

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIA EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Projeto de Monitoramento e Telemetria de um Sistema BAJA utilizando LoRa e RTOS

Guilherme Marmirolli

Trabalho de Conclusão de Curso

GUILHERME MARMIROLI

Projeto de Monitoramento e Telemetria de um Sistema BAJA utilizando LoRa e RTOS

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para obtenção do
título de Bacharel em Engenharia Elétrica
pela Universidade Federal de São Carlos.

Orientador: Prof. Dr. Carlos Alberto De
Francisco

São Carlos
15 de junho de 2026

Resumo

Veículos BAJA SAE operam em ambientes *off-road* adversos, sujeitos a vibrações, impactos e grandes distâncias em relação às estações de referência, o que torna o monitoramento em tempo real desafiador com tecnologias convencionais de comunicação sem fio, que impõem restrições de alcance e robustez. Nesse contexto, este trabalho apresenta o desenvolvimento e a validação experimental de um sistema embarcado de monitoramento e telemetria para veículos BAJA SAE, estruturado sobre comunicação LoRa e gerenciamento de tarefas por Sistema Operacional de Tempo Real (RTOS). O sistema é composto por uma placa de circuito impresso projetada para aquisição de dados de rotação do motor, velocidade, temperatura da CVT e aceleração triaxial, utilizando os sensores Hall A3144, termopar Tipo K com interface MAX6675 e acelerômetro MPU6050. O processamento é realizado por um microcontrolador ESP32 com sistema operacional FreeRTOS, com tarefas distribuídas entre os dois núcleos disponíveis. A transmissão dos dados é feita pelo módulo LoRa E220-900T22D, operando na faixa de 915 MHz, com armazenamento local em cartão SD como registro de dados. A validação experimental demonstrou erro de temperatura dentro da tolerância de aproximadamente 2 °C especificada pelo fabricante, resposta coerente dos sensores Hall para RPM e velocidade, porém com leve atraso e resposta amortecida, e funcionamento adequado do acelerômetro na detecção da aceleração da gravidade. O teste em campo do LoRa validou os requisitos de 500m impostos pela distância da competição, enviando 18 bytes, com taxa de perda de pacotes inferior a 5% e RSSI de -85 dBm nessa distância, utilizando antena omnidirecional de 6 dBi. Os resultados encontrados validam a arquitetura proposta e demonstram sua viabilidade para aplicações de telemetria em tempo real em competições BAJA SAE.

Palavras-chave: sistemas embarcados; telemetria veicular; LoRa; FreeRTOS; BAJA SAE; ESP32.

Abstract

BAJA SAE vehicles operate in adverse off-road environments, subject to vibrations, impacts, and large distances from reference stations, which makes real-time monitoring challenging with conventional wireless communication technologies that impose significant range and robustness limitations. In this context, this work presents the development and experimental validation of an embedded monitoring and telemetry system for BAJA SAE vehicles, based on LoRa communication and Real-Time Operating System (RTOS) task management. The proposed system consists of a custom printed circuit board designed for the acquisition of engine rotational speed, vehicle speed, continuously variable transmission (CVT) temperature, and triaxial acceleration data, using Hall-effect sensors (A3144), a Type-K thermocouple with MAX6675 interface, and an MPU6050 accelerometer. Data processing is performed by an ESP32 microcontroller running FreeRTOS, with task distribution across its dual-core architecture to ensure efficient real-time operation. Wireless data transmission is carried out through the E220-900T22D LoRa module operating at 915 MHz, while local data logging is implemented through an SD card interface. Experimental validation demonstrated temperature measurement error within the manufacturer-specified tolerance of approximately 2 °C, coherent sensor responses for rotational speed and vehicle speed with minor delay and damping effects, and proper accelerometer operation for gravitational acceleration detection. Field tests of the LoRa communication link confirmed compliance with the 500 m range requirement imposed by BAJA competition conditions. Sending 18 bytes, it achieved packet loss rates below 5% and -85 dBm received signal strength at this distance using omnidirectional 6 dBi antennas. The results validate the proposed architecture and demonstrate its technical feasibility for real-time telemetry applications in BAJA SAE competitions.

Keywords: embedded systems; vehicle telemetry; LoRa; FreeRTOS; BAJA SAE; ESP32.

Lista de ilustrações

Figura 1 – Veículo BAJA SAE da Equipe Baja UFSCar	12
Figura 2 – Exemplo de um sistema superlaço com interrupção	15
Figura 3 – Diagrama de blocos da execução em um sistema Real-Time Operating System (RTOS)	16
Figura 4 – Arquitetura básica de um sistema embarcado baseado em microcontrolador	18
Figura 5 – Representação esquemática de um termopar com junta quente e junta de referência	21
Figura 6 – Princípio básico de Funcionamento de Sensores MEMS	23
Figura 7 – Funcionamento do sensor Hall em relação a distância do ímã	24
Figura 8 – Modulação LoRa com 868,1 MHz e largura de banda de 125 kHz (caso teórico)	28
Figura 9 – Arquitetura Geral	30
Figura 10 – Bloco de Alimentação	32
Figura 11 – Bloco de Processamento	34
Figura 12 – Bloco de Sensoriamento	35
Figura 13 – Diagrama subsistema transmissor	37
Figura 14 – Diagrama do subsistema receptor	37
Figura 15 – Plataforma do Altium Designer	38
Figura 16 – Ambiente Visual Studio Code	39
Figura 17 – Repositório no GitHub	40
Figura 18 – Execução da aplicação no modelo baseado em RTOS	41
Figura 19 – Diagrama de prioridades para o Núcleo 0	42
Figura 20 – Diagrama de prioridades para o Núcleo 1	42
Figura 21 – Esquemático do bloco de alimentação da placa embarcada	46
Figura 22 – Esquemático do bloco de processamento da placa embarcada	48
Figura 23 – Esquemático do bloco de sensoriamento da placa embarcada	50
Figura 24 – Esquemático do bloco de comunicação LoRa da placa embarcada	52
Figura 25 – Planejamento de layout da placa embarcada	53
Figura 26 – Vista superior 3D da PCI embarcada	54
Figura 27 – Vista inferior 3D da PCI embarcada	54
Figura 28 – Placa de Circuito Impresso	66
Figura 29 – Placa de Circuito Impresso após montagem dos componentes	67

Figura 30 – Termopar tipo K utilizado com o módulo MAX6675	70
Figura 31 – Termopar tipo K utilizado com o multímetro MS-10	70
Figura 32 – Configuração utilizada para o ensaio	71
Figura 33 – Comparação entre instrumentos em baixa temperatura.	71
Figura 34 – Configuração do ensaio em alta temperatura.	73
Figura 35 – Comparação entre instrumentos em alta temperatura.	73
Figura 36 – Instalação do sensor e ímãs na polia do virabrequim	75
Figura 37 – Comparação entre leituras de RPM do sistema e via CAN	76
Figura 38 – Instalação do sensor Hall A3144 e ímã no semieixo	77
Figura 39 – Comparação entre referência e dado obtido para velocidade	78
Figura 40 – Resposta dos eixos de aceleração durante o teste de orientação.	79
Figura 41 – Diagrama GG obtido a partir dos dados de aceleração	80
Figura 42 – Vista satélite do local do ensaio de avaliação do enlace LoRa	81
Figura 43 – Comparação entre valores teóricos e experimentais de RSSI	83

