dados tratados conforme a necessidade do usuário de interesse. Ressalta-se que no contexto deste trabalho admite-se tempo real como o mais rápido possível e não o conceito de que as tarefas tem restrição temporal para execução, como em (FARINES; FRAGA; OLIVEIRA, 2000).

3.1.1 Requisitos Funcionais

Tendo em vista o cumprimento das considerações acerca do sistema, os seguintes requisitos funcionais de sistema foram definidos:

- RF1: O sistema deve ser capaz de medir a velocidade do veículo.
- RF2: O sistema deve ser capaz de fornecer a informação de potência elétrica.
- RF3: O sistema deve ser capaz de medir o estado de carga da bateria do veículo.
- RF4: O sistema deve ser capaz de verificar a localização do veículo.
- RF5: O sistema deve ser capaz de realizar o monitoramento da temperatura próxima à bateria do veículo.
- RF6: O sistema deve ser capaz de armazenar as informações em um banco de dados externo ao carro.

- RF7: O sistema deve possuir uma interface de visualização dos dados para o piloto.
- RF8: O sistema deve ser capaz de mostrar os dados coletados para a equipe externa.

3.1.2 Requisitos Não-Funcionais

Considerando as condições climáticas que podem ser encontradas durante as competições, o regulamento da SEM e critérios para estabelecer uma comunicação fluída entre a equipe e o piloto, os seguintes requisitos não-funcionais foram determinados (JANG, 2014; FIGUEIROA, 2021; SHELL, 2023):

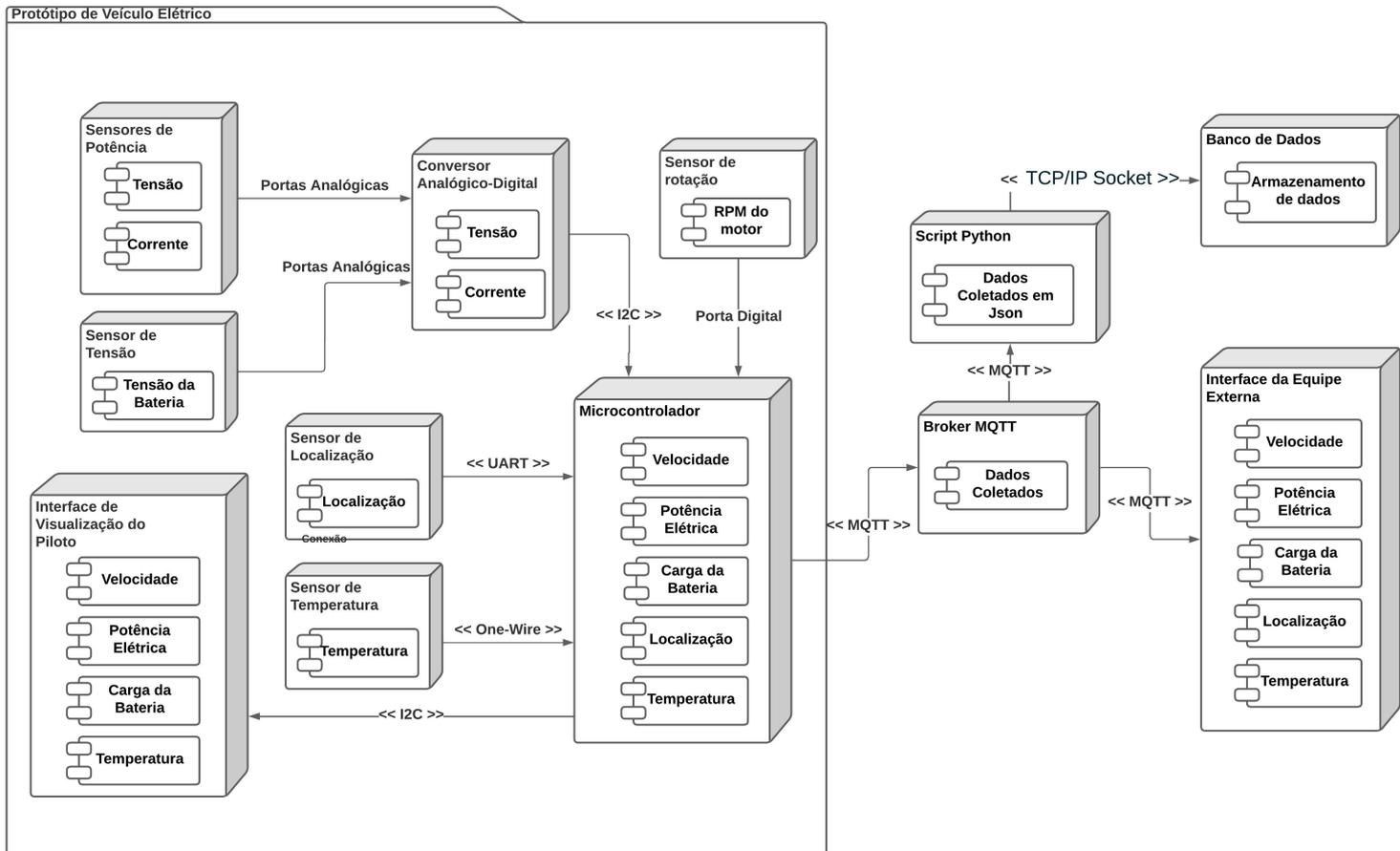
- RNF1: O sistema deve captar medições de temperatura da bateria entre -20 °C e 75 °C .
- RNF2: O sistema deve coletar e transmitir os dados da telemetria em tempo real.
- RNF3: Os componentes do sistema devem ser modulares, de forma a facilitar processos de manutenção e manipulação pela equipe.
- RNF4: O sistema deve ser capaz de ler corrente com valores entre 0 e 20 Amperes.
- RNF5: O sistema deve ser capaz de ler tensão com valores entre 0 e 36 Volts.

3.1.3 Arquitetura de Projeto

Com base nos requisitos e ferramentas disponíveis foi proposta uma arquitetura de projeto, que pode ser observado no diagrama de implantação da Figura 2, com base nos equipamentos já disponíveis na equipe, e para permitir a comunicação mesmo em ambientes fechados (boxes), foi definida a utilização da rede celular fornecida por meio de um dispositivo móvel como meio de comunicação com a equipe externa. Dessa forma um microcontrolador fica responsável pelo tratamento dos dados coletados pelos sensores dispostos no protótipo veicular, parte dos dados coletados são exibidos localmente ao piloto, conforme definições e grau de importância ao dos dados coletados, e considerando que todas as informações serão externalizadas, de forma que a equipe externa consiga visualizar e um banco de dados armazene os dados coletados.

É possível imaginar o envio das informações à equipe externa por meio de dois principais meios de comunicação. A primeira proposta (Figura 3) tem como base a transferência dos dados por meio do protocolo MQTT (*Message Queuing Telemetry Transport*), protocolo desenvolvimento com foco no transporte de dados de telemetria. Nesse cenário, a equipe externa realiza a coleta dos dados diretamente do *broker* MQTT, enquanto um *script* realiza outra inscrição no *broker* para enviar os dados coletados ao banco de dados para análise posterior.

Figura 2 – Diagrama de implantação do sistema proposto



Fonte: Autor (2023).

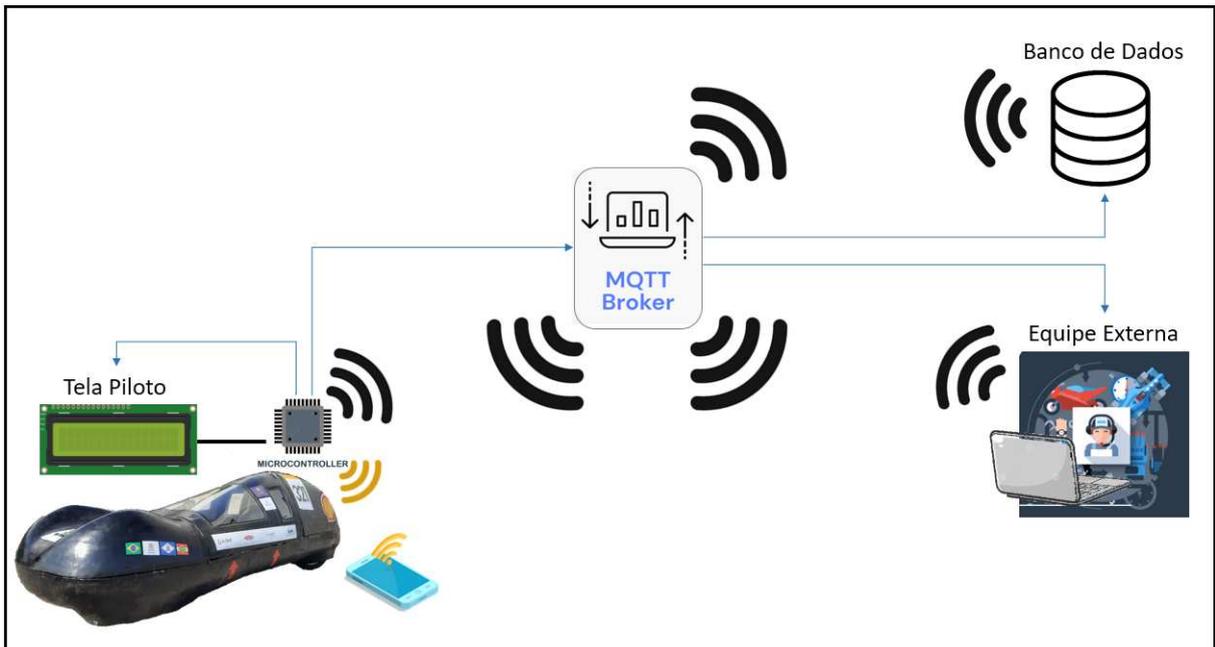
No segundo cenário, após tratar os dados, o controlador envia os dados diretamente ao banco de dados, conforme protocolo aceito pelo banco de dados selecionado, e a equipe externa realiza a visualização por meio de uma interface gráfica ligada ao banco de dados, conforme pode ser vista no esquemático da Figura 4.

A principal diferença entre as duas propostas se dá pelo protocolo de comunicação utilizado para envio dos dados dos sensores do carro para a equipe externa. Desse modo, é esperado que haja uma diferença no tempo de acesso aos dados, pela equipe externa, entre as propostas apresentadas.

3.2 IMPLEMENTAÇÃO DO PROTÓTIPO

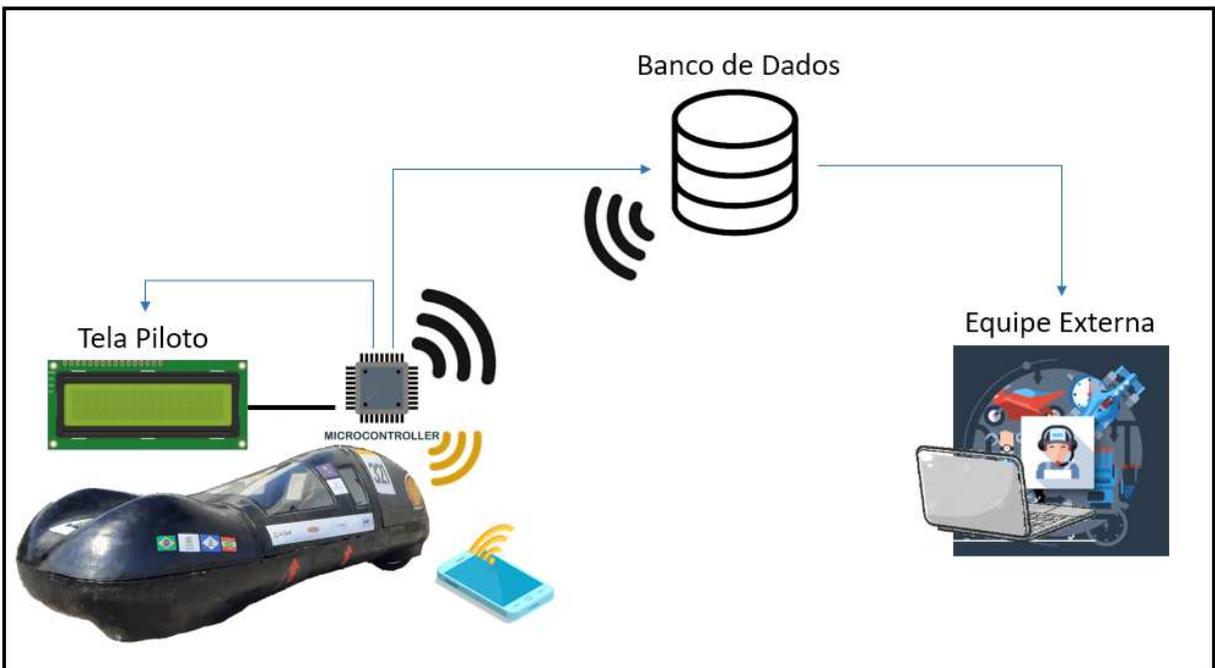
Para determinar as ferramentas utilizadas, além dos requisitos de sistema, devem ser consideradas outras variáveis decorrentes da competição, afetando diretamente na utilização de alguns dispositivos ou softwares específicos. Tendo em vista o período de três dias de competição, deve ser priorizada a utilização de hardwares de fácil obtenção e transporte, visando eventuais manutenções necessárias,

Figura 3 – Escopo de projeto 1



Fonte: Autor (2023).

Figura 4 – Escopo de projeto 2



Fonte: Autor (2023).

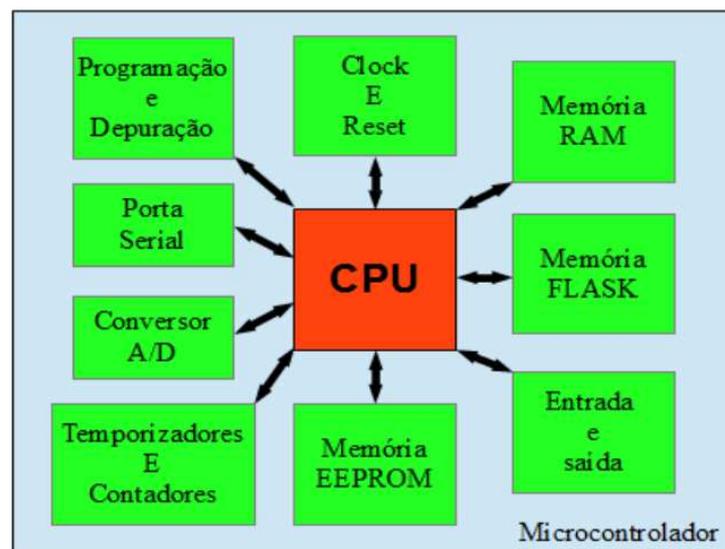
bem como priorizar equipamentos de menor custo. Além disso, considera-se que a pista é aberta, sem a presença de paredes ou teto, mas a oficina em que a equipe externa permanece está situada em ambiente fechado. Nas seguintes subseções serão discutidos os componentes presentes no sistema de telemetria aqui proposto, quais sejam, microcontrolador, sensores, sistemas de comunicação e aspectos de software.

3.2.1 Microcontroladores

Os microcontroladores são placas de circuitos integrados microprocessadas compostas de uma unidade central de processamento (CPU) programável que junto a um conjunto de dispositivos acoplados a uma única placa, contém todo o hardware necessário para realizar diversas operações, sendo necessário apenas realizar sua alimentação (KERSCHBAUMER, 2018).

Além do CPU, um microcontrolador conta com memórias de dados, memórias de programas, circuito de *clock*, portas programáveis (GPIOs) que promovem a interface entre o microcontrolador e periféricos externos, entre outros módulos integrados (Figura 5) que podem variar conforme os fabricantes. Sua arquitetura tende a ser mais simples com relação a de computadores, mas que possibilitam a realização do controle de diversos processos a partir de diversas linguagens de programação (KERSCHBAUMER, 2018).

Figura 5 – Dispositivos encontrados na arquitetura de um microcontrolador



Fonte: (KERSCHBAUMER, 2018).

Sua arquitetura mais simples e de tamanho reduzido, integrado diretamente a placa visam garantir aos microcontroladores a possibilidade de fabricação de hardwares, mais baratos, menores e com recursos eficientes, confiáveis e dedicados, com baixo consumo de energia, normalmente destinando seu uso ao controle de processos únicos (OKI; MANTOVANI, 2013).

Para a determinação do microcontrolador, é importante pensar nos meios de comunicação disponibilizados para realizar a transmissão dos dados e capacidade de operação do mesmo, considerando a possibilidade de leitura de sinais analógicos e digitais, e as interfaces de visualização disponíveis em função da estrutura do hardware disponível. A escolha do microcontrolador deve levar em consideração a utilização de rede 3G/4G/5G, não nativas no microcontrolador Arduino, sendo necessário a utilização

de um *shield* de conversão, que por muitas vezes, não é facilmente encontrado. Dessa forma, serão priorizados os estudos em torno do microcomputador Raspberry Pi 4 B, modelo já utilizado pela equipe que possibilita a implementação de interfaces de visualização complexas em função da entrada HDMI, possibilitando a ampliação do projeto e desenvolvimento em novas aplicações, e do microcontrolador ESP-12F, microcontrolador de baixo consumo e hardware de tamanho reduzido, ambos com conexão a rede sem fio (Wi-Fi) nativas.

O Raspberry Pi possui uma estrutura mais robusta quando comparado aos microcontroladores comerciais mais comuns, possuindo uma arquitetura que atribui a ele o título de computador, com suporte ao kernel linux e conjunto de instruções ARMv8 (*Advanced RISC Machine*) nativas para seu controlador Quad core 64-bit ARM-Cortex A72 de 1,5 Ghz e 2 Gb de LPDDR4 (*Low-Power Double Data Rate*) RAM. A alimentação da versão 4 modelo B é realizada via conector usb-c, alimentado com 5V e 3A. O Raspberry Pi não possui entradas analógicas, sendo necessária a utilização de conversores analógico-digitais para que seja possível realizar medições dessa natureza. Ele possui suporte para comunicação I2C, UART and SPI e canais PWM a partir de seus GPIOs (LTD., 2019).