Lista de tabelas

Tabela 1 – Comparação entre protocolos seriais	20
Tabela 2 – Comparação entre tecnologias de comunicação sem fio	27
Tabela 3 – Estimativa de potência máxima consumida pelos componentes	33
Tabela 4 – Sensores utilizados no sistema de sensoriamento	36
Tabela 5 – Bibliotecas utilizadas no desenvolvimento do firmware	40
Tabela 6 – Configuração de Teste dos Sensores	43
Tabela 7 – Correntes estimadas e especificações dos reguladores	47
Tabela 8 – Mapeamento de pinos do ESP32 e suas conexões	49
Tabela 9 – Distribuição das tarefas FreeRTOS nos núcleos do ESP32	56
Tabela 10 – Parâmetros de configuração do módulo embarcado	62
Tabela 11 – Parâmetros de configuração do módulo receptor	64
Tabela 12 – Resultados do ensaio de Verificação em Bancada	68
Tabela 13 – Estatísticas de tempo de execução da tarefa gravarSD()	69
Tabela 14 – Resultados do LoRa em função da distância	82

Lista de abreviaturas e siglas

ADC	Conversor Analógico-Digital
CSS	Chirp Spread Spectrum
CVT	Continuously Variable Transmission
DRC	Design Rule Check
I2C	Inter-Integrated Circuit
IoT	Internet of Things
ISR	Interrupt Service Routine
LoRa	Long Range
MEMS	Microelectromechanical Systems
PCI	Placa de Circuito Impresso
RAM	Random Access Memory
RPM	Rotações por Minuto
RSSI	Received Signal Strength Indicator
RTOS	Real-Time Operating System
SF	Spreading Factor
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

Sumário

1	INTRODUÇÃO	9
1.1	Estrutura do Trabalho	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Competição BAJA SAE e Normas Técnicas Atribuídas	11
2.2	Sistemas Embarcados e Microcontroladores	13
2.2.1	Sistemas Embarcados	13
2.2.1.1	Sistema de superlaço com interrupções	14
2.2.1.2	Sistemas Operacionais em Tempo Real (RTOS)	15
2.2.2	Microcontroladores em Sistemas Embarcados	18
2.3	Sensoriamento e Instrumentação Automotiva	19
2.3.1	Classificação e Interfaces de Sensores	19
2.3.2	Princípios Físicos de Sensoriamento	21
2.3.2.1	Termopares	21
2.3.2.2	Acelerômetros	22
2.3.2.3	Efeito Hall	23
2.4	Comunicação Sem Fio e Telemetria	26
2.4.1	Conceitos de Telemetria em Sistemas Embarcados	26
2.4.2	Princípios da Comunicação LoRa	27
3	MATERIAIS E MÉTODOS	30
3.1	Arquitetura Geral do Sistema	30
3.2	Materiais e Arquitetura de Hardware	31
3.2.1	Sistema de Alimentação	31
3.2.2	Unidade de Processamento	34
3.2.3	Sistema de Sensoriamento	35
3.2.4	Comunicação Sem Fio LoRa	36
3.2.5	Bloco do Receptor	37
3.2.6	Ferramentas de Projeto de Hardware	38
3.3	Arquitetura de Software	39
3.3.1	Ambiente de Desenvolvimento	39
3.3.2	Estrutura do Firmware Embarcado	40
3.4	Validação Experimental do Sistema	43

4	DESENVOLVIMENTO	45
4.1	Desenvolvimento da Placa Embarcada	45
4.1.1	Sistema de Alimentação	45
4.1.2	Unidade de Processamento	47
4.1.3	Sistema de Sensoriamento	49
4.1.4	Módulo de Comunicação LoRa	51
4.1.5	Considerações de Layout	52
4.2	Desenvolvimento de Firmware Embarcado	55
4.2.1	Arquitetura de Tarefas	55
4.2.2	Aquisição de Sensores	57
4.2.3	Monitoramento de Saúde do Sistema	59
4.2.4	Armazenamento em Cartão SD	60
4.2.5	Comunicação LoRa	61
4.3	Desenvolvimento do Receptor e Suporte de Logs	64
5	RESULTADOS OBTIDOS	66
5.1	Placa de Circuito Impresso	66
5.2	Validação Experimental do Sistema	67
5.2.1	Verificação Funcional em Bancada	67
5.2.2	Validação dos Sensores	69
5.2.2.1	Medição de Temperatura	69
5.2.2.1.1	Baixa temperatura	71
5.2.2.1.2	Alta temperatura	72
5.2.2.2	Sensores Hall A3144	74
5.2.2.2.1	Rotações por minuto	74
5.2.2.2.2	Velocidade	76
5.2.2.3	Aceleração	78
5.2.3	Avaliação do Enlace LoRa em Campo	80
5.2.4	Síntese dos Resultados	83
6	CONCLUSÃO	84
6.1	Trabalhos futuros	84
6.2	Considerações finais	84
	REFERÊNCIAS	86

1 INTRODUÇÃO

O setor automotivo brasileiro é um dos segmentos industriais de maior relevância econômica do país. Em 2024, o conjunto de autoveículos, autopeças, máquinas agrícolas e outros produtos do setor movimentou cerca de US\$ 35,9 bilhões em importações (AN-FAVEA, 2025), evidenciando a escala e importância da área. Nesse cenário, a demanda por soluções embarcadas de monitoramento veicular em tempo real tem se intensificado, impulsionada tanto pelo avanço do mercado automotivo quanto pela difusão de aplicações de Internet das Coisas (IoT).

Em aplicações *off-road*, essas exigências tornam-se ainda mais críticas, uma vez que os sistemas estão sujeitos a adversidades mecânicas, como vibração e impactos, e a uma longa distância aos postos de referência, que normalmente são utilizados como ancoragem de dados. Nesse contexto, os veículos do tipo BAJA utilizam-se de sistemas embarcados para viabilizar a telemetria, permitindo a coleta e a transmissão de dados em tempo real e o monitoramento das condições do veículo à distância.

Contudo, nessas condições adversas, soluções de comunicação sem fio convencionais impõem restrições de alcance e robustez que comprometem a integridade da transmissão. Somado a isso, projetos acadêmicos frequentemente operam sob restrições orçamentárias, o que torna comum o uso de módulos comerciais com limitações de precisão e latência, ocasionando erros acumulados ao longo do processamento.

Este trabalho adota como hipótese que a combinação de uma tecnologia de comunicação de longo alcance com uma abordagem estruturada de gerenciamento de tarefas possibilita o desenvolvimento de um sistema de telemetria embarcado mais robusto, com maior alcance de comunicação e maior confiabilidade na transmissão dos dados em campo.

Com base nessa premissa, o objetivo deste trabalho é desenvolver e validar experimentalmente um sistema embarcado de monitoramento e telemetria para veículos BAJA, estruturado sobre comunicação *Long Range* (LoRa) e gerenciamento de tarefas por Sistema Operacional em Tempo Real (RTOS). Para isso, consideram-se a definição da arquitetura do sistema, a implementação do hardware e do software, e a avaliação experimental do desempenho, com ênfase em métricas de alcance, precisão dos sensores e latência no processamento de tarefas.

1.1 Estrutura do Trabalho

O Capítulo 2 apresenta os fundamentos teóricos que embasam o sistema desenvolvido, abordando sistemas embarcados, gerenciamento de tarefas com Sistema Operacional em Tempo Real (RTOS), princípios de sensoriamento e comunicação sem fio com tecnologia *Long Range* (LoRa). O Capítulo 3 descreve os materiais e métodos adotados, incluindo a arquitetura de hardware e software, bem como os ensaios experimentais planejados para validação do sistema. O Capítulo 4 detalha o processo de implementação da placa de circuito impresso, do firmware embarcado e do subsistema receptor. Por fim, são apresentados os resultados obtidos, seguidos das conclusões e das perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos relacionados ao desenvolvimento do sistema de telemetria embarcada proposto neste trabalho. O projeto consiste na aquisição, processamento, armazenamento e transmissão de dados operacionais de um veículo BAJA SAE, incluindo parâmetros como rotação do motor, velocidade, temperatura e aceleração triaxial.

Nesse contexto, inicialmente são apresentados os principais regulamentos, da competição que o BAJA está inserido, que estruturam e delimitam as decisões de projeto. Em seguida, são abordados os conceitos técnicos, como sistemas embarcados e microcontroladores, incluindo arquiteturas de software utilizadas nesse tipo de aplicação. Posteriormente, são discutidos os princípios de funcionamento dos sensores empregados no sistema e, por fim, os fundamentos da comunicação sem fio e da telemetria utilizada para transmissão dos dados.

2.1 Competição BAJA SAE e Normas Técnicas Atribuídas

O projeto BAJA SAE promove o desenvolvimento universitário de veículos *off-road*, que competem em provas compostas por etapas estáticas e dinâmicas. A finalidade é gerar um ambiente de aprendizado prático e inovador, com soluções de engenharia adaptadas para os desafios das competições.

Entre as provas realizadas na competição nacional, destacam-se as avaliações de apresentações de projeto, na qual o grupo avaliador analisa a qualidade do projeto desenvolvido; as provas de aceleração e velocidade, utilizadas para analisar o desempenho do veículo; as provas de tração e manobrabilidade, que avaliam o comportamento dinâmico em terrenos irregulares; e a prova de enduro, caracterizada por funcionamento ativo durante um período de 4 horas em um percurso com obstáculos, exigindo resistência e confiabilidade do veículo.

Além das provas dinâmicas, as equipes também participam de apresentações técnicas de projeto, nas quais são avaliadas decisões de engenharia relacionadas aos sistemas desenvolvidos para o veículo. No final, a equipe que obtiver a maior pontuação total, considerando o desempenho nas provas e a avaliação técnica, é declarada vencedora da competição.

A Figura 1 apresenta o veículo BAJA SAE utilizado pela equipe Baja UFSCar.

Figura 1 – Veículo BAJA SAE da Equipe Baja UFSCar



Fonte: Autoria própria

Todo o veículo é projetado para enfrentar as provas da maneira mais eficiente possível: suspensão exposta para facilitar manutenção, pneus selecionados para melhor tração, rodas abertas visando melhor manobrabilidade, entre outros aspectos. Todavia, além do desempenho, a segurança é fator crucial destes projetos. Por este motivo, a organização SAE estabelece um conjunto de requisitos técnicos e de segurança, que impactam o projeto geral (SAE BRASIL, 2025).

Nesse contexto, o regulamento estabelece regras mínimas para que o veículo seja considerado seguro a participar da competição. Para o sistema elétrico, destacam-se requisitos relacionados às chaves gerais, que possuem modelos e posicionamentos especificados e devem desligar a ignição do motor quando pressionadas; a luz de freio, que deve ser acionada quando o piloto exercer pressão suficiente no pedal de freio; e aos chicotes elétricos, que devem possuir diâmetro mínimo 22 AWG, isolamento resistente à chama e proteção por tubo automotivo.

Adicionalmente, no sistema de alimentação, as baterias devem ser seladas e garantir proteção contra estresses mecânicos e químicos. Além disso, nenhuma bateria pode possuir

capacidade superior a 240 Wh individualmente e, caso exista mais de uma bateria, a soma das capacidades deve ser menor ou igual a 360 Wh. Para baterias que forneçam corrente superior a 800 mA, é necessário o uso de um fusível de 1 A. Caso seja necessário capacidade superior, um memorial de dimensionamento deve ser submetido previamente à organização da competição.

Em contraste, os sistemas de aquisição de dados e telemetria, que são foco deste trabalho, não possuem especificações rígidas quanto à arquitetura ou implementação. A competição permite que as equipes desenvolvam soluções próprias para monitoramento e análise de desempenho, desde que não interfiram nos sistemas críticos do veículo. Essa flexibilidade possibilita diferentes níveis de complexidade e abordagem entre as equipes, variando desde sistemas simples de monitoramento até arquiteturas mais elaboradas de aquisição e transmissão de dados, como apresentado nas seções seguintes.

2.2 Sistemas Embarcados e Microcontroladores

Nesta seção, será apresentado o que é um sistema embarcado, seus requisitos e sua capacidade. Além disso, será fundamentada a peça-chave para este sistema, os microcontroladores.

2.2.1 Sistemas Embarcados

Os sistemas embarcados são o conjunto de hardware e software cujos componentes possuem requisitos e funções limitadas e pré-definidas, com pouca ou nenhuma flexibilidade para adaptação a novas tarefas. Com isso, sua construção e arquitetura visa o alto desempenho, o menor custo e a eficiência energética, visto que muitos destes sistemas são alimentados por baterias (BARR, 1999).

Como exemplos de tais sistemas, no setor automotivo, estão o cluster, freio ABS e a multimídia. Observa-se que, apesar de permitirem um certo grau de personalização, existe uma função clara e definida para cada um, não permitindo a substituição de um sistema por outro.

A partir desses conceitos estabelece-se propriedades fundamentais que definem os sistemas embarcados:

1. Operação dedicada: o sistema é planejado para executar uma função específica ou um conjunto limitado de tarefas. Dessa forma, todo o projeto é orientado ao limitado conjunto de tarefas propostas.

2. Restrições de energia: muitas das vezes, tais sistemas estão alocados em locais com restrições de energia. Com isso, projetos de sistemas embarcados visam baixo consumo energético.
3. Tempo real: a principal característica de sistemas embarcados é a capacidade de operar em tempo real. Isso porque, a tomada de decisão, seja pela própria aplicação ou por supervisor externo, deve ocorrer dentro de prazos específicos.
 - a) Tempo real leve (*Soft real time*): são sistemas que possuem prazos para tarefas mais maleáveis e flexíveis. A perda de tais prazos não causam falha ou interrupção do sistema, apenas perda de dados não críticos. Como exemplo, estão dados multimídia e tarefas de jogos.
 - b) Tempo real forte (*Hard real time*): são sistemas que possuem prazos para tarefas extremamente restritos. A perda de tais prazos causam falha e interrupção instantânea ao sistema. Como exemplo, estão os equipamentos médicos e militares.
4. Confiabilidade e estabilidade: sistemas embarcados, por operarem em tempo real, possuem uma tolerância menor ao acúmulo de erros. Logo, devem possuir uma arquitetura robusta e ser desenvolvidos utilizando normas de detecção de erro.

Em aplicações de aquisição de dados e telemetria veicular em veículos Baja SAE, as exigências são menos críticas, visto o caráter de monitoramento. Nesse viés, o sistema é definido com tempo real leve, devendo ser capaz de manter-se em funcionamento e com cobertura suficiente para aquisição de dados relevantes durante toda a duração da prova, estimada em 5 horas.