O microcontrolador ESP-12F, produzida pela Espressif Systems, por outro lado, possui uma arquitetura menos robusta, de baixo custo, com um núcleo processador Tensilica L106 32 *bit* à 80 ou 160 MHz com 160 KB de memória SRAM. Ele possui tensão de operação de 3,3 V e corrente de à 80 mA. Também possuindo suporte para comunicação I2C, UART and SPI e canais de controle PWM. Além disso, ele conta com um pino para leitura analógica, com suporte para tensão de até 1 V (Shenzhen Ai-Thinker Technology, 2022).

3.2.2 Sensor de velocidade

A informação sobre a velocidade do protótipo possui importância principalmente para o piloto, uma vez que há um limite máximo permitido na pista, auxiliando diretamente no controle do tempo de prova, permitindo o piloto acelerar sem ser penalizado, além de possuir relação direta com o consumo de energia, e portanto, a eficiência do protótipo.

O sensoramento de velocidade possui como princípio a conversão de um movimento mecânico, gerado pelo motor, em um sinal elétrico de forma a estabelecer uma relação direta entre o sinal elétrico obtido com a velocidade de rotação e com a velocidade instantânea do veículo. A medição em veículos urbanos é realizada por meio de um sinal de pulso enviado a unidade de comando eletrônica, sendo este sinal de pulso emitido, normalmente, a partir de um sensor magnético, sensor de efeito *hall* ou sensor de efeito óptico.

Os sensores magnéticos e de efeito *Hall* possuem o mesmo princípio de

funcionamento, sendo acionados e gerando uma diferença de tensão ao serem expostos a um campo magnético. Com a presença de materiais ferromagnéticos fixados a roda é possível emitir uma interrupção e determinar, desse modo, o tempo necessário para a roda realizar uma volta completa (BAHOS et al., 2016). Possuindo o diâmetro da roda, é possível calcular indiretamente a velocidade (Equação 1).

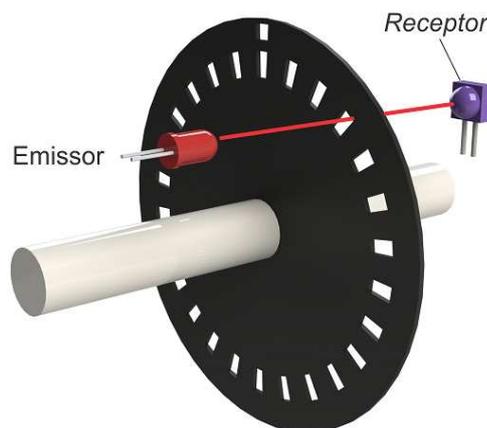
$$Velocidade = 3,6 * \frac{2\pi r}{T_{atual} - T_{anterior}} \quad (1)$$

Em que:

- Velocidade em [km/h]
- 3,6: Fator de conversão de [m/s] para [km/h]
- $2\pi r$: circunferência do pneu do veículo
- $T_{anterior}$: Horário lido na interrupção anterior
- T_{atual} : Horário lido na interrupção atual

A leitura de velocidade através de sensores ópticos ocorre por meio da incidência de luz em um disco com perfurações fixados no eixo do rotor, de forma a captar a luz que passa pelos furos do disco (Figura 6), sendo possível verificar o tempo necessário para realizar uma volta completa no rotor, obtendo indiretamente a velocidade linear do veículo.

Figura 6 – Método de leitura de velocidade via sensor óptico



Fonte: (ALMEIDA, 2017).

Existem ainda novas tecnologias surgindo no mercado para realizar a leitura de velocidade e cadência de bicicletas, com base no funcionamento de sensores giroscópios acoplados diretamente ao cubo da roda das bicicletas, enviando informações via *bluetooth*, buscando trazer facilidade para o acompanhamento da velocidade de atletas e entusiastas do ciclismo. No entanto, estes sensores possuem

custo elevado e pouca informação quanto a sua precisão e repetibilidade disponíveis (GARMIN, 2023).

3.2.3 Sensor de potência

A eficiência do protótipo, medida em [Km/KWh], é atualmente adquirida pela equipe da SEM por meio de um jouímetro instalado em um local pré-estabelecido e previsto no regulamento (SHELL, 2023, p. 28). Os dados coletados pelo jouímetro são verificados ao final da bateria de voltas realizadas na competição, após o protótipo deixar a pista, e de uso exclusivo da equipe da SEM.

Para a obtenção da eficiência do protótipo durante o trajeto, a equipe deve realizar um novo sensoreamento de forma a obter a potência consumida pelo circuito, o horário de realização da coleta e a distância percorrida pelo piloto durante o período de medição. No entanto, medir e exibir a potência instantânea torna-se mais vantajoso para o piloto e para a equipe de suporte, uma vez que poderão tomar decisões em tempo real, auxiliando na redução do gasto de potência na pista, uma vez que o piloto pode atuar e reduzir o consumo de carga de acordo com sua observação, sendo a eficiência calculada ao fim da prova para posterior análise da equipe externa.

A potência (P) em um circuito de corrente contínua pode ser medida diretamente pela multiplicação da corrente (I) e a tensão (V) (Equação 2), de forma que, ao inserir sensores que capturam a medição dessas duas grandezas entre a bateria e o motor do protótipo, é possível verificar a potência instantânea consumida por ele.

$$P_{inst.} = V * I \quad (2)$$

3.2.3.1 Sensor de corrente

Para a determinação de corrente de um circuito, é possível utilizar dois diferentes tipos de sensores, os invasivos, inseridos em série com o circuito que pretende-se medir, sendo necessário a interrupção física do fio de condução, e os não invasivos, sendo mais móveis e maleáveis quanto ao local de inserção para a medição, inserido e removido sem a necessidade de efetuar interrupções físicas no circuito.

Dentro da categoria dos sensores invasivos há uma variedade maior de sensores com relação aos não invasivos. A leitura invasiva permite a determinação da corrente através da leitura da queda de tensão (V) em resistores *shunt*, sendo o valor da resistência (R) conhecida, pela lei de ohm, é possível determinar o valor da corrente (I), que circula no circuito (Equação 3), método utilizado em sensores como o INA219 e o MAX471.

$$I = \frac{V}{R} \quad (3)$$

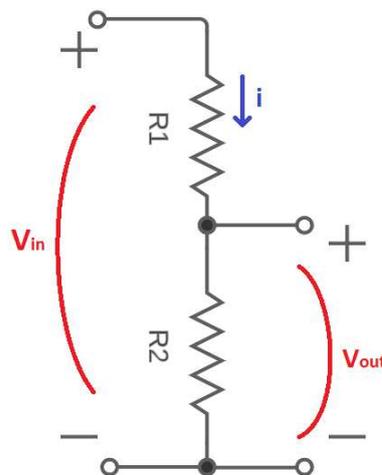
Outra forma de determinar a corrente por meios invasivos é a partir de sensores de efeito *Hall*, como o ACS712 e ACS758, onde o fluxo de corrente que circula no

circuito gera um campo magnético perpendicular, convertido pelo sensor para uma diferença de potencial representada por uma tensão linear equivalente a corrente (ALLEGRO, 2022). O efeito *Hall* também é utilizado na construção de sensores não invasivos, como apresentado na estrutura do SCT-013, que a partir do mesmo princípio, realiza a leitura de corrente alternada em circuito sem necessitar de contato físico com o condutor.

3.2.3.2 Sensor de tensão

A determinação de tensão presente na bateria pode ser realizada de forma direta, utilizando um divisor de tensão (Figura 7). Um circuito divisor de tensão conta com dois resistores em série, a partir da Lei de Ohm é possível determinar uma tensão diretamente proporcional a sua resistência (Equações 4 e 5), desse modo, é possível selecionar valores para os resistores que compõem o circuito divisor de tensão para obter proporções equivalentes (Equação 6) conforme a necessidade e possibilidade de leitura.

Figura 7 – Circuito divisor de tensão



Fonte: Autor (2023).

É possível observar a utilização de circuitos divisores de tensão em sensores comerciais, como no INA219 já citado, que utiliza a verificação de tensão no resistor *shunt* (R_2) em série com a carga (representado por R_1) de forma a obter como saída uma fração fixa da tensão de entrada (V_{in}) diretamente relacionada a tensão de saída (V_{out}) obtida. Além dele, é possível encontrar outros módulos comerciais, sem código de identificação, normalmente com tensões de entrada entre 0 V e 24 V para circuitos de corrente contínua.

$$V_{in} = I * (R_1 + R_2) \quad (4)$$

$$V_{out} = I * R_2 \quad (5)$$

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2} \quad (6)$$

Outra forma de realizar a medição de tensão é a utilizada pelo sensor ZMPT101B, a partir de um pequeno transformador de tensão de 2mA:2mA o sensor é capaz de converter tensão de 0 V a 250 V em redes de corrente alternada. Por essas características, esse sensor é comumente utilizado para medição de tensão em redes residenciais (ABUBAKAR et al., 2017).

3.2.4 Sensor de carga da Bateria

A determinação da carga de uma bateria pode ser realizada medindo diretamente a tensão por ela apresentada, de forma que seja possível atribuir valores de carga adequados para garantir o bom funcionamento do sistema. Deste modo, é possível utilizar os mesmos métodos para medição de tensão apresentados no capítulo de potência.

3.2.5 Sensor de localização

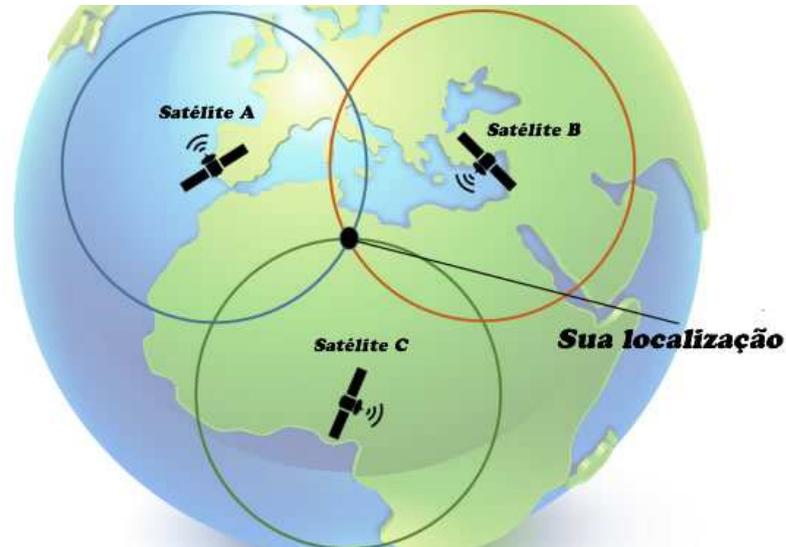
Realizar a verificação geográfica em tempo real do veículo durante a volta pode ser uma grande vantagem quando se trata de sistemas de telemetria desenvolvidos no escopo de competições de eficiência energética. Atribuir os valores obtidos às condições geográficas do carro tornam a análise mais completa, sendo possível verificar o comportamento do protótipo na realização de curvas, subidas e descidas, por exemplo. Além disso, a verificação de geolocalização pode ser utilizada para realizar o mapeamento da pista, auxiliando no preparo e planejamento do piloto para voltas posteriores. Deste modo, é possível realizar estudos e melhorias para aumentar a eficiência em determinados trajetos e condições.

Este trabalho tem por objetivo introduzir a leitura e obtenção da geolocalização via sensoriamento no protótipo, mas não tem por objetivo realizar o tratamento destes dados, atividade destinada para futuras pesquisas. A localização de um robô móvel é definida por THRUN, BURGARD e FOX (2000) como a posição relativa deste robô com relação ao espaço e ambiente a sua volta. A partir dessa definição, a utilização de um sensor LIDAR (Light Detection and Ranging), sensor utilizado pelas duas primeiras equipes a se classificarem no DARPA Challenge em 2007 (HABERMANN, 2016), competição com foco no desenvolvimento de veículos autônomos oferecida pela DARPA (Defense Advanced Research Projects Agency).

Outro método difundido para realizar a geolocalização em pistas, é utilizando diferentes módulos GPS (Global Positioning System) , sendo eles embarcados ou em dispositivos móveis, a verificação de localização ocorre com a conexão de um receptor GPS a pelo menos três satélites, realizando a triangulação necessária para o cálculo de posição do indivíduo (Figura 8). Assim, é possível se obter a localização

em qualquer ponto do globo devido ao conjunto de 24 satélites distribuídos entre seis planos orbitais de mesmo tamanho (MENEZES, 2021).

Figura 8 – Triangulação de localização do GPS a partir dos satélites



Fonte: (Virtuous Tecnologia da Informação, entre 2008 e 2023).

Meneguzzi, Treis e Cendron (2016) apresentam a utilização de um acelerômetro com giroscópio como forma de mapeamento do ambiente. Este mapeamento é possível ao extrair a aceleração em determinado eixo, com um intervalo de tempo conhecido. Com o tratamento desses dados, é possível obter a velocidade e o deslocamento, associados ao ponto inicial, a partir da primeira e segunda integração, respectivamente, sendo teoricamente possível determinar um percurso percorrido por um veículo.

3.2.6 Sensor de temperatura

A temperatura é uma grandeza física, que de um ponto de vista molecular, é diretamente proporcional a energia cinética média das moléculas que permeiam o meio (CINDRA; TEIXEIRA, 2004). Deste modo, pode ainda ser interpretado pelo grau de agitação dessas moléculas e pelo calor dissipado pelo material que elas compõem.

O monitoramento de temperatura é importante para garantir o funcionamento de diversos componentes, podendo ser um fator decisivo para garantir a eficiência dos dispositivos de todo um sistema. Um dispositivo que pode ser diretamente influenciado pela temperatura do ambiente é a bateria, tendo em vista que as células que compõem a bateria do Eficem, INR18650-25R, possui uma faixa de operação entre -20 °C e 75 °C, sendo recomendado a operação abaixo de 60 °C para garantir eficiência do dispositivo (JANG, 2014), esse monitoramento se faz necessário em função da bateria permanecer em uma caixa fechada, por segurança, conforme determina a Shell (2023), e as altas temperaturas registradas anualmente na cidade do Rio de Janeiro, conforme mostrado no relatório fornecido por Weather Spark (2022), cidade-sede da SEM.

Existem diversas formas amplamente difundidas para realizar a leitura de temperatura, como diferentes tipos de termômetros, com funcionamento baseado em expansão de gases, líquidos e dilatação bi-metálica ou por meio de Termo-resistências metálicas. No entanto, para que seja possível realizar o envio destes dados para a equipe externa, e de fato monitorar esses dispositivos ao longo do tempo, é necessário a utilização de um transdutor de temperatura, que seja capaz de converter essa grandeza em um sinal digital equivalente. Mesmo se tratando de transdutores, há diversas possibilidades com diferentes tipos de princípios de funcionamento, como os termômetros de resistência, termistores e diferentes tipos de termopares, habitualmente utilizados em meio industrial (MARTELLI, 2012).

Para delimitar a busca por sensores mais adequados, além de atender o requisito de faixa de operação da bateria, serão priorizados novamente o uso de sensores comerciais com aplicações conhecidas para microcontroladores, sensores de baixo custo e sem contato direto à bateria, facilitando a recorrente remoção do dispositivo na competição.

Dentro das delimitações especificadas, é possível citar o sensor DS18B20, com temperatura recomendada de $-10\text{ }^{\circ}\text{C}$ a $85\text{ }^{\circ}\text{C}$ com acurácia de $\pm 0,5\text{ }^{\circ}\text{C}$, mas atingindo leituras que variam entre $-55\text{ }^{\circ}\text{C}$ e $125\text{ }^{\circ}\text{C}$ com redução na acurácia para $\pm 2\text{ }^{\circ}\text{C}$ (Maxim Integrated Products, 2019). O sensor é capaz de realizar essa medição a partir de um termistor de temperatura, um instrumento capaz de alterar sua resistência elétrica termicamente, por meio de um microcontrolador de 8 bits em seu interior, tratar e transformar em sinais digitais o valor de resistência lido e armazenar na memória *scratchpad*, possuindo uma resolução de 12 bits (CANTÚ, 2013).

Outro sensor aplicável para realizar a leitura de temperatura em um ambiente fechado é da família de sensores de temperatura e umidade DHT, com suporte para temperatura recomendada entre $10\text{ }^{\circ}\text{C}$ a $40\text{ }^{\circ}\text{C}$ com acurácia de $\pm 1\text{ }^{\circ}\text{C}$ no modelo DHT11 (LIU, 2020b) e de $-40\text{ }^{\circ}\text{C}$ a $80\text{ }^{\circ}\text{C}$ $\pm 0,5\text{ }^{\circ}\text{C}$ para o modelo DHT22 (LIU, 2020a). Sua leitura possui o mesmo princípio de funcionamento do DS18B20, realizando a medição a partir de um termistor.

Os dois sensores realizam a comunicação com o microcontrolador por meio do protocolo *One-Wire*, que possui como vantagem a utilização de apenas um cabo de comunicação de dados, em função do processamento interno realizado pelos sensores.

3.2.7 Banco de dados

O primeiro sistema comercial considerado um Banco de dados data de 1960, estruturado com base nos primitivos sistemas de arquivos utilizado na época, denominado Sistema Gerenciador de Banco de Dados (SGBD). Com o passar do tempo as estruturas de dados evoluíram e os modelos de dados passaram por diversas mudanças e atualmente apresentam diferentes estruturas (TAKAI; ITALIANO;

FERREIRA, 2005).

O SGBD é visto como a plataforma que determina as regras e centraliza diversas ferramentas de gerenciamento de dados, de forma que um banco de dados, interface que chega de fato ao cliente final, estará inserida em uma SGBD.

A definição de qual banco de dados utilizar para determinado sistema pode passar por inúmeras etapas de decisões, considerando desde a plataforma em que ela está inserida, protocolos de comunicação utilizados, tipo de estrutura de dados e requisitos de segurança. Deste modo, no contexto deste trabalho serão analisados aspectos como a forma de armazenamento dos dados obtidos, a utilização de plataformas sem custo, facilidade na manutenção do sistema e capacidade de ampliação para que a equipe consiga dar prosseguimento no projeto.