2.2.1.1 Sistema de superlaço com interrupções

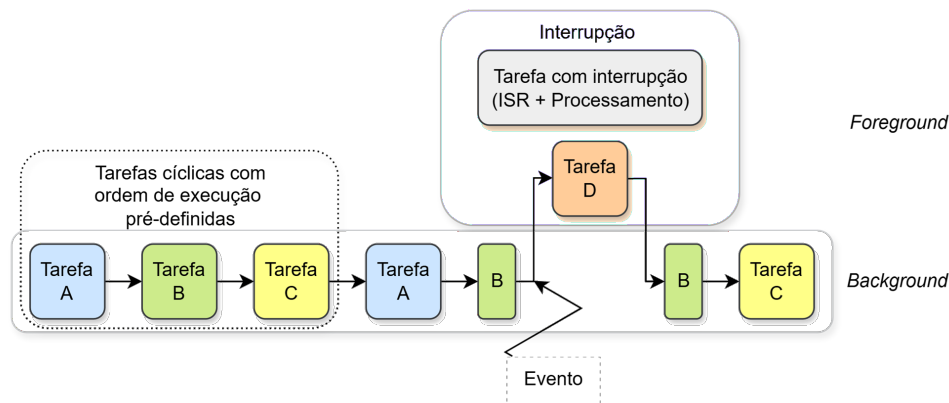
No sistema baseado em superlaço, o programa principal consiste em um laço infinito, ou seja, um bloco de código que se repete indefinidamente enquanto o sistema estiver ligado. A cada repetição, o laço percorre um conjunto fixo de funções, sempre na mesma ordem e sem pular nenhuma etapa. É como uma lista de tarefas que o processador executa do início ao fim, e ao terminar, recomeça do início.

Esse contexto de execução contínua é chamado de ambiente *background*. Nele, as funções são chamadas uma após a outra, e uma nova tarefa só tem início quando a anterior é concluída, ou seja, não há concorrência entre elas. Este tipo de aplicação é amplamente utilizado para sistemas simples, onde não há necessidade de priorizar tarefas críticas e a temporização do laço não gera problemas de desempenho.

Para aplicações mais complexas, é possível integrar o sistema de superlaço com o uso de Interrupções de Serviço (ISR). Nesse mecanismo, ao ocorrer um evento como temporização ou recepção de dados, o processador suspende a tarefa atual, salva seu contexto e direciona o processamento para a tarefa com a interrupção, movendo-a para o *foreground*, onde tem maior prioridade. Após concluir, a execução anterior retorna do ponto interrompido (DENARDIN; BARRIQUELLO, 2019).

Na Figura 2, está caracterizado um sistema de superlaço, mostrando a troca de prioridade, acionadas por interrupções de serviços de rotina.

Figura 2 – Exemplo de um sistema superlaço com interrupção



Fonte: Autoria própria

O laço principal, ou laço *background*, desempenha livremente, com sequência definida pela ordem de inclusão no código - A, B e C, na imagem acima. A troca de prioridade acontece quando a interrupção (ISR) é chamada por um evento, fazendo com que a tarefa que estava em execução seja pausada, para que outra tarefa com maior prioridade aconteça - Tarefa B pausada para que a Tarefa D execute. Após o término da tarefa com interrupção, a tarefa pausada é retomada no mesmo ponto.

A troca de execuções e interrupções temporais fazem com que seja crucial assegurar que a temporização seja ideal para o sistema desenvolvido e que as rotinas críticas sejam priorizadas. Caso contrário, poderá gerar sobreescrita de dados ou rompimento do tempo limite, prejudicando toda a rotina de execução.

2.2.1.2 Sistemas Operacionais em Tempo Real (RTOS)

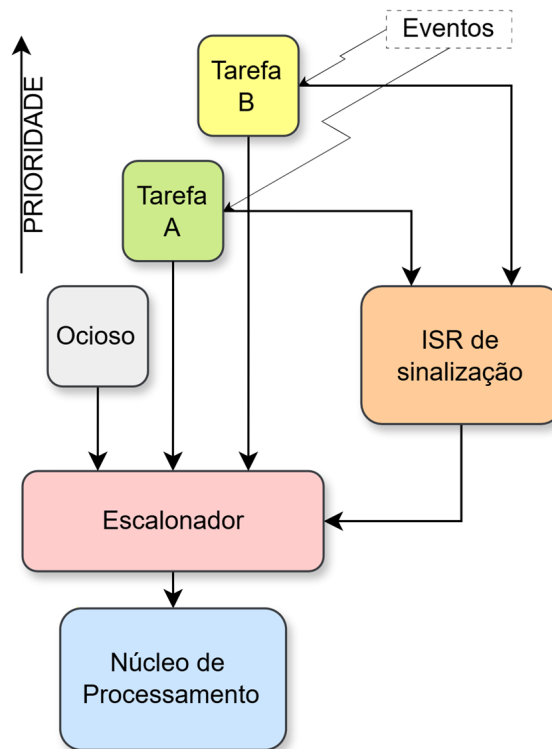
Sistemas Operacionais em Tempo Real (do inglês, *Real-Time Operating System* (RTOS)) surgem como uma solução para organizar e gerenciar a execução de múltiplas tarefas de forma determinística, superando as limitações dos modelos sequenciais baseados em superlaço.

Nesse modelo, a tarefa pronta de maior prioridade pode assumir a execução de forma preemptiva, interrompendo tarefas de menor prioridade ou o estado inativo do processador, o que contribui para menor latência e maior previsibilidade temporal.

Diferentemente da abordagem sequencial baseada em superlaço, na qual parte do processamento costuma ser executada diretamente em rotinas de interrupção, sistemas baseados em Real-Time Operating System (RTOS) utilizam interrupções principalmente para sinalização de eventos, enquanto o processamento principal é realizado no contexto de tarefas gerenciadas (KOPETZ, 2011).

A Figura 3 ilustra o comportamento típico de um sistema operacional em tempo real, com escalonamento preemptivo por prioridade.

Figura 3 – Diagrama de blocos da execução em um sistema RTOS



Fonte: Autoria própria

As tarefas são escalonadas dinamicamente de acordo com sua prioridade e estado, podendo assumir estados como pronta, em execução, suspensa ou bloqueada. Quando não há tarefas prontas para execução, o sistema permanece em estado ocioso.

Quando uma tarefa é desbloqueada, o escalonador avalia as tarefas prontas e seleciona aquela de maior prioridade para execução. Caso uma tarefa de maior prioridade fique pronta enquanto outra tarefa está em execução, o sistema realiza a troca de contexto de

forma preemptiva, interrompendo temporariamente a tarefa atual para executar a tarefa mais prioritária.

Nesse modelo, as interrupções são utilizadas principalmente para sinalização de eventos e sincronização entre tarefas, enquanto o processamento principal é realizado no contexto das tarefas gerenciadas pelo sistema operacional. Dessa forma, as ISR em sistemas RTOS tendem a ser mais curtas e com menor impacto temporal sobre o restante da aplicação, em comparação ao modelo apresentado na Seção 2.2.1.1.

Nesse contexto, podem-se definir os principais mecanismos de um sistema RTOS:

- Escalonamento preemptivo por prioridade: o escalonador define qual tarefa será executada de acordo com sua prioridade, realizando a troca de contexto caso necessário.
- Fatiamento de tempo (*Round-Robin*): quando múltiplas tarefas prontas possuem o mesmo nível de prioridade, o escalonador alterna a execução entre essas tarefas a cada interrupção de relógio, permitindo que ambas executem juntas.
- Troca de contexto: ato de alternar a execução das tarefas gerenciadas pelo escalonador, sem que ocorra perda de contexto ou processamento.
- Mutex: controla o acesso a recursos compartilhados por meio de exclusão mútua, prevenindo corrupção de dados quando múltiplas tarefas tentam acessar simultaneamente o mesmo dado. Isso é feito através de um mecanismo de acesso, como uma chave, que permite que apenas uma tarefa use a variável por vez.
- Grupos de eventos (*Event Groups*): permitem associar diversos eventos dentro do código a um conjunto de bits, de forma que uma tarefa possa aguardar qualquer combinação de sinalizações sem a necessidade de sinalizações independentes.
- Temporização periódica controlada (`vTaskDelayUntil`): bloqueia uma tarefa até um instante absoluto de tempo. Para tanto, ela realiza a compensação do tempo em que a tarefa ficou em execução, fornecendo ao usuário um controle maior sobre a temporização, promovendo tarefas mais previsíveis ao evitar o acúmulo de atrasos gerados nas execuções das tarefas.

Tais características tornam o RTOS adequado para sistemas embarcados de aquisição e telemetria com tarefas concorrentes e temporizações distintas. Isso porque mecanismos como mutex podem ser utilizados para proteger a troca de dados entre tarefas que tentam acessar a mesma variável ao mesmo tempo, viabilizando, por exemplo, a escrita por parte dos sensores e a leitura por parte da comunicação, sem que haja riscos de perda de dados.

Além disso, o mecanismo Grupos de Eventos pode atuar como uma estrutura global de supervisão dos módulos do sistema, possibilitando o registro e a sinalização rápida de falhas. Por fim, a temporização por RTOS *vTaskDelayUntil()* garante maior controle temporal quando comparado ao *delay()* convencional, visto que faz a compensação do tempo de execução da tarefa, ao calcular os intervalos de tempo.

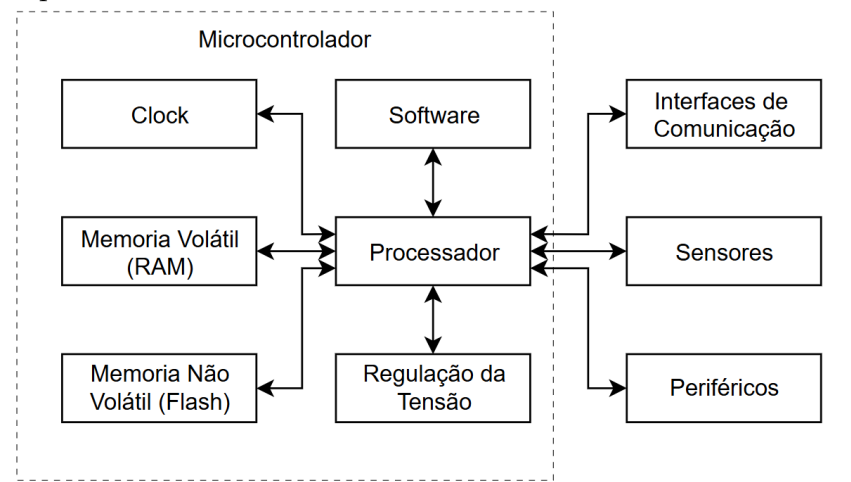
Em suma, o sistema operacional descrito nessa seção oferece mecanismos internos que contribuem de maneira significativa para implementação de sistemas embarcados em tempo real, fornecendo maior controle temporal, estruturas de proteção e maior eficiência na execução das tarefas.

2.2.2 Microcontroladores em Sistemas Embarcados

Microcontroladores são dispositivos eletrônicos que integram, em um único circuito, os principais elementos necessários para a implementação de sistemas computacionais dedicados, incluindo unidade de processamento, memória e periféricos de entrada e saída. Essa integração permite que hardware e software sejam projetados de forma conjunta e otimizada para a execução de tarefas específicas, característica fundamental dos sistemas embarcados (DAWOUD, 2010).

A arquitetura básica de um sistema embarcado baseado em microcontrolador pode ser observada na Figura 4, que ilustra a integração entre os principais blocos funcionais.

Figura 4 – Arquitetura básica de um sistema embarcado baseado em microcontrolador



Fonte: Adaptado de (DAWOUD, 2010)

No núcleo do sistema está o processador, responsável pela execução das instruções do software. Associadas a ele, estão as memórias voláteis (RAM), e não volátil (Memória Flash) Além disso, microcontroladores integram periféricos como temporizadores, conversores ADC e interfaces de comunicação, como: UART, SPI e I2C.

Tais componentes são cruciais na hora da escolha do microcontrolador, que devem atender os requisitos da aplicação e garantir que os recursos disponíveis sejam adequados. Entre os critérios, destaca-se a quantidade de pinos, quantidade de núcleos, frequência de operação, capacidade de memória e as interfaces disponíveis. Também devem ser considerados aspectos como custo e o ecossistema de desenvolvimento, incluindo ferramentas e documentação, que impactam diretamente na implementação e na manutenção do sistema.

A partir disso, passa-se à análise dos elementos responsáveis pela obtenção dos dados no sistema, em particular os sensores e seus princípios de funcionamento.

2.3 Sensoriamento e Instrumentação Automotiva

O sensoriamento automotivo envolve a correta escolha dos dispositivos e a integração destes ao sistema embarcado, considerando aspectos de interface de comunicação, tempo de resposta e estabilidade da medição. A compreensão dos princípios físicos associados a cada tipo de sensor e seu protocolo de comunicação é essencial para garantir a confiabilidade dos dados adquiridos e transmitidos.

2.3.1 Classificação e Interfaces de Sensores

Os sensores utilizados em sistemas embarcados podem ser classificados, de forma geral, em sensores analógicos e digitais. Sensores analógicos fornecem sinais contínuos, normalmente na forma de tensão ou corrente proporcional à grandeza medida, exigindo conversão analógico-digital para processamento pelo microcontrolador. Já sensores digitais realizam internamente essa conversão e disponibilizam os dados por meio de protocolos de comunicação padronizados (WILSON, 2005).

Pode-se definir os protocolos de comunicação serial a partir da necessidade de sincronismo, tipo de transmissão de dados e modo de operação.

1. Sincronismo:

- a) Assíncronos: não necessitam de sinal *clock*. Transferem um único byte por período, marcado por um bit para início e um bit para o fim da mensagem útil.
- b) Síncronos: necessitam de sinal *clock*. Transferem vários bytes seguidos, pois o transmissor e receptor sincronizados bit a bit, a cada pulso do *clock*.

2. Duplex:

- a) *Simplex*: a comunicação ocorre em apenas em um sentido.

- b) *Half Duplex*: a comunicação ocorre nos dois sentidos, mas não ao mesmo tempo. Os dispositivos alternam entre transmitir e receber.
- c) *Full Duplex*: a comunicação ocorre nos dois sentidos simultaneamente. Ambos os dispositivos podem transmitir e receber ao mesmo tempo.

3. Tipo:

- a) Ponto a ponto: permitem apenas dois dispositivos na linha. Ambos podem transmitir e receber dados.
- b) Mestre e escravo: permitem vários dispositivos na linha. Apenas o dispositivo mestre pode iniciar a comunicação. Os dispositivos escravos respondem às solicitações do mestre, enviando ou recebendo dados conforme necessário.
- c) Multi-mestre: permite vários dispositivos na linha. Qualquer dispositivo pode assumir o controle do barramento e iniciar uma transmissão de dados.