3.2.8 Interface de Visualização

A interface de visualização deve ser abordada por duas diferentes frentes, a visualização local pelo piloto e a realizada pela equipe externa, também em tempo real. A interface de visualização do piloto deverá ser selecionada com base na disponibilidade de conexão do microcontrolador escolhido. Tendo em vista que ao utilizar o Raspberry há uma flexibilidade no uso de *displays*, passíveis de conexões via GPIO ou HDMI (*High-Definition Multimedia Interface*), e limitado a ligações em GPIOs para os demais microcontroladores.

Uma maneira flexível de comunicação visual é a partir do *display* LCD (*Liquid Crystal Display*) 16X2, capaz de apresentar caracteres alfanuméricas dispostos em 2 linhas contendo 16 colunas. Essa tela pode ser conectada diretamente ao GPIOs dos microcontroladores, ou ainda, com a utilização de um módulo de conversão, realizar a comunicação via I_2C , podendo implementar um número maior de telas conforme a necessidade do número de informações a serem visualizadas pelo piloto (PADILHA; ROSA; SOUZA, 2021).

Outra forma versátil para realizar a gestão visual é a partir de telas OLED (*Organic Light-Emitting Diode*), telas compostas por uma matriz de OLEDs individuais que podem ser ativados e desativados de forma coordenada para transmitir informações. Sua comunicação também é realizada via I_2C (Solomon Systech, 2008).

4 METODOLOGIA

Para que seja possível realizar a validação do sistema como um todo, em um primeiro momento se faz necessário a realização de testes nos sensores que não possuem *datasheet*. Além disso, para garantir que não haja interferência entre medições e para que sejam identificados possíveis erros com maior facilidade em determinadas condições de uso para cada um dos hardwares integrados ao sistema, os testes devem ser realizados em circuitos separados.

Para a realização dos testes dos sensores, foram consideradas a confiabilidade das leituras realizadas durante a volta, o tempo em que os sensores deverão permanecer ligados coletando dados e o envio/recebimento dos dados pelo piloto e equipe externa considerando os dados visualizados em tempo real e os armazenados em um banco de dados para posterior análise.

Dessa forma, foram definidas 4 principais frentes para a realização dos testes: (1) análise dos microcontroladores, (2) análises de bancadas individuais para os sensores, validando os valores obtidos com equipamentos de medição reconhecidamente precisos, (3) testes de tempo de execução, mantendo os sensores em funcionamentos por tempo determinado para garantir a confiabilidade da comunicação e realizado com o circuito completo montado, com o intuito de verificar a coleta, transmissão e recebimento dos dados de todos os sensores simultaneamente, e (4) verificação dos dados apresentados nas interfaces em comparação com os transmitidos, verificando se atende o requisito de transmissão em tempo real.

Os testes para tempo de execução levam em consideração que o sistema deve permanecer ligado por pelo menos 25 minutos, tempo limite para a realização das 10 voltas na SEM, no entanto, também deve ser considerada a realização de fora da pista, que podem levar até 30 minutos, com base em testes anteriores realizados pela equipe.

4.1 MICROCONTROLADORES

Apesar da telemetria não influenciar na contagem do joulímetro responsável pela leitura de consumo de energia pelo protótipo durante a realização das voltas, o sistema consome carga da bateria, sendo necessário a medição de tensão e correntes consumidas pelos dois microcontroladores utilizados.

Além disso, foram realizados testes de continuidade de conexão à internet, mantendo os dois dispositivos conectados a uma mesma rede monitorando a conexão por duas horas a partir de funções que armazenam e retornam a estabilidade de conexão ao longo do tempo, cada um baseado em suas principais linguagens de programação, sendo elas, C++ para o ESP8266 e Python para o Raspberry Pi.

4.2 SENSOR DE VELOCIDADE

Uma vantagem da utilização do sensor *hall* quando comparado a utilização de *encoders* para a medição de velocidade de rotação, está no fato de que ele não necessita de uma estrutura grande presa a roda, facilitando a remoção da mesma para manutenção, ou ajustes de estruturas próximas, como os freios, atividades recorrentes durante a competição. O sensor pode ser fixado à carenagem ou ao eixo fixo do protótipo de forma a permanecer próximo o suficiente da roda, sendo necessário apenas fixar um ímã à roda, que podem ser removidos juntos, sem prejudicar sua livre movimentação.

Alguns sensores *hall* são produzidos de forma a facilitar o uso em aplicações para leitura de velocidade. O módulo sensor *hall* A3144 possui uma saída analógica e uma digital, sendo possível realizar diretamente a leitura binária da rotação da roda. Além disso, conta com um amplificador operacional LM393 integrado ao próprio sensor, possibilitando a leitura em baixas tensões.

4.3 SENSOR DE CORRENTE

Tendo em vista o cumprimento dos requisitos de medição de corrente até 20 A, definidos pelo *datasheet* do motor utilizado pelo Eficem, considerando uma corrente contínua de alimentação, torna-se necessário a utilização do sensor ACS712. Em sua versão de 20 A possui uma sensibilidade de 100 mV/A conforme o *datasheet* fornecido pela fabricante Allegro, para uma alimentação de 5 V, no entanto, não é disponibilizada a informação de sensibilidade do sensor para uma alimentação de 3,3 V. Considerando a redução de alimentação de 5 V para 3,3 V, pode-se considerar que a resolução de tensão/corrente gere uma redução proporcional, atingindo, portanto, uma sensibilidade de 0,66 mV/A.

Para o teste de confiabilidade deste sensor utilizando a alimentação reduzida, foi construído um circuito com o sensor e um multímetro digital da politerm modelo POL-76, com capacidade de leitura de até 20 A, em série, realizando simultaneamente leitura que variam de 0 A a 6 A. O circuito utilizado para os testes, apresentavam uma resistência fixa com valor conhecido, realizando a alteração da tensão pelo incremento de 0.1 V a cada leitura realizada, resultando diretamente em um incremento linear na corrente do sistema, sendo possível traçar uma comparação entre os valores lidos pelo instrumento com precisão conhecida e o sensor ACS712. Com os valores obtidos, é possível determinar o erro associado a cada uma das leituras na faixa de operação entre 0 e 6 A, determinando o erro associado à medida de potência informada ao piloto, além de confirmar a sensibilidade para o sensor com uma alimentação de 3,3 V e obter uma previsão de funcionamento para a faixa de 20 A.

4.4 SENSOR DE TENSÃO

Conforme determina o regulamento (SHELL, 2023), a bateria deve possuir uma tensão máxima de 36 V à 48 V, o sensor de tensão mais próximo a essa faixa de operação é um módulo divisor de tensão comercial com entrada de 0 V à 25 V e saída de 5 V. Desse modo, para a sua utilização, seriam necessários dois novos divisores de tensão, um 2:1 para reduzir a entrada da bateria para a metade, e outro 3:2 reduzindo a saída do sensor a 3,3 V para proteção dos pinos de leitura do ESP. Portanto, optou-se pela utilização de apenas um divisor de tensão, com base na atual bateria que a equipe utiliza, possuindo uma tensão máxima de 36V e reduzindo a 3,3 V.

A escolha dos resistores que irão compor o divisor de tensão deve levar em consideração a corrente residual após a passagem pelo circuito, de forma que a leitura no sensor ADS1115 não ultrapasse 10 mA, limite de corrente suportada pelo sensor, conforme datasheet apresentado pelo fabricante Texas Instruments (2018). Conhecendo a tensão de saída de 3,3V e corrente de 10 mA é possível determinar uma resistência mínima de 330Ω necessária para o resistor 2 por meio da Equação 5. Considerando resistores comerciais e já obtidos pela equipe, foi estabelecida a utilização de um resistor de $1 K\Omega$, de forma que, pela Equação 6, foi determinado um resistor de $10 K\Omega$.

Para a validação da escolha dos resistores foram realizadas medições de 0 V a 20 V em um circuito alimentado por uma fonte de bancada POL-16E (POLITERM, 2019) com um multímetro digital da politerm modelo POL-76 conectado em paralelo, realizando leituras com intervalos de 1 V, comparando os valores obtidos por meio do divisor de tensão com um instrumento de laboratório de precisão conhecida. O intervalo de 0 V a 20 V foi determinado com o intuito de manter a escala de operação do multímetro mais próxima a escala de operação da bateria, de forma que haveria uma resolução menor ao utilizar a escala de 0 V a 200 V (POLITERM, 2008).

4.5 SENSOR DE TEMPERATURA

Primeiramente, levou-se em consideração a faixa de operação recomendada pelo fabricante, que possui precisão no funcionamento de $-10 \text{ }^\circ\text{C}$ a $85 \text{ }^\circ\text{C}$, mas suporta temperaturas entre $-55 \text{ }^\circ\text{C}$ e $125 \text{ }^\circ\text{C}$. Essa faixa permite o monitoramento por todo o limite de temperatura da bateria INR18650-25R. Além disso, foi considerada a resolução de 12 bits oferecida pelo sensor, demonstrando precisão suficiente para as medições de temperatura necessárias.

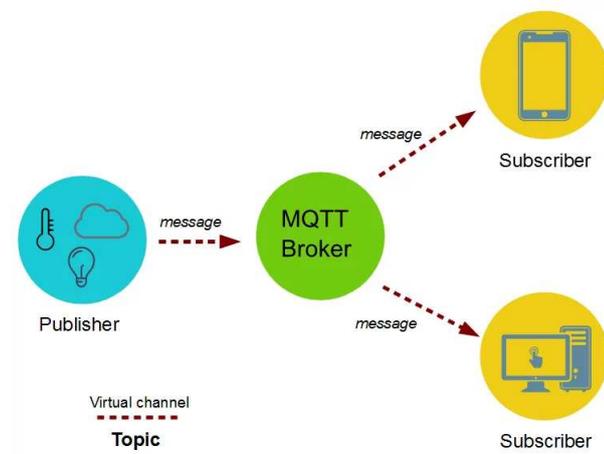
A escolha de uma versão do sensor DS18B20 com carcaça de proteção é essencial para acoplar o sensor próximo à bateria em um ambiente sujeito a vibrações e riscos de impacto uma vez que a própria bateria não permanece totalmente imóvel no seu suporte.

4.6 COMUNICAÇÃO

Tendo em vista a importância do armazenamento para posterior análise, além da visualização desta análise em tempo real pela equipe externa, foram estipulados dois escopos principais, considerando o envio dos dados diretamente a um banco de dados e realizando a visualização em tempo real a partir da busca neste banco, e o envio das informações via protocolo MQTT, onde um *script* é responsável pela inscrição no *broker* e posterior envio dos dados coletados ao banco de dados, e outro *script* paralelo é responsável pela demonstração dos dados para a equipe externa, buscando-as diretamente do *broker* MQTT.

O uso do protocolo MQTT para aplicações de monitoramento de telemetria traz vantagens significativas. O MQTT é um protocolo leve de mensagens projetado especificamente para cenários de Internet das Coisas (IoT) e comunicação entre dispositivos de baixo consumo de energia, apresentando melhor desempenho quando comparado a outros protocolos utilizados nesse mesmo contexto, como o HTTP (SOARES, 2019).

Figura 9 – Comunicação MQTT



Fonte: (Cap Sistema, 2020)

O modelo de publicação/assinatura do MQTT (Figura 9) permite que múltiplos dispositivos enviem informações para um *broker* central, da mesma forma que múltiplos interessados na visualização desses dados podem se inscrever nesse *broker* para recebê-los em tempo real, facilitando na distribuição de informações para equipes externas, permitindo que acompanhem a telemetria em tempo real de qualquer lugar, filtrando apenas as informações de interesse para cada uma das células. Além disso, o MQTT suporta recursos como *Quality of Service* (QoS), que garante a entrega confiável das mensagens, e tópicos com hierarquia, permitindo a organização e filtragem eficiente dos dados (CHEN; KUNZ, 2016). Essas características tornam o MQTT uma opção robusta e escalável para aplicações de monitoramento de telemetria, fornecendo uma

solução flexível e eficaz para o envio e visualização de dados em tempo real.

4.7 BANCO DE DADOS

A definição do banco de dados será realizada pela perspectiva de um administrador do sistema, a partir de um paralelo traçado entre a realidade de uma equipe de competição e os critérios trazidos por Marins e Guarienti (2019) para as funções do administrador da base de dados de gerir: segurança do banco de dados, recuperação de dados, disponibilidade de conexão, suporte a equipe de desenvolvimento e implementação do banco de dados.

- Segurança: aplicada principalmente ao acesso ao banco de dados, garantindo verificação por usuário e limitação do acesso exclusivo dos dispositivos da equipe, uma vez que o ambiente de competição prevê a participação de diversas equipes e a utilização de diferentes redes de acesso públicas.
- Recuperação: a equipe deve ter a capacidade de realizar a verificação dos dados armazenados sempre que necessário e exportar as informações coletadas para a realização de backups e análises.
- Disponibilidade: deve sempre permanecer disponível para a realização da prova e eventuais testes, garantida por visualização de conexão pela equipe externa em tempo real.
- Suporte a equipe de desenvolvimento: deve permitir fácil integração com softwares para análise de dados e coleta de dados via Python, linguagem comum de desenvolvimento e tratamento de dados.
- Implementação: a estruturação do banco de dados deve ser simples e modular, em função da manutenção e ampliação de desenvolvimento realizada pelos membros futuros da equipe.

5 RESULTADOS

Neste capítulo são apresentados os resultados encontrados nas realizações dos testes de cada um dos sensores selecionados, conforme a metodologia estabelecida no Capítulo 4, bem como os procedimentos adotados, desafios e correções realizadas a partir dos testes. Posteriormente será indicado um panorama geral do projeto definido e apresentação dos códigos, circuitos e conclusões.

5.1 MICROCONTROLADORES

Para a realização dos testes foram utilizadas as linguagens C, C++ e Python, conforme a necessidade do procedimento realizado. Os sensores foram testados e validados com o ESP-12F que utiliza o, utilizando a IDE (*Integrated Development Environment*) do Arduino como plataforma de desenvolvimento.

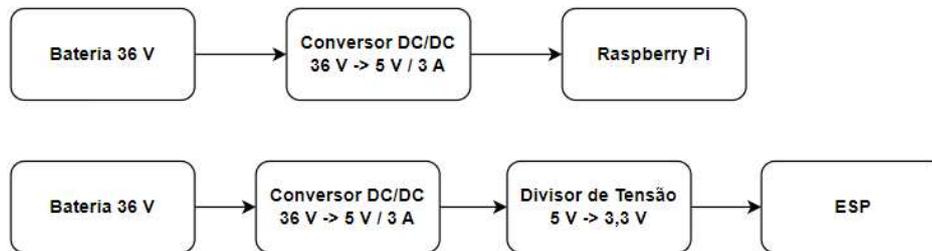
O Raspberry Pi utilizado possui um sistema operacional (SO) Raspberry Pi OS instalado. Esse sistema é baseado na distribuição linux Debian e é recomendada pelo fabricante, pois seu sistema é otimizado para o hardware do Raspberry (Fundação Raspberry Pi, entre 2012 e 2023). Foi instalada a versão básica, não possuindo pacotes e programas extras pré-instalados, apenas os já possuídos pelo sistema padrão.

5.1.1 Análise de Carga

O primeiro teste realizado com os controladores selecionados, ESP-12F e Raspberry Pi 4 model B, foi a determinação do consumo de cada um deles, considerando apenas a conexão da alimentação, sem sofrer influência do consumo de qualquer dispositivo extra como sensores e periféricos. Os dois dispositivos testados apresentavam pleno funcionamento quando utilizados e alimentados por meio de fontes padrões, sendo a fonte de alimentação original do Raspberry, com 5 V e 3 A de capacidade, e a alimentação via cabo micro USB no ESP. Para realizar essa comparação, os dois controladores foram alimentados diretamente nos pinos pela bateria utilizada pelo EFICEM, de forma que, nos dois testes a bateria se encontrava com sua carga totalmente carregada.

Por se tratar de uma bateria de 36 V, se fez necessária a utilização de um conversor DC/DC para reduzir a tensão à 5V, este conversor, possui capacidade de alimentação de até 5 A. Para garantir as mesmas condições, o conversor de 5 V foi utilizado no circuito para realizar o teste dos dois controladores, sendo necessária a utilização de um divisor resistivo de tensão extra para o microcontrolador ESP-12F, reduzindo a alimentação de 5 V para 3,3 V, conforme expresso na Figura 10.

Figura 10 – Diagrama de blocos para alimentação dos controladores



Fonte: Autor (2023).

Inicialmente, foram realizadas leituras alimentando os controladores diretamente pelos pinos, de forma a garantir que o Raspberry Pi recebesse 5 V de tensão de entrada. Para a alimentação pelos pinos, o Raspberry Pi não conseguiu inicializar, fornecendo uma corrente ($I_{Rasp-pinos}$) máxima de 0,68 A ao microcomputador e reiniciando ao tentar realizar *boot* no sistema. Conforme determina o fabricante Ltd. (2019), a principal interface de alimentação é via conector USB-C disponibilizado pelo hardware. Deste modo, foi adquirido um conector adaptador usb tipo C (Figura 11, com dois fios para alimentação, para que fosse possível conectar a bateria via módulo USB-C.

Figura 11 – Conector USB tipo C utilizado nos testes



Fonte: Autor (2023).

Para a alimentação via conector, a corrente de alimentação do equipamento conseguiu atingir o máximo ($I_{Rasp-USB-C}$) de 1,12 A, no entanto, o dispositivo por todo o período emitiu um alerta de alimentação insuficiente, ferramenta disponibilizada pelo

próprio software de controle de funcionamento (*firmware*) do equipamento. Nesse contexto, dois comportamentos foram predominantes, um realizando desligamentos forçados ao tentar executar algum aplicativo/programa, e outro apresentando lentidão e mal funcionamento em geral, se tornando muito difícil utilizar alguma funcionalidade. Vale ressaltar que todos os fios utilizados foram substituídos durante a realização dos testes, reduzindo a possibilidade de problemas elétricos externos ao sistema.