Com isso, pode-se organizar os tipos de comunicações mais comuns na indústria automotiva de acordo com a Tabela 1.

Tabela 1 – Comparação entre protocolos seriais

Característica	UART	I2C	SPI
Sincronismo	Assíncrono	Síncrono	Síncrono
Tipo	Ponto a ponto	Multi-mestre	Multi-mestre
Duplex	Full-duplex	Half-duplex	Full-duplex
Taxa típica	9,6-115,2 kbps	400 kbps	25 Mbps
Complexidade	Baixa	Média	Alta
Número de fios	2	2	4

Fonte: Autoria própria

A escolha do protocolo de comunicação está diretamente relacionada com a arquitetura do sistema, pois define a velocidade de transmissão e tamanho dos dados a serem compartilhados. Em contraste, sistemas com maiores capacidades possuem maior complexidade de implementação, tanto no nível de registradores do software, ao nível físico, como interferência a ruídos.

De forma geral, cada componente possui um protocolo de comunicação que melhor se encaixa ao seu perfil. A seguir, será discutido os princípios físicos dos sensores utilizados no setor automotivo e apresentado seu protocolo de comunicação.

2.3.2 Princípios Físicos de Sensoriamento

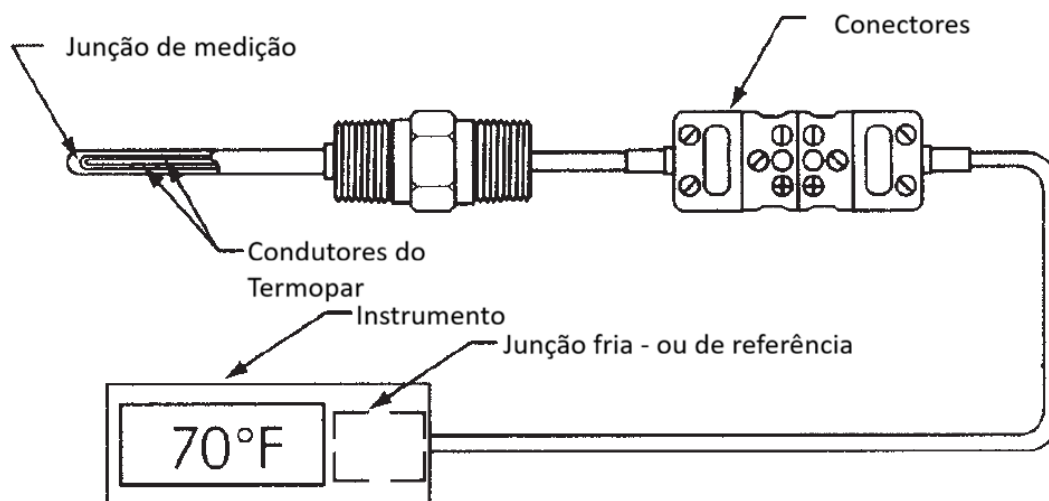
A medição de grandezas físicas em aplicações veiculares baseia-se em diferentes princípios físicos, de acordo com o fenômeno a ser observado.

2.3.2.1 Termopares

O termopar é um dos sensores de temperatura mais amplamente utilizados em aplicações industriais e embarcadas, principalmente devido à sua robustez, ampla faixa de operação e simplicidade construtiva. Seu princípio de funcionamento baseia-se no efeito Seebeck, no qual uma diferença de potencial elétrico é gerada quando dois materiais condutores distintos são submetidos a um gradiente de temperatura.

Fisicamente, o termopar é formado pela junta de dois metais diferentes, criando duas regiões principais: a junta de medição (ou junta quente), exposta à temperatura de interesse, e a junta de referência (ou junta fria), mantida a uma temperatura conhecida ou compensada eletronicamente, como apresentado na Figura 5. A diferença de temperatura entre essas junções resulta na geração de uma força eletromotriz, tipicamente da ordem de microvolts, que pode ser correlacionada com a temperatura (WILSON, 2005).

Figura 5 – Representação esquemática de um termopar com junta quente e junta de referência



Fonte: Adaptado de (DESMARAIS; BREUER, 2001)

Observa-se a presença dos dois condutores metálicos distintos formando a junta de medição, enquanto as extremidades opostas são conectadas ao sistema de aquisição. A diferença de temperatura entre a junta quente e a junta de referência é responsável pela geração da tensão medida. Na prática, sistemas modernos incorporam compensação de

junta fria, eliminando a necessidade de manter fisicamente essa junta a uma temperatura conhecida.

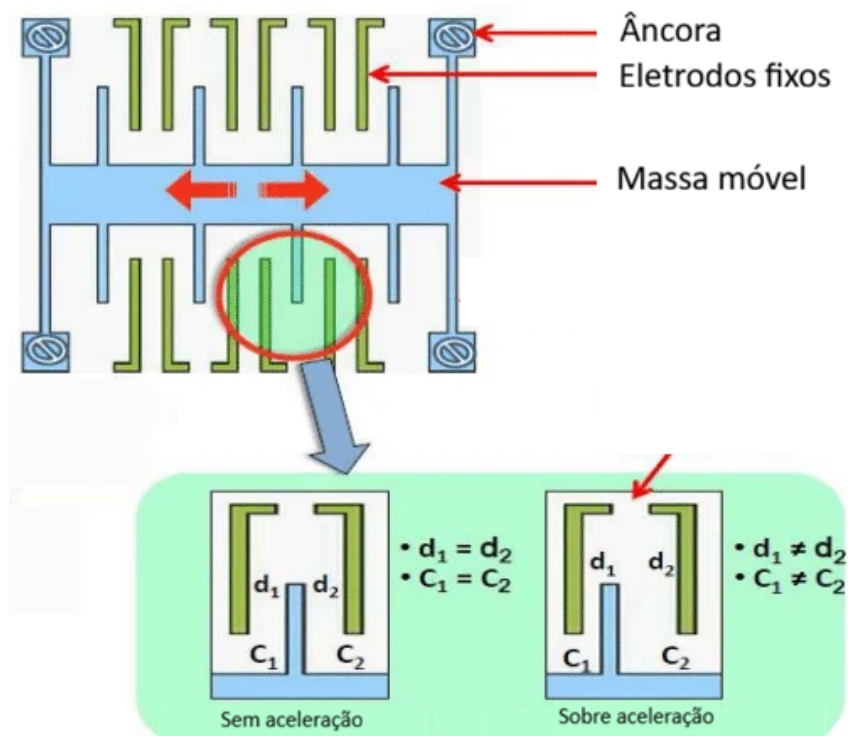
Do ponto de vista de instrumentação, a baixa amplitude do sinal gerado exige o uso de circuitos de condicionamento, como amplificadores de ganho elevado e baixo ruído, além de conversores ADC com resolução adequada. Nesse quesito, surge módulos comerciais prontos para projetos-protótipo, como o MAX6675, que realiza a compensação de junta fria e converte o sinal do termopar tipo K para digital, oferecendo uma saída SPI (MAXIN INTEGRATED, 2021), ideal para integração com microcontroladores e projetos de prototipagem.

2.3.2.2 Acelerômetros

Acelerômetros são sensores capazes de mensurar aceleração linear ao longo de um ou mais eixos, sendo amplamente utilizados em sistemas embarcados para detecção de movimento, orientação e vibração. Em aplicações veiculares, esses sensores são fundamentais para análise dinâmica, aquisição de dados inerciais e estratégias de controle.

Os acelerômetros modernos são, em sua maioria, baseados em tecnologia Sistemas Micro-eletromecânicos (MEMS). Nesse tipo de dispositivo, estruturas mecânicas microscópicas são integradas a circuitos eletrônicos em um mesmo encapsulamento. O princípio de funcionamento baseia-se na variação de grandezas físicas internas, como capacitância, causada pelo deslocamento de uma massa inercial suspensa por microestruturas elásticas quando submetida a uma aceleração. Uma representação é apresentada na Figura 6.

Figura 6 – Princípio básico de Funcionamento de Sensores MEMS



Fonte: Adaptado de (AMERINI, 2016)

Aceleração provoca o deslocamento da massa interna, alterando a distância entre placas capacitivas. Essa variação é convertida em um sinal elétrico proporcional à aceleração aplicada. A leitura é normalmente realizada em três eixos ortogonais, permitindo a medição da aceleração no espaço tridimensional.

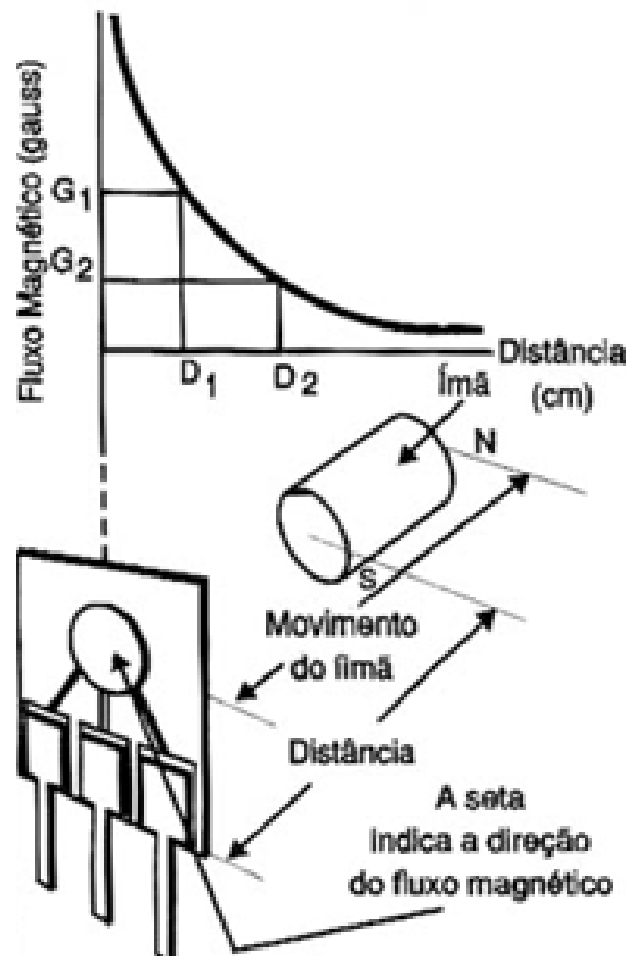
Uma solução comercial de baixo-custo para acelerômetros é o MPU6050 que integra, em um único encapsulamento, um acelerômetro e um giroscópio triaxial, um sensor de temperatura, um conversor Conversor Analógico-Digital (ADC) e um processador interno dedicado ao tratamento dos dados, disponibilizando uma saída I2C (INVENSENSE, 2013). Tais fatores fazem com que este seja um dos sensores mais utilizados em projetos protótipos.

2.3.2.3 Efeito Hall

Sensores baseados no efeito Hall são amplamente utilizados para detecção de campo magnético, posição, velocidade e corrente elétrica em sistemas embarcados e industriais. Sua aplicação é recorrente em ambientes automotivos, especialmente em medições de rotação, detecção de proximidade e monitoramento de sistemas elétricos.

O princípio de funcionamento baseia-se no efeito Hall, fenômeno físico observado quando um condutor ou semicondutor percorrido por corrente elétrica é submetido a um campo magnético perpendicular ao fluxo de carga. Nessas condições, ocorre o acúmulo de cargas em uma das faces do material, gerando uma diferença de potencial transversal, denominada tensão de Hall (WILSON, 2005). A Figura 7 exemplifica um uso do sensor de efeito Hall.

Figura 7 – Funcionamento do sensor Hall em relação a distância do ímã



Fonte: (INSTITUTO NCB, s.d.)

Observa-se que, quanto mais perto do ímã, maior o fluxo magnético percebido pelo sensor e, conseqüentemente, maior a diferença potencial e tensão de saída. Com isso, pode-se desenvolver aplicações em eixos girantes, como motores, nos quais é criado um gatilho para disparar quando o fluxo magnético atinge um limiar definido, contabilizando um pulso. Com a contagem de pulsos sendo realizada em intervalos fixos de tempo, é possível determinar as rotações por minuto (RPM) do sistema, o que é fundamental para monitoramento de condição de funcionamento de motores a combustão.

Além disso, é possível utilizar sensores de efeito Hall para determinação da velocidade linear do veículo, utilizando o princípio da relação entre movimento rotacional e translacional.

A velocidade linear v está relacionada à velocidade angular ω pela Equação 2.1:

$$v = \omega R \quad (2.1)$$

em que R representa o raio efetivo do pneu.

No entanto, em aplicações automotivas, a velocidade angular não é medida diretamente. Em vez disso, utiliza-se um sensor acoplado a um eixo girante ou à roda fônica do pneu, com N ímãs ou dentes, gerando pulsos elétricos de frequência f proporcional à rotação da roda.

A velocidade angular pode ser expressa em função da frequência dos pulsos conforme a Equação 2.2:

$$\omega = \frac{2\pi f}{N} \quad (2.2)$$

Substituindo a Equação 2.2 em 2.1, obtém-se a expressão para a velocidade linear em função da frequência medida:

$$v = \frac{2\pi R}{N} \cdot f \quad (2.3)$$

Dessa forma, a velocidade do veículo pode ser determinada diretamente a partir da frequência dos pulsos gerados pela roda fônica.

Com o funcionamento do sensor e método de conversão de pulso para variáveis desejadas conhecidos, a integração em sistemas embarcados torna-se simples. Para as opções comerciais mais populares, sensores digitais de efeito Hall, como o A3144 (ALLEGRO, s.d.), apresentam interface simples, utilizando três pinos: alimentação (5V), terra e saída lógica compatível com microcontroladores. Contudo, por possuírem saída em coletor aberto, é recomendável utilizar resistor de *pull-up* de 10k Ω para evitar flutuações no sinal.

Concluída a fundamentação dos sensores, passa-se à teoria de comunicação sem fio e telemetria, que são essenciais para o monitoramento em tempo real e a análise de desempenho durante as provas.

2.4 Comunicação Sem Fio e Telemetria

Para sistemas embarcados em aplicações móveis, a comunicação sem fio é indispensável para viabilizar o uso em tempo real das informações coletadas. A escolha da tecnologia empregada impacta diretamente o desempenho do sistema, exigindo a consideração de parâmetros como alcance, taxa de transmissão e frequência de envio dos dados, que influenciam na viabilidade e na confiabilidade da solução.