Os testes foram realizados novamente alimentando o microcomputador pelos pinos e conversor USB tipo C, por meio de uma fonte de bancada POL-16E, mas os resultados se assemelham à alimentação realizada por meio da bateria. Por fim, foram desabilitadas algumas funcionalidades do Raspberry Pi com o intuito de reduzir o consumo do equipamento, desabilitando protocolos de conexões de qualquer tipo, desabilitando a inicialização do *bluetooth* ativada por padrão e desligando os LEDs (*Light Emitting Diode*) de sinalização, também padrão do dispositivo, sem gerar alterações perceptíveis às leituras, apresentada por $I_{Rasp-Config}$ no Quadro 1.

Quadro 1 – Medições de Carga do Raspberry Pi e ESP

	Corrente máxima [A]	Operação
$I_{Rasp-pinos}$	0,68	Não conseguiu inicializar o sistema, reiniciando constantemente.
$I_{Rasp-USBC}$	1,12	O sistema apresentou lentidão e falha na operação, além de reinicializar ao tentar executar alguns programar.
$I_{Rasp-Config}$	1,10	O sistema apresentou lentidão e falha na operação, além de reinicializar ao tentar executar alguns programar.
I_{ESP}	0,18	O dispositivos não apresentou nenhum problema, funcionando corretamente por todo teste.

Fonte: Autor (2023).

Para a medição de corrente do ESP-12F não ocorreu nenhum problema, atingindo uma corrente máxima (I_{ESP}) de 0,18 A, o dispositivo apresentou bom funcionamento e operou normalmente.

5.1.2 Teste de conexão

Para a realização dos testes de conexão a rede de internet, foram utilizados dois diferentes códigos, considerando as ferramentas disponíveis e plataformas de desenvolvimento padrão de cada um dos controladores, foram considerados a utilização da linguagem Python para o Raspberry Pi (Apêndice A) e a linguagem C e C++ a partir da IDE do Arduino para a programação do ESP (Apêndice B).

A utilização de duas diferentes linguagens e plataformas para a programação dos controladores nos impedem de realizar uma análise objetiva acerca dos resultados obtidos, tendo em vista as diferenças lógicas, estruturais e no próprio funcionamento de execução da chamada de funções e do código. No entanto, foi mantida uma mesma proposta para o funcionamento dos dois códigos, sendo possível traçar paralelos e analisar alguns dos pontos observados.

O princípio geral dos códigos se inicia com a conexão dos controladores na rede de internet, neste caso, utilizando uma rede 4G de celular, conforme a disponibilizada na competição para a realização da comunicação. Uma vez que os dois dispositivos estão conectados na mesma rede, é realizada uma verificação única de conexão com a internet, confirmando as credenciais utilizadas.

Posteriormente é aberta uma conexão com um *socket* TCP simultaneamente nos dois códigos, solicitando conexão a um mesmo servidor. É enviada uma mensagem e a partir do retorno é confirmada que o dispositivo está conectado à rede de internet. Essas verificações ocorrem a cada 5 segundos nos dois códigos, quando uma conexão é bem sucedida e é recebido o retorno do servidor, um *ping* é contabilizado, quando a conexão com o servidor externo é perdida, é iniciada uma contagem para verificar o tempo em que esteve desconectado, estes valores são armazenados em um histórico de registro (Log) para cada um dos dispositivos. Vale ressaltar que os testes foram realizados mantendo os dois dispositivos a uma mesma distância de 60 cm do ponto de conexão, exercendo um maior estresse na rede em um primeiro momento (Teste 1), utilizando a rede para outras funções e distanciando as conexões, e mantendo a rede estática e próxima em um segundo momento (Teste 2).

Quadro 2 – Teste de Conexão dos Microcontroladores

	Teste 1		Teste 2	
	Raspberry Pi	ESP	Raspberry Pi	ESP
Tempo de conexão [min]	73	73	179	178
Pings Máximos	876	876	2148	2136
Pings Realizados	830	826	2110	2100
n° Conexões perdidas	20	21	3	5
Tempo em Conexão perdida [seg]	72	56,96	31	7,88

Fonte: Autor (2023).

Considerando as leituras com intervalo de 5 segundos, e sabendo o tempo de realização do teste (Tempo de conexão), é possível determinar um valor máximo de *pings* (Pings Máximos) que poderão ser realizados. Este não deve ser o número real, mesmo que não haja problemas de conexão, em função do tempo de processamento de algumas das atividades realizadas no código, como abrir e inserir informações nos arquivos de Log ou tempo em que os controladores aguardam resposta do

servidor conectado. Deste modo, o número de conexões perdidas e o tempo em que o controlador permaneceu desconectado, tornam-se valores comparativamente mais interessantes. De todo modo, é possível observar pelos dados coletados, que o número de pings realizados e conexões perdidas, dos dois dispositivos, permaneceram próximos, sendo possível associar o distanciamento da fonte de rede de internet, bem como instabilidades da própria rede 4G, como fatores diretamente ligados às quedas de conexão, não necessariamente exclusivas de problemas de capacidade de hardware dos dispositivos de estudo.

No entanto, é possível notar um tempo de desconexão maior associado ao Raspberry. As formas como a gestão de tempo é lida em cada Log é diferente. Para o código do ESP é realizada a leitura em mili segundos a partir das ferramentas disponibilizadas pelo dispositivo. Os Logs do raspberry são com base na biblioteca de data disponibilizada por ele, obtendo a resolução mínima de 1 segundo. Além disso, o ESP possui um delay de 1 segundo entre as tentativas de reconexão, somado ao valores extras decorrentes do não arredondamento, era possível imaginar um tempo de desconexão maior para ele. No entanto, o Raspberry apresentou tempos médios (Apêndice K) para reconectar maiores com relação ao ESP (Apêndice J), principalmente no testes de pouca turbulência, dois períodos de queda por 15 segundos, como poder ser verificado nos Logs na íntegra .

5.2 VELOCIDADE

Para que fosse possível realizar o teste e validação da utilização do sensor Hall para medição de velocidade, uma roda foi montada em um suporte para que pudesse girar livremente, e nela foi acoplado um imã, de forma que ao girar a roda, o imã passe próximo o suficiente para ser detectado pelo sensor e ative o sinal em seu pino digital, garantindo um pulso confirmando a realização de uma volta completa realizada pela roda.

Para a obtenção de velocidade do protótipo, inicialmente foi realizada a programação (Apêndice C) do pino que receberia o sinal do sensor *Hall* (*hallCount*) para identificar apenas sua borda de subida, e neste pino foi configurada uma interrupção, garantido sua leitura. Para atender ao requisito de leitura de velocidade, inicialmente foi desenvolvida uma função que realiza o calculo de rotações por minuto (*rpm*) com base na leitura do sensor em relação ao tempo levado para realizar uma volta completa (*elapsedTime*), conforme visto na Equação 7.

$$RPM = hallCount * 60.0 * \frac{1000.0}{elapsedTime} \quad (7)$$

O tempo para a realização de uma volta completa é obtido por meio da função *millis()* do ESP, um recurso capaz de retornar o tempo de operação do ESP, desde

o momento em que ele foi ligado, em milissegundos. Por esse motivo, `elapsedTime` é convertido para segundos e multiplicada por 60 para obter o tempo de rotação em minutos. Para comparar os obtidos por meio do sensor Hall e calculado via ESP, foi acoplado ao sistema um tacômetro digital TDV100 (VONDER, 2018), de forma a realizar a leitura simultaneamente ao sensor *hall*, sendo possível realizar uma comparação direta entre os valores medidos, conforme a Tabela 1. Após a realização dos testes considerando uma rotação por minuto mais alta e próxima do uso comum, foi realizado um novo teste em baixa rotação para verificar o comportamento e confiabilidade em situações próximas da parada ou arrancada, os valores obtidos podem ser vistos na Tabela 2.

Tabela 1 – Medições realizadas para determinação de RPM alto

Tacômetro	Sensor Hall	Erro
18	23,09	5,09
89,4	89,25	-0,15
94,2	94,01	-0,19
98,8	98,4	-0,4
106,9	100,17	-6,73
116,2	116,28	0,08
127,7	124,29	-3,41
136,2	126,85	-9,35
150,7	151,23	0,53
159,4	159,79	0,39
160,3	161,4	1,1
160,7	160,64	-0,06
161,7	161,18	-0,52
172,2	169,13	-3,07
180,2	172,79	-7,41
188,6	188,24	-0,36
189,4	189,72	0,32
191,1	190,33	-0,77
192,1	192,31	0,21
194,4	193,7	-0,7

Fonte: Autor (2023).

Tabela 2 – Medições realizadas para determinação de RPM baixo

Tacômetro	Sensor Hall	Erro
13,2	13,28	0,08
16,1	16,43	0,33
16,4	16,55	0,15
16,5	16,74	0,24
18,0	18,05	0,05
18,4	18,45	0,05
26,4	27,03	0,63
26,4	26,75	0,35
26,5	26,58	0,08
26,6	26,49	-0,11
26,7	26,91	0,21
26,7	26,71	0,01
27,2	27,32	0,12
27,3	27,32	0,02
27,3	27,35	0,05

Fonte: Autor (2023).

Os dados foram reorganizados e apresentados em ordem crescente selecionando um a cada três dos dados coletados para compor a visualização, é possível verificar todos os dados coletados no Apêndice N. Durante o teste foi possível observar as maiores diferenças de leitura estavam associadas aos momentos de maior

aceleração/desaceleração, associando os maiores valores de erro à pequena diferença de tempo entre a leitura dos dois dispositivos, mantendo um valor próximo se tratando de rotações mais estáveis, obtendo a partir de todas as leituras um erro absoluto médio de 0,691. O fato da roda estar montada em um suporte, removendo o atrito do sistema, faz com que a rotação possa variar de forma mais rápida do que a realmente encontrada no veículo.

Durante os testes realizados em baixa rotação foram verificados momentos em que a leitura não era realizada pelo tacômetro, sendo desconsiderada na contabilização dos dados por não ser possível determinar se eram erros de leitura ou uma medição igual à anterior, mantendo dessa forma, um máximo de uma repetição da leitura. Considerando rotações menores a variação entre os valores da aceleração/desaceleração foram menores, gerando um erro médio absoluto de 0,131, consideravelmente menor com relação ao Teste 1, e possivelmente mais próxima da realidade, uma vez que a aceleração e desaceleração eram menores.

Com o valor da rotação por minuto definida, é possível determinar a velocidade linear [Km/h] do veículo, tendo conhecimento da circunferência, obtida a partir do raio (r) da roda em relação ao tempo de rotação já conhecida, conforme Equação 8.

$$Velocidade = 3.6 * RPM * (2 * \pi * r) / 60; \quad (8)$$

No código desenvolvido foi utilizado a leitura do sensor como condição para a realização dos cálculos, realizando portanto, uma leitura de velocidade a cada uma volta realizada pela roda. A partir do maior valor de erro médio absoluto obtido, de 0,691, e considerando uma circunferência de 1,5708 m para a roda utilizada, é atribuída à velocidade uma variação máxima de 0,4 Km/h da leitura real.

5.3 CORRENTE - ACS712

Para os dois controladores considerados, torna-se necessário a utilização de um conversor analógico-digital (AD) para realização das leituras dos sensores de tensão e corrente, grandezas contínuas. Deste modo, foi determinado a utilização do ADS1115 no projeto, um conversor AD capaz de transmitir os dados coletados por meio do protocolo I_2C , suportado por ambos dispositivos.

Deste modo, o desenvolvimento do código para obtenção da leitura do ACS712 (Apêndice D) foi realizada por meio da biblioteca `Adafruit_ADS1X15.h` (Adafruit Industries, 2023) disponibilizada pela Adafruit para leitura dos pinos do ADS1115. Considerando o sensor ACS712 - Modelo EL-20, Allegro (2022) determina que sua faixa de operação será entre -20 A e 20 A, considerando a leitura nos dois sentidos de operação da corrente. Considerando uma alimentação de 3,3 V no sensor, serão disponibilizados 1,65 V para a faixa de operação entre 0 A e 20 A, de forma que a sensibilidade mínima considerada pelo sensor alimentado por 5V é de 66 mV / A.

Deste modo, para a conversão da tensão lida no pino do conversor para uma corrente equivalente foi considerado a Equação 9.

$$Corrente = \frac{V_{Lido} - 1,65}{0,066} \quad (9)$$

As medições ocorreram alterando o valor de corrente em intervalos de 0,2 A, até um limite de 5 A determinado pela fonte de bancada, de forma que houvessem diferenças menores para os primeiros valores, em função do maior erro associado a essas medições em função da influência que uma pequena alteração de tensão pode gerar no cálculo final. Os principais valores obtidos podem ser visualização na Tabela 3. Ao observar todas as leituras (Apêndice O), é possível verificar um intervalo de operação em que o sensor se mostra mais preciso, neste caso, parametrizado em torno dos 4 A. Durante os testes foi possível confirmar a resolução de 0,066 V/A, de forma que ao alterar essa relação na Equação 9 é possível alterar a faixa de maior estabilidade na leitura do sensor, sendo factível movimentar essa operação ao valor nominal do motor para trabalhar com valores mais precisos, de forma análoga a uma calibração do dispositivo.

Tabela 3 – Medições realizadas pelo sensor ACS712

Corrente [A]		
Amperímetro	ACS712	Erro
0,00	0,04	-0,04
0,10	0,15	-0,05
1,00	1,03	-0,03
2,00	2,01	-0,01
3,00	3,01	-0,01
4,00	4,00	0,00
5,00	4,98	0,02

Fonte: Autor (2023).

5.4 TENSÃO - DIVISOR DE TENSÃO

A medição de tensão foi realizada em conjunto com a medição de corrente, por meio do conversor analógico-digital ADS1115 (Apêndice D). Para o cálculo realizado no programa, foi determinada a relação entre as resistências conforme o valor Comercial dos resistores (Ω Comercial), sendo eles, 1 K Ω para R_2 e 10 K Ω para R_1 , portanto, a tensão de entrada, neste caso análogo à tensão que a bateria está fornecendo, pode ser obtida ao isolar (V_{in}) na Equação 6, gerando uma relação linear entre o valor lido e tensão fornecida pela bateria conforme Equação 10, resultando na coluna de tensão

lida para Ω Comercial.

$$V_{in} = \frac{V_{Lido}}{0,090909090909} \quad (10)$$

Quadro 3 – Medições realizadas a partir do circuito divisor de tensão

Tensão				
Referência Voltímetro [V]	Ω Comercial		Ω Corrigida	
	[V]	Erro	[V]	Erro
0,00	0,00	0,00	0,00	0,00
2,00	2,01	0,01	2,00	0,00
4,00	4,02	0,02	4,00	0,00
6,00	6,03	0,03	6,00	0,00
8,00	8,05	0,05	7,99	-0,01
10,00	10,05	0,05	9,99	-0,01
12,00	12,07	0,07	12,00	0,00
14,00	14,08	0,08	14,00	0,00
16,00	16,10	0,10	16,00	0,00
18,00	18,12	0,12	18,00	0,00
20,00	20,15	0,15	20,00	0,00

Fonte: Autor (2023).

A partir das leituras foi possível observar um valor de erro crescente com relação linear aos valores de referência. Dessa forma, é possível determinar uma nova relação de resistência, considerando valores mais próximos dos reais aos resistores utilizados, utilizando os novos valores obtidos e a Equação 5 é possível encontrar um novo valor para R_2 , determinando também, um novo valor de R_1 por meio da Equação 4. A partir dela foi determinada a nova relação entre as resistências de $0,0903 \Omega$. Ao realizar novamente as leituras (Ω Corrigida), foi possível levar o erro próximo a 0, conforme apresentado na coluna final do quadro 3.

5.5 LOCALIZAÇÃO - GPS-NEO6

Os dados obtidos por meio do sensor GPS possuem grande influência de construções e estruturas quando suas medições são realizadas em seu interior, sendo necessário realizar o teste em lugares abertos. Os pontos definidos seguiram a proposta de se assemelhar ao cenário encontrado na pista onde a competição da SEM ocorre, deste modo, pontos abertos próximos a construções. Para os pontos selecionados, foram realizadas 30 leituras buscando um padrão e repetibilidade dos valores lidos, realizando uma média e comparando com os valores de latitude e longitude coletados a partir da ferramenta Google Maps (Google LLC, 2023) para o ponto analisado.

Considerando todos os dados possíveis para se obter com o módulo GY-NEO6m, foi desenvolvido um código (Apêndice E) utilizando as bibliotecas *TinyGPS++.h* e *TinyGPSPlus.h*, de forma a tratar a mensagem recebida via satélites e realocá-la em posição (lat, lng, altitude), data (ano, mês, dia) e hora (hour, minute, second). Das 30 (Apêndice L) leituras realizadas em cada um dos pontos selecionados, foram extraídas as médias dos valores obtidos através do sensor, e realizado a conferência do erro em metro, a partir da ferramenta de medição de distância do Google Maps (Google LLC, 2023). Os valores obtidos podem ser visualizados na Tabela

Tabela 4 – Comparativo entre dados coletados por sensor GPS e via Google Maps

Ponto	Google Maps		Média Valores Lidos		Erro [m]
	Latitude	Longitude	Latitude	Longitude	
1	-26.236393	-48.885917	-26.236389	-48.885948	3.10
2	-26.236265	-48.885962	-26.236225	-48.885977	4.60
3	-26.235069	-48.885142	-26.235113	-48.885104	6.23
4	-26.234742	-48.884935	-26.234752	-48.884958	2.26
5	-26.234862	-48.885115	-26.234864	-48.885078	3.68
6	-26.235279	-48.884698	-26.235257	-48.884635	6.78
7	-26.235603	-48.884836	-26.235629	-48.884868	4.17
8	-26.234701	-48.884229	-26.234697	-48.884222	1.09
9	-26.234689	-48.884790	-26.234670	-48.884797	2.27
10	-26.235161	-48.885618	-26.235207	-48.885614	4.97

Fonte: Autor (2023).

5.6 TEMPERATURA - DS18B20

As características apresentadas pelo sensor DS18B20 garantem que a equipe possa monitorar a temperatura dentro da faixa de operação correta, mantendo a segurança dos dispositivos e sensores do sistema, contribuindo para o bom funcionamento e desempenho do veículo elétrico durante as competições.

Considerando a utilização do ESP-12F como microcontrolador, foi desenvolvido um código utilizando a IDE do Arduino (Apêndice F). Esse código baseia-se nas bibliotecas *OneWire.h* e *DallasTemperature.h*, que fornecem as funcionalidades necessárias para a comunicação e leitura do sensor de temperatura. Utilizando o pino D4 (GPIO 2) para realizar a comunicação via One-Wire com o sensor de temperatura, o programa foi escrito para permitir a adição de mais sensores para monitoramento de temperatura de mais dispositivos no futuro.

5.7 COMUNICAÇÃO - MQTT

Para fins didáticos, e de manutenção, o desenvolvimento do código de comunicação com o *broker* MQTT e da conexão Wi-Fi foram realizados juntos, de forma

a centralizar a comunicação em um único teste, bem como permitir a comunicação MQTT com *broker* de testes gratuito online (Apêndice G). Em um mesmo código foi realizada a publicação e inscrição no mesmo tópico, sequencialmente, para que seja possível realizar a postagem e inscrição para visualização conforme necessidade. No entanto, este código irá compor o controlador, realizando apenas a publicação nos tópicos do *broker*, possuindo outros dispositivos para realizar a inscrição e visualização destes dados.

5.8 BANCO DE DADOS - MONGODB ATLAS

Após análise e considerando as etapas e critérios mencionados, optou-se pela escolha do banco de dados MongoDB Atlas para o armazenamento dos dados obtidos no sistema de telemetria. O MongoDB Atlas é uma solução de banco de dados não relacional, evitando a necessidade de construção de tabelas e relações que restringiriam a expansão e manutenção por qualquer futuro membro. Além disso, este é um banco de dados presente na nuvem, que oferece escalabilidade, flexibilidade e facilidade de implementação, e dispensa a necessidade de um hardware físico da equipe, uma grande vantagem se tratando de um contexto de competição. Essa escolha é possível e justificada devido ao baixo armazenamento necessário, à facilidade de manutenção do sistema e à capacidade de expansão, permitindo que a equipe prossiga com o projeto sem a necessidade de estudo aprofundados em bancos de dados (PERA, 2021).

A integração entre o sistema de telemetria e o banco de dados MongoDB Atlas foi realizada por meio do desenvolvimento de um código em Python. Esse código permite a inscrição em um broker MQTT, a coleta dos dados provenientes dos sensores e o envio desses dados para o banco de dados MongoDB Atlas (Apêndice H). Essa abordagem garante uma comunicação eficiente e contínua entre o sistema de telemetria e o banco de dados, possibilitando a captura e o armazenamento em tempo real das informações relevantes para análises posteriores.

A utilização do banco de dados MongoDB Atlas traz benefícios significativos para a equipe, uma vez que proporciona um ambiente seguro com conexão validada apenas aos dispositivos previamente autorizados, confiável e disponível em tempo real. Com a estruturação simples e modular do banco de dados, facilita-se a manutenção e ampliação do desenvolvimento realizado pelos membros futuros da equipe. Além disso, a integração do banco de dados com softwares de análise de dados e a coleta de dados via Python promovem a eficiência na análise e tratamento das informações, agregando valor ao processo de tomada de decisões e ao desempenho da equipe durante as competições.

5.9 INTERFACES DE VISUALIZAÇÃO

A interface de visualização desempenha um papel fundamental no sistema de telemetria, permitindo que os dados coletados sejam apresentados de forma clara e acessível tanto para o piloto como para a equipe externa. Neste contexto, foram consideradas duas frentes distintas: a visualização local pelo piloto e a visualização remota pela equipe externa.

Para a visualização local pelo piloto, levando em consideração a utilização do microcontrolador ESP-12F e a necessidade de simplicidade na exibição dos dados, foi selecionada a tela LCD 16x2 (Figura 12). Essa escolha baseou-se na disponibilidade de conexão direta ao GPIO do microcontrolador, garantindo uma comunicação direta e confiável (Apêndice I). A tela LCD 16x2 é capaz de apresentar caracteres alfanuméricos em 2 linhas com 16 colunas, sendo utilizado duas telas para a disposição de todos os dados e suas respectivas identificações, proporcionando uma interface clara e legível para o piloto. Além disso, a utilização dessa tela localmente elimina a necessidade de passar os dados pelo *broker* MQTT ou servidor de banco de dados online, evitando a introdução de atrasos (*delays*) na entrega das informações.

Figura 12 – Tela de visualização do piloto

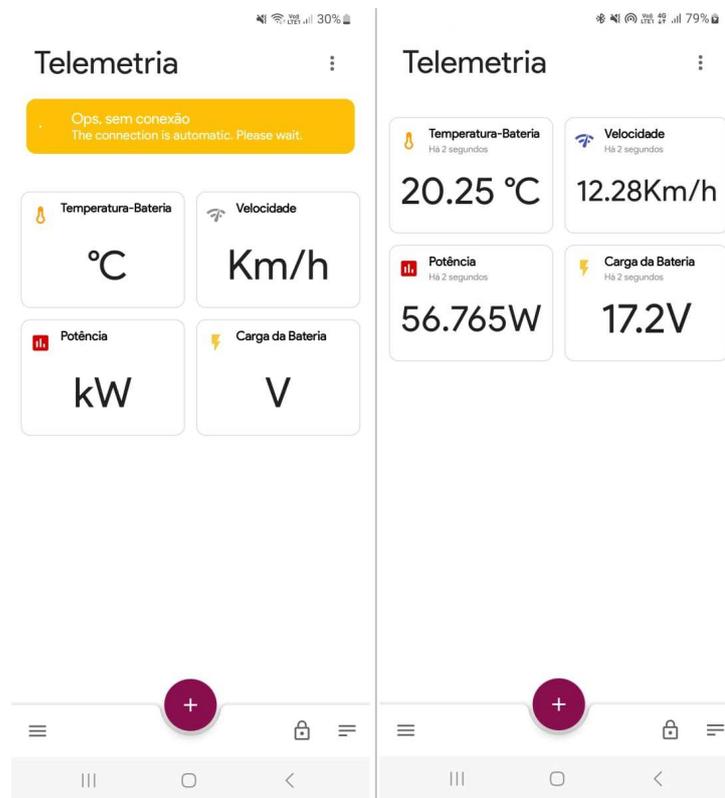


Fonte: Autor (2023).

Para a visualização remota pela equipe externa, foram consideradas duas opções: desenvolver uma interface personalizada por meio de uma linguagem de programação ou utilizar um software pronto terceirizado. Após análise, optou-se pelo uso do aplicativo MQTT Dashboard (Figura 13). Esse aplicativo permite a conexão com um servidor MQTT e a inscrição de cada atributo de interesse ao seus tópicos correspondentes, dependendo de pouca configuração, possibilitando a visualização em tempo real das informações relevantes. Essa escolha foi baseada na praticidade e na eficiência do aplicativo, que oferece uma interface intuitiva e de fácil configuração, atendendo às necessidades da equipe externa de forma eficaz.

Para a escolha do aplicativo MQTT Dashboard como interface de visualização remota pela equipe externa, um dos principais fatores considerados foi a sua flexibilidade e acessibilidade. Esse aplicativo permite que seja utilizado de qualquer

Figura 13 – Dashboard de visualização da equipe externa



Fonte: Autor (2023).

ponto da competição, desde que haja conexão com a rede local, o que proporciona uma grande vantagem em termos de mobilidade e conveniência.

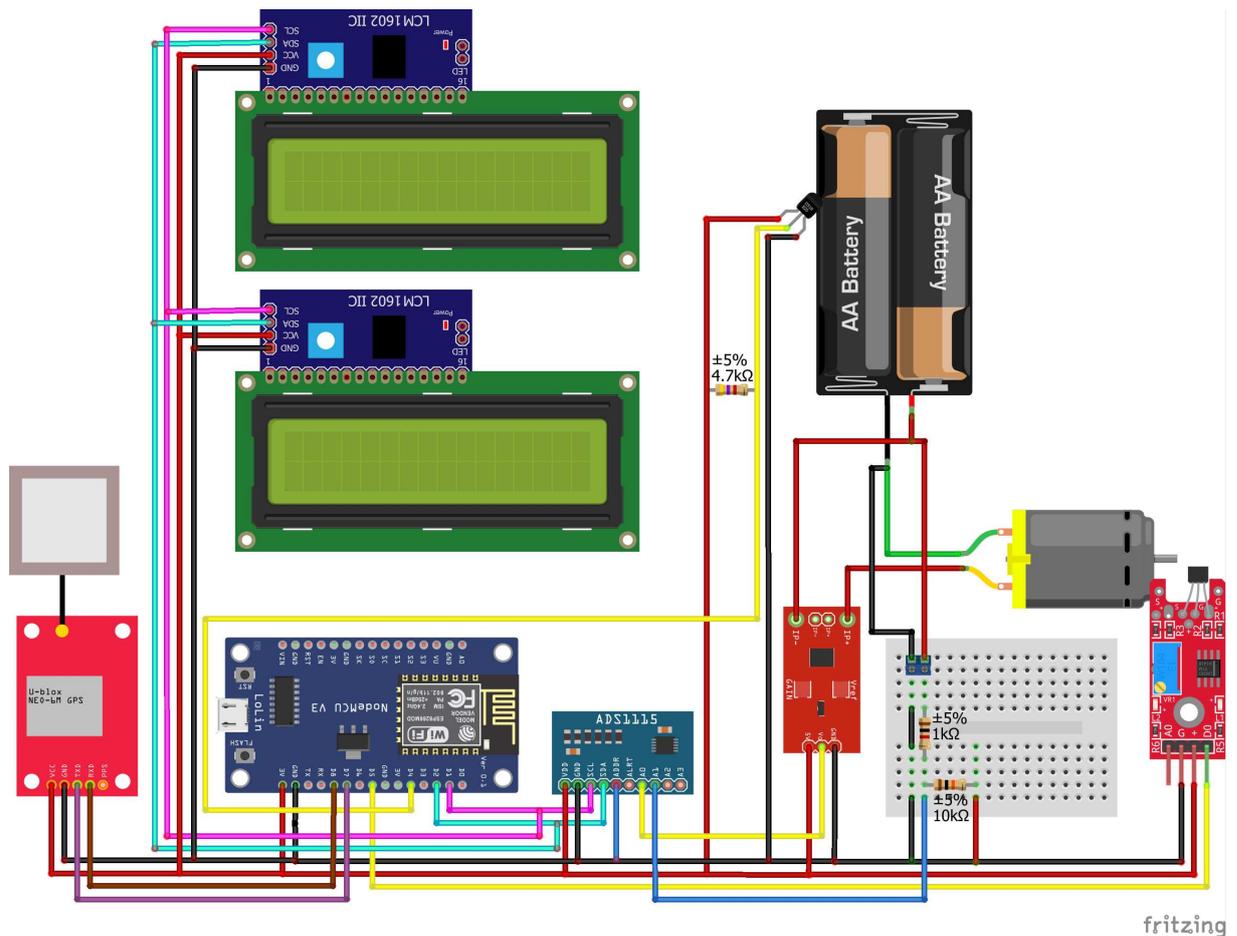
Além disso, o MQTT Dashboard oferece a possibilidade de inscrição de mais de um usuário, o que significa que vários membros da equipe podem ter acesso às informações simultaneamente e visualizar os dados de interesse de forma personalizada. Essa capacidade de suportar múltiplos usuários e diferentes visualizações é extremamente valiosa em uma competição, onde a colaboração e a análise em tempo real são fundamentais para tomar decisões estratégicas.

5.10 MONTAGEM DO SISTEMA DE TELEMETRIA

Conforme apresentado nos resultados e possuindo a definição de todo o hardware e software utilizados no projeto é possível executar a construção do sistema como um todo, com as definições de pinos necessários conforme os protocolos de comunicação utilizados por cada sensor, definindo as bibliotecas necessárias e características físicas e elétricas apresentadas pelo projeto. O sistema completo pode ser observado na Figura 14.

Conforme apresentado nos testes com os microcontroladores, o Raspberry Pi não apresentou desempenho satisfatório realizando a alimentação do sistema por meio

Figura 14 – Sistema de Telemetria para Protótipo de Carro Elétrico



Fonte: Autor (2023).

de seus pinos e adaptadores, a partir da alimentação de carga disponível pela equipe Eficem. Desse modo, o tratamento dos dados obtidos pelos sensores, bem como a comunicação será realizada por meio de um ESP-12F conectado à rede de internet de um telefone móvel. A comunicação será realizada via protocolo MQTT, de forma que os dados coletados serão enviados a um *broker*, e dois clientes fica responsável por realizar as inscrições nos tópicos de interesse, um realizando a visualização em tempo real dos dados coletados por meio do MQTT Dashboard, e outro enviando essas informações a um banco de dados com armazenamento em nuvem, o MongoDB Atlas, conforme apresentado pelo escopo de projeto na Figura 3. Além da utilização de telas LC 16x2 para visualização das informações de forma local para o piloto.

Para os sensores, foram definidas a utilização do ACS712 e um divisor de tensão para determinar a potência instantânea consumida pelo circuito de carga, considerando o consumo realizado pelo motor e pelo controlador. Além disso, a partir da tensão obtida pelo divisor de tensão, é determinada a carga da bateria. Um sensor de temperatura DS18B20 acoplado junto à caixa da bateria realizara o monitoramento da temperatura em seu interior. Para realizar a medição de velocidade do protótipo está

sendo utilizado um sensor *Hall* capaz de captar um ímã acoplado a roda, realizando a medição de rotações e a partir do tempo necessário para a rotação e do diâmetro da roda, determinar a velocidade linear do protótipo. Além disso, está sendo realizada a leitura de localização do protótipo a partir de um sensor GPS GY-Neo6m, possibilitando futuros trabalhos voltados ao mapeamento da pista e mapeamento de pontos de maiores consumos com o intuito de fornecer um sistema de ajuda ao piloto, buscando aumentar a eficiência nas voltas.

6 CONCLUSÕES

Ao longo deste trabalho, foi apresentado o desenvolvimento de um sistema de telemetria para um protótipo de carro elétrico, com foco na aplicação em competições de eficiência energética. O principal objetivo deste projeto foi projetar e implementar um sistema capaz de coletar e fornecer informações relevantes sobre o desempenho do veículo, visando otimizar sua eficiência e maximizar a performance durante as provas.

A fundamentação teórica abordou conceitos importantes sobre veículos elétricos, competições universitárias, telemetria e os componentes essenciais para a implementação do sistema. Com base nesses fundamentos, foram estabelecidos os requisitos de projeto, incluindo tanto os requisitos funcionais quanto os requisitos não-funcionais.

No desenvolvimento do projeto, foram propostas soluções para atender aos requisitos estabelecidos. Foram utilizados microcontroladores, sensores de velocidade, corrente, tensão, temperatura e localização, além de uma estrutura de comunicação e um banco de dados externo para armazenamento das informações coletadas. Também foi implementada uma interface de visualização dos dados, tanto para o piloto quanto para a equipe externa.

Durante os testes realizados, constatou-se que o Raspberry Pi não apresentou desempenho satisfatório na alimentação do sistema, sendo substituído por um ESP-12F. Esse ajuste proporcionou uma melhor capacidade de tratamento dos dados dos sensores e uma comunicação mais eficiente, utilizando o protocolo MQTT. Os resultados dos testes demonstraram a eficácia do sistema na medição de velocidade, potência elétrica, carga da bateria, localização, temperatura, entre outros parâmetros relevantes.

Com base nos resultados obtidos, é possível concluir que o sistema de telemetria desenvolvido atendeu aos requisitos propostos, fornecendo informações em tempo real sobre o desempenho do veículo, conforme pode ser visto no Quadro 4. A utilização de sensores adequados e a integração com um banco de dados externo possibilitaram o armazenamento e análise das informações coletadas, contribuindo para o aprimoramento contínuo do veículo e auxiliando na tomada de decisões estratégicas durante as competições.

Além disso, a modularidade dos componentes utilizados no sistema facilita a manutenção e a manipulação por parte da equipe, garantindo sua viabilidade prática. Os resultados alcançados demonstram o potencial do sistema de telemetria desenvolvido como uma ferramenta eficiente para competições de eficiência energética, contribuindo para o avanço da tecnologia e o desenvolvimento de veículos elétricos mais eficientes.

No entanto, vale ressaltar que este trabalho possui algumas limitações, como a

Quadro 4 – Requisitos de Sistema e erros associados

Referência	Atribuição	Cumprimento	Erro
RF1	Velocidade	✓	±0,4 Km/h
RF2	Potência Elétrica	✓	Variável - ACS712
RF3	Carga da Bateria	✓	0,00
RF4	Localização	✓	±3,92 [m]
RF5	Temperatura	✓	±0,5 °C
RF6	Banco de dados	✓	
RF7	Interface de visualização	✓	
RF8	Visualização da Equipe	✓	

Fonte: Autor (2023).

faixa de operação dos sensores e a necessidade de um ajuste contínuo para atender a requisitos específicos de cada competição e verificação anual para atendimento das normas atualizadas. Sendo assim, sugere-se a continuidade deste trabalho, a fim de aprimorar o sistema, reduzindo tempo de resposta, aumentando a precisão do sistema e atendendo novas demandas, gerando melhorias ao sistema.

O projeto do sistema de telemetria tem potencial para ser utilizado em futuras pesquisas, gerando novos conhecimentos e ampliações do sistema. Sua flexibilidade permite a incorporação de novos sensores e a implementação de algoritmos integrados, possibilitando explorar diferentes aplicações para obter informações mais precisas, realizar um sistema de auxílio ao piloto, entre outras aplicações possíveis. Dessa forma, além de atender às necessidades atuais, o sistema impulsiona a evolução contínua, contribuindo para o avanço científico e o desenvolvimento de soluções eficientes e inovadoras.

REFERÊNCIAS

ABUBAKAR, I. et al. Calibration of zmpt101b voltage sensor module using polynomial regression for accurate load monitoring. **ARPN Journal of Engineering and Applied Sciences**, ARPN, n. 12, p. 1076, fev. 2017.

Adafruit Industries. **Adafruit_ADS1X15**. 2023. Repositório GitHub. Disponível em: https://github.com/adafruit/Adafruit_ADS1X15.

ALLEGRO. **ACS712: Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor**. Manchester, 2022.

ALMEIDA, F. **Triangulação de localização do GPS a partir dos satélites**. 2017. HI Tecnologia. Disponível em: <https://materiais.hitecnologia.com.br/blog/o-que-%C3%A9-encoder-para-que-serve-como-escolher-como-interfacear/>.

AMARAL, P. **Quem inventou o carro elétrico?** 2022. CanalTech. Disponível em: <https://canaltech.com.br/carros/quem-inventou-o-carro-eletrico-209668/>.

BAHOS, A. et al. Sistema de posicionamiento vehicular respaldado por sensores inerciales. In: UNIVERSIDAD INCA GARCILASO DE LA VEGA. **Congreso Internacional de Computación y Telecomunicaciones**. Lima, 2016. p. 198–199.

BARAN, R.; LEGEY, L. F. L. **Veículos Elétricos: História e perspectivas no brasil**. **BNDES Setorial**, BNDES, n. 33, p. 212–214, nov. 2010.

BEZERRA, A. Conceitos básicos de telemetria e sua importância para o setor de transporte. **Amazon Live Journal**, v. 4, n. 1, p. 1–5, jan/fev 2022.

BOTTURA, C. P.; BARRETO, G. **Veículos Elétricos**. Campinas, SP: Editora da UNICAMP, 1989.

CANTÚ, D. **SISTEMA WEB PARA MONITORAMENTO DE SENSORES DE TEMPERATURA E UMIDADE**. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) — Universidade Tecnológica Federal do Paraná, Pato Branco, 2013.

Cap Sistema. **Tutorial do protocolo MQTT: descrição técnica com exemplo prático de Mosquitto**. 2020. Cap Sistema. Disponível em: <https://capsistema.com.br/index.php/2020/11/07/tutorial-do-protocolo-mqtt-descricao-tecnica-com-exemplo-pratico-de-mosquitto/>.

CHEN, Y.; KUNZ, T. Performance evaluation of iot protocols under a constrained wireless access network. **International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)**, Carleton University, 2016.

CHRISPIM, M.; TORRES, J.; SIMÕES, A. Avaliação comparativa entre veículos elétricos e veículos convencionais no contexto de mitigação das mudanças climáticas. **Congreso Internacional de Computación y Telecomunicaciones**, v. 8, n. 1, p. 127–148, jan./mar. 2016.

CINDRA, J. L.; TEIXEIRA, O. P. B. Calor e temperatura e suas explicações por intermédio de um enfoque histórico. In: MARTINS, R. A. et al. (Ed.). **Filosofia e história da ciência no cone sul: 3º Encontro**. Campinas: AFHIC, 2004. p. 240–248.

DELGADO, F. et al. Carros elétricos. **FGV Energia**, v. 4, n. 7, maio 2017.

DURAND, S. A. D. G. . F. R. **ESTUDO DE COMUNICAÇÃO VIA SATÉLITE**. Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) — Universidade Norte do Paraná, Londrina, out. 2008.

EFICEM. **O Eficem: história**. Joinville, 2020. Disponível em: <https://eficem.ufsc.br/site-pt-br/equipe.html>. Acesso em: 10 Dez. 2022.

FARINES, J.-M.; FRAGA, J. d. S.; OLIVEIRA, R. d. **Sistemas de tempo real**. [S.l.: s.n.], 2000. v. 2000. 201 p.

FIGUEIROA, J. P. **COMPARAÇÃO EXPERIMENTAL ENTRE MOTORES BLDC PARA APLICAÇÃO EM TRAÇÃO ELÉTRICA DE VEÍCULOS DE COMPETIÇÃO EM EFICIÊNCIA ENERGÉTICA**. Trabalho de Conclusão de Curso (Graduação em Engenharia Automotiva) — Universidade Federal de Santa Catarina, Joinville, abr. 2021.

Fundação Raspberry Pi. **Raspberry Pi**. entre 2012 e 2023. Disponível em: <https://www.raspberrypi.org/>.

GARMIN. **Sensor de Velocidade 2 e Sensor de Cadência 2**. [S.l.], 2023. Disponível em: <https://www8.garmin.com/manuals/webhelp/cadencespeedsensors2/PT-BR>. Acesso em: 28 mar. 2023.

GOLDENSTEIN, M.; AZEVEDO, R. L. **Combustíveis alternativos e inovações no setor automotivo: será o fim da "era do petróleo"?** **BNDES Setorial**, BNDES, n. 23, p. 238, mar. 2006.

Google LLC. **Google Maps**. 2023. Aplicativo de Mapas. Disponível em: <https://www.google.com/maps>.

HABERMANN, D. **Localização topológica e identificação de obstáculos por meio de sensor laser 3D (LIDAR) para aplicação em navegação de veículos autônomos terrestres**. Trabalho de Conclusão de Curso (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, out. 2016.

HOYER, K. G. **The History of Alternative Fuels in Transportation: The case of electric and hybrid cars**. **Utilities Policy**, Elsevier, 2008.

JANG, I.-Y. **Lithium-ion rechargeable cell for power tools: Model name : INR18650-25R**. [S.l.], 2014.

KERSCHBAUMER, R. **Microcontroladores**. [S.l.]: MINISTÉRIO DA EDUCAÇÃO, 2018. (Engenharia de Controle e Automação).

LACERDA, S. L. M. **Avaliação da Eficiência da Comunicação via Rádio-frequência utilizando o transceiver NRF-24L01+ para Monitoramento de sistemas elétricos no conceito de smart grid**. Dissertação de mestrado (Engenharia Elétrica) — Universidade Federal da Paraíba, João Pessoa, 2015.

LEAL, T. A. C. B.; CONSONI, F. L. **Emissões Poluentes dos Veículos: impacto dos combustíveis utilizados e potencialidades da mobilidade elétrica. Textos para Discussão: Núcleo de Estudos e Pesquisas da Consultoria Legislativa**, Brasília, n. 293, jan. 2021.

LIU, T. **Digital-output relative humidity & temperature sensor/module DHT22**. [S.l.], 2020.

LIU, T. **Temperature and humidity module DHT11 Product Manual**. [S.l.], 2020.

LTD., R. P. T. **Raspberry Pi 4 Model B**. [S.l.], 2019.

OLIVEIRA MARINS, J. A. de; GUARIENTI, G. S. S. **Introdução a Banco de Dados**. [S.l.]: MINISTÉRIO DA EDUCAÇÃO, 2019. (UNIVERSIDADE ABERTA DO BRASIL).

MARTELLI, R. **Automação Industrial**. Indaial: Centro Universitário Leonardo da Vinci – UNIASSELVI, 2012.

Maxim Integrated Products. **DS18B20: Programmable Resolution 1-Wire Digital Thermometer**. [S.l.], 2019.

MENEGUZZI, L.; TREIS, R. F.; CENDRON, M. M. UtilizaÇÃo de giroscÓpio e acelerÔmetro para identificaÇÃo de movimentaÇÃo em ambientes tridimensionais. In: IFC. **FICE - Feira de Iniciação Científica e Extensão**. Camboriú, 2016.

MENEZES, A. **Sistema para rastreamento veicular via GPS**. Trabalho de Conclusão de Curso (Graduação em Engenheiro de Controle e Automação) — Universidade Federal de Santa Catarina, Blumenau, maio 2021.

MOWERY, D. C.; SAMPAT, B. N. Universities in national innovation systems. In: FAGERBERG, J.; MOWERY, D. C.; NELSON, R. R. (Ed.). **The Oxford Handbook of Innovation**. [S.l.]: Oxford University Press, 2005. p. 209–210.

OKI, N.; MANTOVANI, S. C. A. **TEEE I-Projeto de Robôs Móveis: MICROCONTROLADORES -PIC**. [S.l.], 2013. Disponível em: https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/microcontroladores-_pic-1.pdf.

SILVA CASTAGINI PADILHA, A. da; ROSA, C.; SOUZA, S. S. de. **Display LCD 16x2**. [S.l.]: Diretoria de Tecnologia e Inovação, 2021. (**Robótica**: Módulo I).

PERA, B. **Banco de Dados - Relacionais e Não Relacionais**. 3. ed. Paraíba: UNIVERSIDADE DO VALE DO PARAIBA, 2021. Disponível em: URLdaApostila(opcional).

POLITERM. **Multímetro Digital - Cód. Pol-76**. São Paulo, 2008. Disponível em: https://elt09.unifei.edu.br/roteiroslab/Multimetro_Digital_POL_76_VC_9808.pdf.

POLITERM. **Fonte de Alimentação Modelo POL-16E**. [S.l.], 2019.

QUEIROS, J. **Sistema de Sensorização e Telemetria de um VEC (Veículo Eléctrico de Competição)**. Dissertação (Mestrado em Engenharia Electrotécnica e de Computadores) — Faculdade de Engenharia da Universidade do Porto, 2011.

RIBEIRO, B. H.; FERREIRA, T. T. **Veículos eléctricos: aspectos básicos, perspectivas e oportunidades**. **BNDES Setorial**, BNDES, n. 32, p. 267, set. 2010.

SHELL. **About Shell Eco-marathon: A rich history of innovation and inspiration**. 2023. Disponível em: <https://www.shellecomarathon.com/about.html>. Acesso em: 22 fev. 2023.

SHELL. **Shell Eco-marathon: 2023 Official Rules**. 2023. Chapter I. Disponível em: <https://www.shellecomarathon.com/about/global-rules.html>. Acesso em: 4 abr. 2023.

SHELL BRASIL. **Shell Eco-marathon Brasil**. Rio de Janeiro, 2022. Disponível em: <https://www.shell.com.br/energia-e-inovacao/ecomarathon/shell-eco-marathon-brasil.html>. Acesso em: 10 Out. 2022.

Shenzhen Ai-Thinker Technology. **ESP-12F Datasheet**. [S.l.], 2022.

SOARES, R. F. R. D. **Comparação entre protocolos da camada de aplicação para IoT**. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) — Universidade Federal Rural de Pernambuco, Recife, 2019.

Solomon Systech. **SSD1306: 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller**. [S.l.], 2008.

STERNAL, K. et al. lectric vehicle for the students' shell eco-marathon competition. design of the car and telemetry system. **Telematics in the Transport Environment**, v. 4, n. 12, p. 1–5, jan/fev 2012.

TAKAI, O. K.; ITALIANO, I. C.; FERREIRA, J. E. **Introdução a Banco de Dados**. São Paulo: DCC-IME-USP, 2005.

TELEMETRIA. 2023. Dicionário Michaelis Moderno Português. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/TELEMETRIA/>.

Texas Instruments. **ADS111x Ultra-Small, Low-Power, I2C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparato**. Dallas, 2018.

THRUN, S.; BURGARD, W.; FOX, D. **PROBABILISTIC ROBOTICS**. [S.l.]: MIT Press, 2000.

Virtuous Tecnologia da Informação. **Triangulação de localização do GPS a partir dos satélites**. entre 2008 e 2023. Só Física. Disponível em: <http://www.sofisica.com.br/conteudos/curiosidades/gps.php>.

VONDER. **Tacômetro Digital a Laser**. [S.l.], 2018. Disponível em: <https://www.vonder.com.br/estatico/vonder/documentos/3870000100/Manual%20de%20Instru¸ões.pdf>.

Weather Spark. **Histórico de condições meteorológicas em 2022 em Rio de Janeiro**. [S.l.], 2022. Disponível em: <https://pt.weatherspark.com/h/y/30563/2022/Condi%C3%A7%C3%B5es-meteorol%C3%B3gicas-hist%C3%B3ricas-durante-2022-em-Rio-de-Janeiro-Brasil#Figures-Temperature>.

WENDLING, M. **Sensores**. [S.l.], 2010. v. 2. Disponível em: <https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>.

APÊNDICE A - CÓDIGO PARA TESTE DE REDE NO RASPBERRY (PYTHON)

```
1 import os
2 import sys
3 import socket
4 import datetime
5 import time
6
7 FILE = os.path.join(os.getcwd(), "networkinfo.log")
8
9 global pings_enviados
10 pings_enviados = 0
11
12 def ping():
13     global pings_enviados
14     try:
15         socket.setdefaulttimeout(3)
16
17         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
18
19         host = "sheets.googleapis.com"
20         port = 443
21
22         server_address = (host, port)
23         s.connect(server_address)
24
25     except OSError as error:
26         return False
27
28     else:
29         s.close()
30         return True
31
32 def calculate_time(start, stop):
33     difference = stop - start
34     seconds = float(str(difference.total_seconds()))
35     return str(datetime.timedelta(seconds=seconds)).split(".")[0]
36
37 def first_check():
38     global pings_enviados
39
40     if ping():
41         live = "\nCONNECTION ACQUIRED\n"
42         print(live)
```

```
43     connection_aquired_time = datetime.datetime.now()
44     aquiring_message = "connection aquired at: " + \
45         str(connection_aquired_time).split(".")[0]
46     print(aquiring_message)
47
48     with open(FILE, "a") as file:
49
50         file.write(live)
51         file.write(aquiring_message)
52
53     return True
54
55 else:
56     not_live = "\nCONNECTION NOT AQUIRED\n"
57     print(not_live)
58
59     with open(FILE, "a") as file:
60         file.write(not_live)
61
62     return False
63
64 def main():
65     global pings_enviados
66
67     monitor_start_time = datetime.datetime.now()
68     monitoring_date_time = "monitering started at: " + \
69         str(monitor_start_time).split(".")[0]
70
71     if first_check():
72         print(monitoring_date_time)
73
74     else:
75         while True:
76             if not ping():
77                 time.sleep(1)
78             else:
79                 first_check()
80                 print(monitoring_date_time)
81                 break
82
83     with open(FILE, "a") as file:
84         file.write("\n")
85         file.write(monitoring_date_time + "\n")
86
87     while True:
88         if ping():
89             print("ping enviado s : ", datetime.datetime.now())
```

```
90     print("Total de pings: ", pings_enviados)
91     pings_enviados = pings_enviados + 1
92     time.sleep(5)
93
94     else:
95         down_time = datetime.datetime.now()
96         fail_msg = "disconnected at: " + str(down_time).split(".")
[0]
97         print("\n", fail_msg)
98
99         with open(FILE, "a") as file:
100             file.write("\n" + fail_msg + "\n")
101
102         while not ping():
103             time.sleep(1)
104
105         up_time = datetime.datetime.now()
106         uptime_message = "connected again: " + str(up_time).split(".")
") [0]
107         down_time = calculate_time(down_time, up_time)
108         unavailability_time = "connection was unavailable for: " +
down_time
109
110         print(uptime_message)
111         print(unavailability_time)
112
113         with open(FILE, "a") as file:
114
115             file.write(uptime_message + "\n")
116             file.write(unavailability_time + "\n")
117
118 main()
```

APÊNDICE B - CÓDIGO PARA TESTE DE REDE NO ESP (IDE ARDUINO)

```
1 #include <ESP8266WiFi.h>
2
3 #ifndef STASSID
4 #define STASSID "Nome_da_Rede"
5 #define STAPSK "Senha_da_Rede"
6 #endif
7
8 const char* ssid = STASSID;
9 const char* password = STAPSK;
10
11 const char* host = "Servidor-para-conectar";
12 const uint16_t port = porta_de_conexao;
13
14 unsigned long tempoDesconectado;
15 unsigned long tempoConectado;
16
17 bool connect = true;
18
19 unsigned long pings_enviados = 0;
20
21 void setup() {
22   Serial.begin(115200);
23
24   Serial.println();
25   Serial.println();
26   Serial.print("Connecting to ");
27   Serial.println(ssid);
28
29   WiFi.mode(WIFI_STA);
30   WiFi.begin(ssid, password);
31
32   while (WiFi.status() != WL_CONNECTED){
33     delay(500);
34     Serial.print(".");
35   }
36
37   Serial.println("");
38   Serial.println("WiFi connected");
39   Serial.println("IP address: ");
40   Serial.println(WiFi.localIP());
41 }
42
```

```
43 void loop() {
44     static bool wait = false;
45
46     WiFiClient client;
47
48     if (!client.connect(host, port)) {
49         tempoDesconectado = millis();
50         connect = false;
51
52         Serial.println(">> Nao foi possivel conectar.");
53         delay(1000);
54         return;
55     }
56
57     if (client.connected()) {
58
59         if (connect == false) {
60             tempoConectado = millis();
61             Serial.print("Tempo desconectado [s]: ");
62             Serial.print((tempoConectado - tempoDesconectado) / 1000.0);
63             Serial.print(" - Pings enviados: ");
64             Serial.println(pings_enviados);
65             connect = true;
66         }
67
68         client.println("Cliente conectado!");
69     }
70
71     unsigned long timeout = millis();
72     while (client.available() == 0) {
73         if (millis() - timeout > 5000) {
74             Serial.println(">>> Client Timeout !");
75             client.stop();
76             delay(5000);
77             return;
78         }
79     }
80
81     if (client.available()) {
82         //Serial.print("Ping enviado!");
83         pings_enviados++;
84     }
85
86     client.stop();
87
88     if (wait) {
89         delay(5000);
```

```
90 }  
91 wait = true;  
92 }
```

APÊNDICE C - DETERMINAÇÃO DE RPM E VELOCIDADE DO MOTOR DO PROTÓTIPO (IDE ARDUINO)

```
1 const int hallPin = 14;
2 volatile unsigned int hallCount = 0;
3 unsigned long previousMillis = 0;
4 double rpm = 0;
5
6 int voltas = 0;
7 float circunferencia = 1.5708;
8 float distancia = 0;
9 float velocidade = 0;
10 float velocidadeAnterior = 0;
11
12 void IRAM_ATTR hallInterrupt() {
13     hallCount++;
14     voltas++;
15 }
16
17 void setup() {
18     Serial.begin(115200);
19     pinMode(hallPin, INPUT_PULLUP);
20     attachInterrupt(digitalPinToInterrupt(hallPin), hallInterrupt, RISING)
21     ;
22     previousMillis=millis();
23 }
24
25 void loop() {
26     velocidadeAnterior = velocidade;
27     distancia += circunferencia;
28     unsigned long currentMillis = millis();
29     unsigned long elapsedTime = currentMillis - previousMillis;
30
31
32     if (hallCount == 1) {
33         detachInterrupt(digitalPinToInterrupt(hallPin));
34         rpm = (hallCount * 60.0) / (elapsedTime/1000.0);
35         hallCount = 0;
36
37         velocidade = 3.6*rpm*(circunferencia)/60;
38
39         Serial.print("Rpm: ");
40         Serial.print(rpm);
```

```
41     Serial.println(" RPM");
42     Serial.print("Velocidade: ");
43     Serial.print(velocidade);
44     Serial.println( "m/s");
45     Serial.print("voltas: ");
46     Serial.print(voltas);
47     Serial.println(" voltas");
48     Serial.print("Segundos: ");
49     Serial.println(elapsedTime /1000.0);
50
51     attachInterrupt(digitalPinToInterrupt(hallPin), hallInterrupt,
52     FALLING);
53     previousMillis = currentMillis;
54 }
55 }
```

APÊNDICE D - CÓDIGO PARA TESTE DE CORRENTE E TENSÃO (IDE ARDUINO)

```
1 #include <Adafruit_ADS1X15.h>
2
3 Adafruit_ADS1115 ads;
4
5 float current = 0;
6
7 void setup() {
8   Serial.begin(115200);
9
10  ads.setGain(GAIN_ONE);
11
12  if (!ads.begin()) {
13    Serial.println("Falha ao inicializar ADS.");
14    while (1);
15  }
16 }
17
18 void loop() {
19   int16_t adc0;
20   int16_t adc1;
21
22   float volts0;
23   float volts1;
24
25   adc0 = ads.readADC_SingleEnded(0);
26   adc1 = ads.readADC_SingleEnded(1);
27
28   volts0 = ads.computeVolts(adc0);
29   volts1 = ads.computeVolts(adc1);
30
31   current = (volts0 - 1.65) / 0.066;
32
33   Serial.print("Corrente : ");
34   Serial.println(current);
35
36   Serial.println(" ");
37
38   Serial.print("Tensao: ");
39   Serial.println(volts1 / 0.09090909);
40
41   Serial.println(" ----- ");
42
```

```
43   delay(1000);  
44 }
```

APÊNDICE E - CÓDIGO PARA OBTENÇÃO DE LOCALIZAÇÃO, DATA E HORA (IDE ARDUINO)

```
1 #include <Arduino.h>
2 #include <SoftwareSerial.h>
3 #include <TinyGPS++.h>
4 #include <TinyGPSPlus.h>
5
6 TinyGPSPlus gps;
7 SoftwareSerial gpsSerial(13, 15); // RX, TX
8 char buffer[100];
9
10 void printData() {
11     if (gps.location.isUpdated()) {
12         double lat = gps.location.lat();
13         double lng = gps.location.lng();
14
15         double altitude = gps.altitude.meters();
16
17         int year = gps.date.year();
18         int month = gps.date.month();
19         int day = gps.date.day();
20
21         int hour = gps.time.hour();
22         int minute = gps.time.minute();
23         int second = gps.time.second();
24
25         snprintf(buffer, sizeof(buffer),
26                 "Latitude: %.8f, Longitude: %.8f, Altitude: %.2f m, "
27                 "Date/Time: %d-%02d-%02d %02d:%02d:%02d",
28                 lat, lng, altitude, year, month, day, (hour - 3), minute,
29                 second);
30         Serial.println(buffer);
31     }
32 }
33
34 void setup() {
35     Serial.begin(9600);
36     gpsSerial.begin(9600);
37 }
38
39 void loop() {
40     while (gpsSerial.available() > 0) {
```

```
41     if (gps.encode(gpsSerial.read())) {  
42         printData();  
43     }  
44 }  
45 }
```

APÊNDICE F - CÓDIGO PARA TESTE DE TEMPERATURA (IDE ARDUINO)

```
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3
4 #define dados 2
5
6 OneWire oneWire(dados);
7 DallasTemperature sensors(&oneWire);
8
9 void setup(void){
10   Serial.begin(9600);
11   Serial.println("Demonstra o do funcionamento do sensor");
12   sensors.begin();
13 }
14
15 void loop(void){
16   sensors.requestTemperatures();
17   Serial.print("A temperatura: ");
18   Serial.println(sensors.getTempCByIndex(0));
19 }
```

APÊNDICE G - SCRIPT DE PUBLICAÇÃO DA TEMPERATURA EM BROKER MQTT (IDE ARDUINO)

```
1 #include "ESP8266WiFi.h"
2 #include <PubSubClient.h>
3
4 #include <OneWire.h>
5 #include <DallasTemperature.h>
6
7 #define dados 2
8
9 OneWire oneWire(dados);
10 DallasTemperature sensors(&oneWire);
11
12 #define MSG_BUFFER_SIZE (50)
13 char msg[MSG_BUFFER_SIZE];
14
15 //Parametros de conexão
16 const char *ssid = "Nome_da_Rede"; //rede
17 const char *password = "Senha_da_Rede"; //senha
18
19
20 // MQTT Broker
21 const char *mqtt_broker = "test.mosquitto.org"; //Host do broker
22 const char *topic = "Seu_topico/subtopico"; //Topico a ser subscrito e
    publicado
23 const char *mqtt_username = ""; //Usuario
24 const char *mqtt_password = ""; //Senha
25 const int mqtt_port = 1883; //Porta
26
27 bool mqttStatus = 0;
28
29 WiFiClient espClient;
30 PubSubClient client(espClient);
31
32 bool connectMQTT();
33 void callback(char *topic, byte *payload, unsigned int length);
34
35 void setup(void) {
36     Serial.begin(9600);
37     sensors.begin();
38
39     WiFi.begin(ssid, password);
40
```

```
41 Serial.println();
42 Serial.print("Conectando");
43 while (WiFi.status() != WL_CONNECTED) {
44     delay(500);
45     Serial.print(".");
46 }
47 Serial.println("");
48 Serial.println("WiFi connected");
49
50 Serial.println(WiFi.localIP());
51
52 mqttStatus = connectMQTT();
53 }
54
55 void loop() {
56     sensors.requestTemperatures();
57
58     float abc = 0;
59
60     abc = (float)sensors.getTempCByIndex(0);
61     snprintf (msg, 12, "%.f", abc);
62     Serial.println(msg);
63     client.publish(topic, msg);
64
65     if (mqttStatus) {
66         client.loop();
67     }
68 }
69
70 bool connectMQTT() {
71     byte tentativa = 0;
72     client.setServer(mqtt_broker, mqtt_port);
73     client.setCallback(callback);
74
75     do {
76         String client_id = "Eficem-";
77         client_id += String(WiFi.macAddress());
78
79         if (client.connect(client_id.c_str(), mqtt_username, mqtt_password))
80         {
81             Serial.println("Exito na conex o:");
82             Serial.printf("Cliente %s conectado ao broker\n", client_id.c_str
83             ());
84         } else {
85             Serial.print("Falha ao conectar: ");
86             Serial.print(client.state());
87             Serial.println();
88         }
89     } while (tentativa < 10);
90 }
```

```
86     Serial.print("Tentativa: ");
87     Serial.println(tentativa);
88     delay(2000);
89 }
90     tentativa++;
91 } while (!client.connected() && tentativa < 5);
92
93 if (tentativa < 5) {
94     client.subscribe(topic);
95     return 1;
96 } else {
97     Serial.println("N o conectado");
98     return 0;
99 }
100 }
101
102 void callback(char *topic, byte *payload, unsigned int length) {
103     Serial.print("Message arrived in topic: ");
104     Serial.println(topic);
105     Serial.print("Message:");
106     for (int i = 0; i < length; i++) {
107         Serial.print((char)payload[i]);
108     }
109     Serial.println();
110     Serial.println("-----");
111 }
```

APÊNDICE H - SCRIPT DE INSCRIÇÃO EM BROKER MQTT QUE ENVIA INFORMAÇÕES AO BANCO DE DADOS (PYTHON)

```
1 import paho.mqtt.client as mqtt
2 from struct import unpack
3 from time import sleep
4
5 topic = " Eficem/telemetria/#"
6
7 def on_connect (client, data, rc):
8     client.subscribe ([[topic, 0]])
9
10 def on_message(client, userdata, msg):
11     v = unpack ( " >H " , msg . payload ) [0]
12     print msg . topic + " / " + str ( v )
13
14 client = mqtt.Client(client_id = SCADA ,
15                       protocol = mqtt . MQTTv31 )
16
17 client.on_connect = on_connect
18 client.on_message = on_message
19
20 client . connect ( " 127.0.0.1 " , 1883)
21
22 client . loop_forever ( )
23
24 def get_database():
25     from pymongo import MongoClient
26     import pymongo
27
28     CONNECTION_STRING = "mongodb+srv://<usuario>:<senha>@cluster-nome.3
29 nk2a.mongodb.net
30 /myFirstDatabase?retryWrites=true&w=majority"
31
32     client = MongoClient(CONNECTION_STRING)
33
34     return client[ telemetria ]
35
36 dbname = get_database() # conexao com database
37
38 collection_name = dbname["sensores"] # conexao com colecao
39
40 leitura_sensores = {
41     "_id" : id_num,
```

```
41     "potencia" : pot,  
42     "velocidade" : vel,  
43     "temperatura" : temp  
44 }  
45  
46 collection_name.insert_many([leitura_sensores])
```

APÊNDICE I - SCRIPT PARA EXIBIR VALORES NA TELA DO PILOTO

```
1 #include <LiquidCrystal_I2C.h>
2
3 int lcdColumns = 16;
4 int lcdRows = 2;
5
6 float pot = 0;
7 float vel = 0;
8 float temp = 0;
9
10 #define MSG_BUFFER_SIZE (50)
11
12 char msg [ MSG_BUFFER_SIZE ];
13
14
15 LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
16
17 void setup(){
18     lcd.init();
19     lcd.backlight();
20 }
21
22 void loop(){
23     snprintf (msg, 4, "%2.2f", vel);
24     lcd.setCursor(0, 0);
25     lcd.print(msg);
26     lcd.setCursor(6, 0);
27     lcd.print("Km/h");
28
29     snprintf (msg ,5, "%2.3f" , pot);
30     lcd.setCursor(0,1);
31     lcd.print(msg);
32     lcd.setCursor(6,1);
33     lcd.print(" W");
34
35     snprintf (msg ,3, "%2.1f" , temp);
36     lcd.setCursor(10,1);
37     lcd.print(msg);
38     lcd.setCursor(15,1);
39     lcd.print("C");
40
41
42     delay(2000);
```

```
43  lcd.clear();  
44  }
```

APÊNDICE J - LOG DE TESTES DE REDE REALIZADOS PELO ESP

```
1 12:55:31.735 -> >> Nao foi possivel conectar.
2 12:55:32.728 -> >> Nao foi possivel conectar.
3 12:55:33.809 -> Tempo desconectado [s]: 1.07 - Pings enviados: 131
4 13:18:57.609 -> >>> Client Timeout !
5 13:20:53.828 -> >> Nao foi possivel conectar.
6 13:20:57.460 -> Tempo desconectado [s]: 3.65 - Pings enviados: 420
7 13:40:22.020 -> >> Nao foi possivel conectar.
8 13:40:25.668 -> Tempo desconectado [s]: 3.68 - Pings enviados: 643
9 13:40:46.381 -> >> Nao foi possivel conectar.
10 13:40:52.394 -> >> Nao foi possivel conectar.
11 13:40:53.470 -> Tempo desconectado [s]: 1.06 - Pings enviados: 647
12 13:42:39.814 -> >>> Client Timeout !
13 13:43:15.690 -> >> Nao foi possivel conectar.
14 13:43:16.886 -> Tempo desconectado [s]: 1.19 - Pings enviados: 671
15 13:44:34.532 -> >> Nao foi possivel conectar.
16 13:44:38.291 -> Tempo desconectado [s]: 3.76 - Pings enviados: 685
17 13:45:09.725 -> >> Nao foi possivel conectar.
18 13:45:11.160 -> >> Nao foi possivel conectar.
19 13:45:12.280 -> Tempo desconectado [s]: 1.12 - Pings enviados: 690
20 13:45:38.220 -> >> Nao foi possivel conectar.
21 13:45:42.077 -> Tempo desconectado [s]: 3.86 - Pings enviados: 694
22 13:46:08.122 -> >> Nao foi possivel conectar.
23 13:46:09.167 -> Tempo desconectado [s]: 1.06 - Pings enviados: 698
24 13:48:19.611 -> >> Nao foi possivel conectar.
25 13:48:20.846 -> Tempo desconectado [s]: 1.23 - Pings enviados: 722
26 13:49:50.930 -> >> Nao foi possivel conectar.
27 13:49:54.808 -> Tempo desconectado [s]: 3.87 - Pings enviados: 738
28 13:50:51.083 -> >> Nao foi possivel conectar.
29 13:50:54.599 -> Tempo desconectado [s]: 3.53 - Pings enviados: 748
30 13:51:20.902 -> >> Nao foi possivel conectar.
31 13:51:22.087 -> Tempo desconectado [s]: 1.17 - Pings enviados: 752
32 13:52:27.039 -> >>> Client Timeout !
33 13:53:19.065 -> >>> Client Timeout !
34 13:53:44.817 -> >> Nao foi possivel conectar.
35 13:53:48.402 -> Tempo desconectado [s]: 3.56 - Pings enviados: 774
36 13:55:50.973 -> >> Nao foi possivel conectar.
37 13:55:54.824 -> Tempo desconectado [s]: 3.87 - Pings enviados: 796
38 13:57:31.370 -> >> Nao foi possivel conectar.
39 13:57:34.991 -> Tempo desconectado [s]: 3.63 - Pings enviados: 813
40 13:58:48.330 -> >> Nao foi possivel conectar.
41 13:58:52.361 -> Tempo desconectado [s]: 4.05 - Pings enviados: 826
42
```

```
43 -----
44
45 14:00:57.817 -> .....
46 14:01:01.534 -> WiFi connected
47 14:01:01.534 -> IP address:
48 14:01:01.534 -> 192.168.225.157
49 15:26:50.775 -> >> Nao foi possivel conectar.
50 15:26:51.751 -> >> Nao foi possivel conectar.
51 15:26:52.737 -> >> Nao foi possivel conectar.
52 15:26:53.778 -> >> Nao foi possivel conectar.
53 15:26:54.772 -> >> Nao foi possivel conectar.
54 15:26:55.791 -> >> Nao foi possivel conectar.
55 15:26:56.837 -> Tempo desconectado [s]: 1.07 - Pings enviados: 998
56 15:48:11.541 -> >>> Client Timeout !
57 15:58:32.808 -> >> Nao foi possivel conectar.
58 15:58:33.884 -> Tempo desconectado [s]: 1.08 - Pings enviados: 1362
59 15:58:55.154 -> >> Nao foi possivel conectar.
60 15:58:56.232 -> Tempo desconectado [s]: 1.09 - Pings enviados: 1366
61 16:04:40.304 -> >> Nao foi possivel conectar.
62 16:04:46.332 -> >> Nao foi possivel conectar.
63 16:04:56.498 -> >> Nao foi possivel conectar.
64 16:04:57.624 -> Tempo desconectado [s]: 1.11 - Pings enviados: 1430
65 16:11:08.093 -> >> Nao foi possivel conectar.
66 16:11:11.615 -> Tempo desconectado [s]: 3.53 - Pings enviados: 1501
```

APÊNDICE K - LOG DE TESTES DE REDE REALIZADOS PELO RASPBERRY

```
1
2 CONNECTION AQUIRED
3 connection aquired at: 2023-06-18 12:44:23
4 monitering started at: 2023-06-18 12:44:23
5
6 disconnected at: 2023-06-18 12:55:26
7 connected again: 2023-06-18 12:55:44
8 connection was unavailable for: 0:00:18
9
10 disconnected at: 2023-06-18 13:18:55
11 connected again: 2023-06-18 13:18:55
12 connection was unavailable for: 0:00:00
13
14 disconnected at: 2023-06-18 13:20:53
15 connected again: 2023-06-18 13:20:57
16 connection was unavailable for: 0:00:04
17
18 disconnected at: 2023-06-18 13:38:36
19 connected again: 2023-06-18 13:38:36
20 connection was unavailable for: 0:00:00
21
22 disconnected at: 2023-06-18 13:39:51
23 connected again: 2023-06-18 13:39:51
24 connection was unavailable for: 0:00:00
25
26 disconnected at: 2023-06-18 13:40:25
27 connected again: 2023-06-18 13:40:25
28 connection was unavailable for: 0:00:00
29
30 disconnected at: 2023-06-18 13:40:54
31 connected again: 2023-06-18 13:40:54
32 connection was unavailable for: 0:00:00
33
34 disconnected at: 2023-06-18 13:42:35
35 connected again: 2023-06-18 13:42:39
36 connection was unavailable for: 0:00:04
37
38 disconnected at: 2023-06-18 13:43:13
39 connected again: 2023-06-18 13:43:17
40 connection was unavailable for: 0:00:04
41
42 disconnected at: 2023-06-18 13:44:32
```

```
43 connected again: 2023-06-18 13:44:37
44 connection was unavailable for: 0:00:04
45
46 disconnected at: 2023-06-18 13:45:05
47 connected again: 2023-06-18 13:45:09
48 connection was unavailable for: 0:00:04
49
50 disconnected at: 2023-06-18 13:45:38
51 connected again: 2023-06-18 13:45:43
52 connection was unavailable for: 0:00:04
53
54 disconnected at: 2023-06-18 13:46:07
55 connected again: 2023-06-18 13:46:11
56 connection was unavailable for: 0:00:04
57
58 disconnected at: 2023-06-18 13:48:17
59 connected again: 2023-06-18 13:48:18
60 connection was unavailable for: 0:00:01
61
62 disconnected at: 2023-06-18 13:49:54
63 connected again: 2023-06-18 13:49:54
64 connection was unavailable for: 0:00:00
65
66 disconnected at: 2023-06-18 13:50:49
67 connected again: 2023-06-18 13:50:53
68 connection was unavailable for: 0:00:04
69
70 disconnected at: 2023-06-18 13:51:16
71 connected again: 2023-06-18 13:51:17
72 connection was unavailable for: 0:00:00
73
74 disconnected at: 2023-06-18 13:52:22
75 connected again: 2023-06-18 13:52:22
76 connection was unavailable for: 0:00:00
77
78 disconnected at: 2023-06-18 13:55:48
79 connected again: 2023-06-18 13:55:54
80 connection was unavailable for: 0:00:05
81
82 disconnected at: 2023-06-18 13:57:29
83 connected again: 2023-06-18 13:57:33
84 connection was unavailable for: 0:00:04
85
86
87 -----
88
89
```

```
90 CONNECTION AQUIRED
91 connection aquired at: 2023-06-18 14:01:00
92 monitering started at: 2023-06-18 14:00:59
93
94 disconnected at: 2023-06-18 15:26:51
95 connected again: 2023-06-18 15:27:07
96 connection was unavailable for: 0:00:15
97
98 disconnected at: 2023-06-18 16:11:08
99 connected again: 2023-06-18 16:11:09
100 connection was unavailable for: 0:00:01
101
102 disconnected at: 2023-06-18 17:02:24
103 connected again: 2023-06-18 17:02:39
104 connection was unavailable for: 0:00:15
```

APÊNDICE L - DADOS OBTIDOS ATRAVÉS DO SENSOR GPS

-26.23639267	-48.88594767	-26.23622533	-48.88597817	-26.23512750	-48.88509383	-26.23479717	-48.88494933	-26.23519200	-48.88555467
-26.23639267	-48.88594767	-26.23622533	-48.88597817	-26.23512617	-48.88509483	-26.23479717	-48.88494933	-26.23519200	-48.88555467
-26.23639150	-48.88594783	-26.23622500	-48.88597750	-26.23512617	-48.88509483	-26.23479167	-48.88495050	-26.23519467	-48.88556333
-26.23639150	-48.88594783	-26.23622500	-48.88597750	-26.23512400	-48.88509667	-26.23479167	-48.88495050	-26.23519467	-48.88556333
-26.23639033	-48.88594767	-26.23622517	-48.88597700	-26.23512400	-48.88509667	-26.23477800	-48.88495367	-26.23519667	-48.88557300
-26.23639033	-48.88594767	-26.23622517	-48.88597700	-26.23512183	-48.88509817	-26.23477800	-48.88495367	-26.23519667	-48.88557300
-26.23639000	-48.88594767	-26.23622517	-48.88597667	-26.23512183	-48.88509817	-26.23477133	-48.88495383	-26.23519833	-48.88558267
-26.23639000	-48.88594767	-26.23622517	-48.88597667	-26.23511650	-48.88510217	-26.23477133	-48.88495383	-26.23519833	-48.88558267
-26.23638617	-48.88594850	-26.23622517	-48.88597650	-26.23511650	-48.88510217	-26.23476333	-48.88495400	-26.23520100	-48.88559167
-26.23638617	-48.88594850	-26.23622517	-48.88597650	-26.23511200	-48.88510533	-26.23476333	-48.88495400	-26.23520100	-48.88559167
-26.23638817	-48.88594850	-26.23622533	-48.88597617	-26.23511200	-48.88510533	-26.23475250	-48.88495667	-26.23520350	-48.88560067
-26.23638817	-48.88594850	-26.23622533	-48.88597617	-26.23510883	-48.88510783	-26.23475250	-48.88495667	-26.23520350	-48.88560067
-26.23638717	-48.88594900	-26.23622500	-48.88597633	-26.23510883	-48.88510783	-26.23473933	-48.88496033	-26.23520600	-48.88560950
-26.23638717	-48.88594900	-26.23622500	-48.88597633	-26.23510717	-48.88510917	-26.23473933	-48.88496033	-26.23520600	-48.88560950
-26.23638683	-48.88594900	-26.23622483	-48.88597633	-26.23510717	-48.88510917	-26.23472550	-48.88496267	-26.23520733	-48.88561683
-26.23638683	-48.88594900	-26.23622483	-48.88597633	-26.23510583	-48.88511083	-26.23472550	-48.88496267	-26.23520733	-48.88561683
-26.23638700	-48.88594900	-26.23622500	-48.88597667	-26.23510583	-48.88511083	-26.23471633	-48.88496450	-26.23520883	-48.88562433
-26.23638700	-48.88594900	-26.23622500	-48.88597667	-26.23510450	-48.88511233	-26.23471633	-48.88496450	-26.23520883	-48.88562433
-26.23638583	-48.88594917	-26.23622483	-48.88597667	-26.23510450	-48.88511233	-26.23471417	-48.88496367	-26.23521150	-48.88563233
-26.23638583	-48.88594917	-26.23622483	-48.88597667	-26.23510400	-48.88511267	-26.23471417	-48.88496367	-26.23521150	-48.88563233
-26.23639067	-48.88594867	-26.23622467	-48.88597650	-26.23510400	-48.88511267	-26.23472017	-48.88496217	-26.23521400	-48.88563817
-26.23639067	-48.88594867	-26.23622467	-48.88597650	-26.23510317	-48.88511267	-26.23472017	-48.88496217	-26.23521400	-48.88563817
-26.23638717	-48.88594767	-26.23622533	-48.88597750	-26.23510400	-48.88511267	-26.23473933	-48.88495667	-26.23521550	-48.88564567
-26.23638717	-48.88594767	-26.23622533	-48.88597750	-26.23510400	-48.88510783	-26.23473933	-48.88495667	-26.23521550	-48.88564567
-26.23638817	-48.88594767	-26.23622533	-48.88597650	-26.23511200	-48.88510783	-26.23472017	-48.88495667	-26.23521883	-48.88565400
-26.23638817	-48.88594767	-26.23622533	-48.88597650	-26.23511200	-48.88510217	-26.23472017	-48.88496450	-26.23521883	-48.88565400
-26.23639267	-48.88594850	-26.23622467	-48.88597633	-26.23511200	-48.88510217	-26.23477800	-48.88496450	-26.23522200	-48.88566267
-26.23639267	-48.88594850	-26.23622467	-48.88597633	-26.23512400	-48.88509817	-26.23477800	-48.88496450	-26.23522200	-48.88566267
-26.23639267	-48.88594767	-26.23622467	-48.88597633	-26.23512400	-48.88509817	-26.23477800	-48.88496450	-26.23521150	-48.88566267
-26.23638817	-48.88594767	-26.23622533	-48.88597617	-26.23510400	-48.88509817	-26.23477800	-48.88496033	-26.23521150	-48.88566267
-26.23638890	-48.88594831	-26.23622506	-48.88597674	-26.23511294	-48.88510446	-26.23475233	-48.88495827	-26.23520678	-48.88561415
-26.23525800	-48.88463617	-26.23568233	-48.88484750	-26.23470600	-48.88422867	-26.23467083	-48.88482400	-26.23500867	-48.88497867
-26.23525800	-48.88463617	-26.23567267	-48.88485083	-26.23470600	-48.88422867	-26.23467100	-48.88482400	-26.23500867	-48.88497867
-26.23525783	-48.88463583	-26.23567267	-48.88485083	-26.23470467	-48.88422700	-26.23467100	-48.88482400	-26.23497267	-48.88500083
-26.23525783	-48.88463583	-26.23566250	-48.88485467	-26.23470467	-48.88422700	-26.23467117	-48.88402383	-26.23497267	-48.88500083
-26.23525733	-48.88463533	-26.23566250	-48.88485467	-26.23470100	-48.88422617	-26.23467117	-48.88482383	-26.23492550	-48.88502950
-26.23525733	-48.88463533	-26.23565200	-48.88485800	-26.23470100	-48.88422617	-26.23467133	-48.88482400	-26.23492550	-48.88502950
-26.23525700	-48.88463500	-26.23565200	-48.88485800	-26.23469983	-48.88422500	-26.23467133	-48.88482400	-26.23488633	-48.88505417
-26.23525700	-48.88463500	-26.23564183	-48.88486267	-26.23469983	-48.88422500	-26.23467133	-48.88482383	-26.23488633	-48.88505417
-26.23525683	-48.88463467	-26.23564183	-48.88486267	-26.23469933	-48.88422400	-26.23467133	-48.88482383	-26.23484900	-48.88507283
-26.23525683	-48.88463467	-26.23563100	-48.88486600	-26.23469933	-48.88422400	-26.23467133	-48.88482350	-26.23484900	-48.88507283
-26.23525683	-48.88463483	-26.23563100	-48.88486600	-26.23469750	-48.88422333	-26.23467133	-48.88482350	-26.23481333	-48.88509617
-26.23525683	-48.88463483	-26.23562083	-48.88486900	-26.23469750	-48.88422333	-26.23467100	-48.88482350	-26.23481333	-48.88509617
-26.23525683	-48.88463483	-26.23562083	-48.88486900	-26.23469717	-48.88422183	-26.23467100	-48.88482350	-26.23478733	-48.88511317
-26.23525683	-48.88463483	-26.23561183	-48.88487200	-26.23469717	-48.88422183	-26.23467033	-48.88482300	-26.23478733	-48.88511317
-26.23525667	-48.88463400	-26.23561183	-48.88487200	-26.23469600	-48.88422100	-26.23467033	-48.88482300	-26.23479250	-48.88511350
-26.23525667	-48.88463400	-26.23560283	-48.88487450	-26.23469600	-48.88422100	-26.23467000	-48.88482317	-26.23479250	-48.88511350
-26.23525683	-48.88463367	-26.23560283	-48.88487450	-26.23469433	-48.88422033	-26.23467000	-48.88482317	-26.23478733	-48.88511867
-26.23525683	-48.88463367	-26.23559400	-48.88487633	-26.23469433	-48.88422033	-26.23466950	-48.88482367	-26.23478733	-48.88511867
-26.23525700	-48.88463383	-26.23559400	-48.88487633	-26.23469333	-48.88421983	-26.23466950	-48.88482367	-26.23477683	-48.88512867
-26.23525700	-48.88463383	-26.23558400	-48.88487933	-26.23469333	-48.88421983	-26.23466933	-48.88482367	-26.23477683	-48.88512867
-26.23525700	-48.88463383	-26.23558400	-48.88487933	-26.23469333	-48.88421900	-26.23466933	-48.88482367	-26.23478217	-48.88512933
-26.23525700	-48.88463383	-26.23557267	-48.88488267	-26.23469333	-48.88421900	-26.23466950	-48.88482317	-26.23478217	-48.88512933
-26.23525683	-48.88463417	-26.23557267	-48.88488267	-26.23469400	-48.88421817	-26.23466950	-48.88482317	-26.23479250	-48.88512933
-26.23525683	-48.88463417	-26.23567267	-48.88488267	-26.23469400	-48.88421817	-26.23466983	-48.88482250	-26.23479250	-48.88512933
-26.23525650	-48.88463433	-26.23567267	-48.88488267	-26.23469267	-48.88421867	-26.23466983	-48.88482250	-26.23492550	-48.88511350
-26.23525650	-48.88463433	-26.23566250	-48.88486900	-26.23469267	-48.88421867	-26.23467000	-48.88482133	-26.23492550	-48.88511350
-26.23525650	-48.88463433	-26.23566250	-48.88486900	-26.23469200	-48.88421967	-26.23467000	-48.88482133	-26.23497267	-48.88509617
-26.23525650	-48.88463433	-26.23563100	-48.88485800	-26.23469200	-48.88421967	-26.23466983	-48.88482133	-26.23497267	-48.88509617
-26.23525683	-48.88463400	-26.23563100	-48.88485800	-26.23469600	-48.88421900	-26.23466983	-48.88482133	-26.23497267	-48.88500083
-26.23525683	-48.88463400	-26.23557267	-48.88487933	-26.23469600	-48.88421900	-26.23467133	-48.88482400	-26.23479250	-48.88500083
-26.23525699	-48.88463459	-26.23562932	-48.88486794	-26.23469714	-48.88422211	-26.23467044	-48.88479657	-26.23486366	-48.88507836

APÊNDICE M - DADOS OBTIDOS ATRAVÉS DO DIVISOR DE TENSÃO

Voltímetro	T1	Erro	T2	Erro
0,00	0,00	0,00	0,00	0,00
0,25	0,25	0,00	0,25	0,00
0,50	0,50	0,00	0,50	0,00
0,75	0,76	0,01	0,75	0,00
1,00	1,00	0,00	1,00	0,00
1,25	1,26	0,01	1,25	0,00
1,50	1,51	0,01	1,50	0,00
1,75	1,76	0,01	1,75	0,00
2,00	2,01	0,01	2,00	0,00
2,25	2,26	0,01	2,25	0,00
2,50	2,51	0,01	2,50	0,00
3,00	3,01	0,01	3,00	0,00
3,25	3,27	0,02	3,25	0,00
3,50	3,51	0,01	3,50	0,00
4,00	4,02	0,02	4,00	0,00
4,50	4,52	0,02	4,50	0,00
5,00	5,02	0,02	5,00	0,00
5,50	5,52	0,02	5,50	0,00
6,00	6,03	0,03	6,00	0,00
6,50	6,53	0,03	6,50	0,00
7,00	7,03	0,03	7,00	0,00
7,50	7,53	0,03	7,50	0,00
8,00	8,05	0,05	7,99	-0,01
8,50	8,54	0,04	8,50	0,00
9,00	9,04	0,04	9,00	0,00
9,50	9,55	0,05	9,50	0,00
10,00	10,05	0,05	9,99	-0,01
10,50	10,55	0,05	10,50	0,00

Voltímetro	T1	Erro	T2	Erro
11,00	11,06	0,06	11,00	0,00
11,50	11,56	0,06	11,50	0,00
12,00	12,07	0,07	12,00	0,00
12,50	12,57	0,07	12,50	0,00
13,00	13,08	0,08	13,00	0,00
13,50	13,59	0,09	13,50	0,00
14,00	14,08	0,08	14,00	0,00
14,50	14,59	0,09	14,50	0,00
15,00	15,10	0,10	15,00	0,00
15,50	15,59	0,09	15,50	0,00
16,00	16,10	0,10	16,00	0,00
16,50	16,61	0,11	16,50	0,00
17,00	17,11	0,11	17,00	0,00
17,50	17,62	0,12	17,50	0,00
18,00	18,12	0,12	18,00	0,00
18,50	18,63	0,13	18,50	0,00
19,00	19,13	0,13	19,00	0,00
19,50	19,64	0,14	19,50	0,00
20,00	20,15	0,15	20,00	0,00

APÊNDICE N - TESTES DE VELOCIDADE REALIZADOS COM RODA LIVRE

Tacômetro	Sensor Hall	Erro	Tacômetro	Sensor Hall	Erro
18	23,09	5,09	160,3	160,32	0,02
41,1	42,22	1,12	160,5	160,64	0,14
84,3	81,16	-3,14	160,7	160,64	-0,06
89,4	89,25	-0,15	161,2	161,4	0,2
90,1	92,49	2,39	161,6	161,73	0,13
93,2	93,2	0	161,7	161,18	-0,52
94,2	94,01	-0,19	168,8	166,67	-2,13
94,2	94,08	-0,12	169	168,66	-0,34
97,2	95,5	-1,7	172,2	169,13	-3,07
98,8	98,4	-0,4	173,5	170,45	-3,05
100	99,17	-0,83	175,8	173,79	-2,01
104,5	106,5	2	180,2	172,79	-7,41
106,9	100,17	-6,73	180,7	177,25	-3,45
111,6	111,42	-0,18	187,2	187,35	0,15
113,2	113,53	0,33	188,6	188,24	-0,36
116,5	116,39	-0,11	189,3	189,13	-0,17
125,6	124,49	-1,11	189,4	189,72	0,32
127,7	124,29	-3,41	189,8	189,72	-0,08
130,8	132,45	1,65	190	190,93	0,93
133,1	133,93	0,83	191,1	190,33	-0,77
136,2	126,85	-9,35	191,5	190,33	-1,17
136,6	137,46	0,86	191,5	191,69	0,19
137,8	131,22	-6,58	191,6	189,42	-2,18
150,7	151,23	0,53	192,1	192,31	0,21
152,1	153,06	0,96	192,1	191,85	-0,25
157,4	155,44	-1,96	192,3	192,62	0,32
159,4	159,79	0,39	192,4	192	-0,4
160,3	160,43	0,13	194,4	193,7	-0,7
160,3	161,4	1,1			

APÊNDICE O - VALORES LIDOS NO TESTE DE CORRENTE

Ampertímetro	ACS712	Erro
0,00	0,04	-0,04
0,10	0,15	-0,05
0,25	0,29	-0,04
0,40	0,45	-0,05
0,60	0,64	-0,04
0,80	0,84	-0,04
1,00	1,03	-0,03
1,20	1,23	-0,03
1,40	1,42	-0,02
1,60	1,63	-0,03
1,80	1,83	-0,03
2,00	2,01	-0,01
2,20	2,22	-0,02
2,40	2,41	-0,01
2,60	2,62	-0,02
2,80	2,81	-0,01
3,00	3,01	-0,01
3,20	3,21	-0,01
3,40	3,41	-0,01
3,60	3,60	0,00
3,80	3,80	0,00
4,00	4,00	0,00
4,20	4,20	0,00
4,40	4,40	0,00
4,60	4,58	0,02
4,80	4,79	0,01
5,00	4,98	0,02