Nos tópicos seguintes, são apresentados os conceitos básicos de telemetria, seguidos de uma análise comparativa entre tecnologias de comunicação sem fio utilizadas em sistemas embarcados. Por fim, são discutidos os princípios da tecnologia LoRa e sua aplicação em sistemas de telemetria veicular.

2.4.1 Conceitos de Telemetria em Sistemas Embarcados

Telemetria pode ser definida como a técnica de aquisição e transmissão remota de dados, permitindo o monitoramento de variáveis físicas sem a necessidade de acesso direto ao sistema monitorado. Em sistemas embarcados móveis, a telemetria desempenha papel central ao possibilitar a análise de desempenho, diagnóstico de falhas e acompanhamento de parâmetros operacionais em tempo real ou quase real.

No contexto veicular, especialmente em aplicações experimentais e de competição, a telemetria permite a coleta de informações relevantes durante a operação do veículo, sem interferir diretamente em sua condução. Essa abordagem amplia a capacidade de análise e tomada de decisão, tanto durante testes quanto em condições reais de uso.

Diversas tecnologias de comunicação sem fio podem ser empregadas em sistemas de telemetria, cada uma apresentando características próprias em termos de alcance, taxa de dados, consumo energético e robustez. Entre as mais comuns destacam-se Wi-Fi, Bluetooth, ZigBee e LoRa.

A Tabela 2 apresenta uma comparação conceitual entre essas tecnologias, considerando alcance e consumo de energia, parâmetros frequentemente utilizados na avaliação de soluções para sistemas embarcados móveis.

Tabela 2 – Comparação entre tecnologias de comunicação sem fio

Parâmetro	Bluetooth	Wi-Fi	ZigBee	LoRa
Frequência de operação	2.4 GHz	2.4 GHz 5 GHz 6 GHz	868 MHz (EU) 915 MHz (EUA) 2.4 GHz	433 MHz 868 MHz (EU) 915 MHz (EUA)
Taxa de dados	1 Mbps	0.1-54 Mbps	250 Kbps 100 Kbps 40 Kbps 20 Kbps	30-50 Kbps
Alcance	10-100 m	100 m	50-500 m	10 km
Potência de transmissão	0-10 dBm	15-20 dBm	(-25)-0 dBm	7-20 dBm

Fonte: (NAIK; REDDY; MANDELA, 2024)

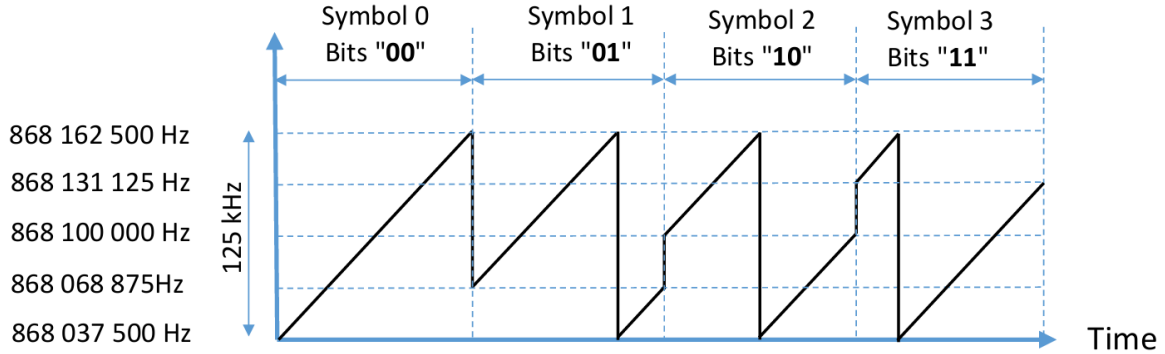
Tecnologias como Wi-Fi e Bluetooth oferecem elevadas taxas de transmissão, porém apresentam alcance limitado e maior consumo energético. O Zigbee ocupa uma camada intermediária, apresentando bom alcance e velocidade significativa. Por fim, o protocolo LoRa prioriza baixo consumo e maior alcance, sendo o mais adequados a aplicações de monitoramento distribuído e telemetria. Por tal motivo, na próxima seção, seu funcionamento será abordado com maior detalhes.

2.4.2 Princípios da Comunicação LoRa

A tecnologia LoRa utiliza a modulação *Chirp Spread Spectrum* (CSS), na qual os símbolos são transmitidos por meio de variações lineares de frequência ao longo do tempo, denominadas *chirps*. Nesse método, a informação é distribuída em uma faixa de frequência mais ampla, aumentando a robustez do sinal frente a interferências e ruídos. Essa característica permite que sinais de baixa potência sejam corretamente detectados pelo receptor, possibilitando comunicações a longas distâncias (MONTAGNY, 2022).

A Figura 8 ilustra o comportamento típico de um sinal modulado em CSS, evidenciando a variação contínua da frequência ao longo do tempo.

Figura 8 – Modulação LoRa com 868,1 MHz e largura de banda de 125 kHz (caso teórico)



Fonte: (MONTAGNY, 2022)

Na representação acima, uma configuração SF2 é utilizada apenas para fins ilustrativos, visto que os valores disponíveis no padrão LoRa variam de SF5 a SF12, nos quais SF representa o fator de espalhamento e é o parâmetro que define o número de estados possíveis de um símbolo e influencia diretamente o tempo de símbolo e a taxa de transmissão.

Para codificar a informação, é necessário analisar a variação da frequência ao longo do tempo, dentro dos limites da largura de banda. No exemplo, para cada combinação de bits (00, 01, 10 e 11), é definida uma frequência inicial distinta, selecionando o valor transmitido pelo chirp.

Além disso, a relação entre o fator de espalhamento e a largura de banda influencia no período do símbolo, conforme apresentado na Equação 2.4.

$$T_s = \frac{2^{SF}}{BW} \quad (2.4)$$

em que T_s representa o tempo de símbolo e BW a largura de banda.

Portanto, o aumento do SF resulta em maior tempo de símbolo, melhorando a sensibilidade do receptor e o alcance da comunicação. Em contrapartida, reduz a taxa de transmissão de dados, visto que o chirp demora mais a ser transmitido.

Nesse viés, surgem métricas para caracterizar a qualidade do enlace, como o Indicador de Força do Sinal Recebido (Received Signal Strength Indicator (RSSI)), que é uma medida da potência do sinal recebido pelo receptor convertido para dBm (LORINCZ et al., 2025).

A título de comparação, para obter um valor teórico de RSSI, pode-se utilizar o Modelo de Perda de Percurso, expresso pela Equação 2.5:

$$RSSI(d) = RSSI(d_0) - 10 n \log_{10} \left(\frac{d}{d_0} \right) \quad (2.5)$$

em que $RSSI(d_0)$ corresponde ao valor medido a uma distância de referência (3 a 5 metros), e n é o fator de perda de percurso, obtido em tabelas de referências ou calculando inversamente a partir de medições experimentais de RSSI.

Dessa forma, observa-se que os parâmetros de configuração do LoRa estabelecem compromissos entre alcance, robustez e taxa de transmissão. Em aplicações de telemetria embarcada, nas quais pequenas quantidades de dados são transmitidas de forma periódica, é comum priorizar maior sensibilidade e alcance em detrimento da taxa de dados. Além disso, como estes sistemas são classificados como *soft real-time*, o aumento do tempo de comunicação causado pela utilização de maiores valores de SF não compromete o funcionamento do sistema.

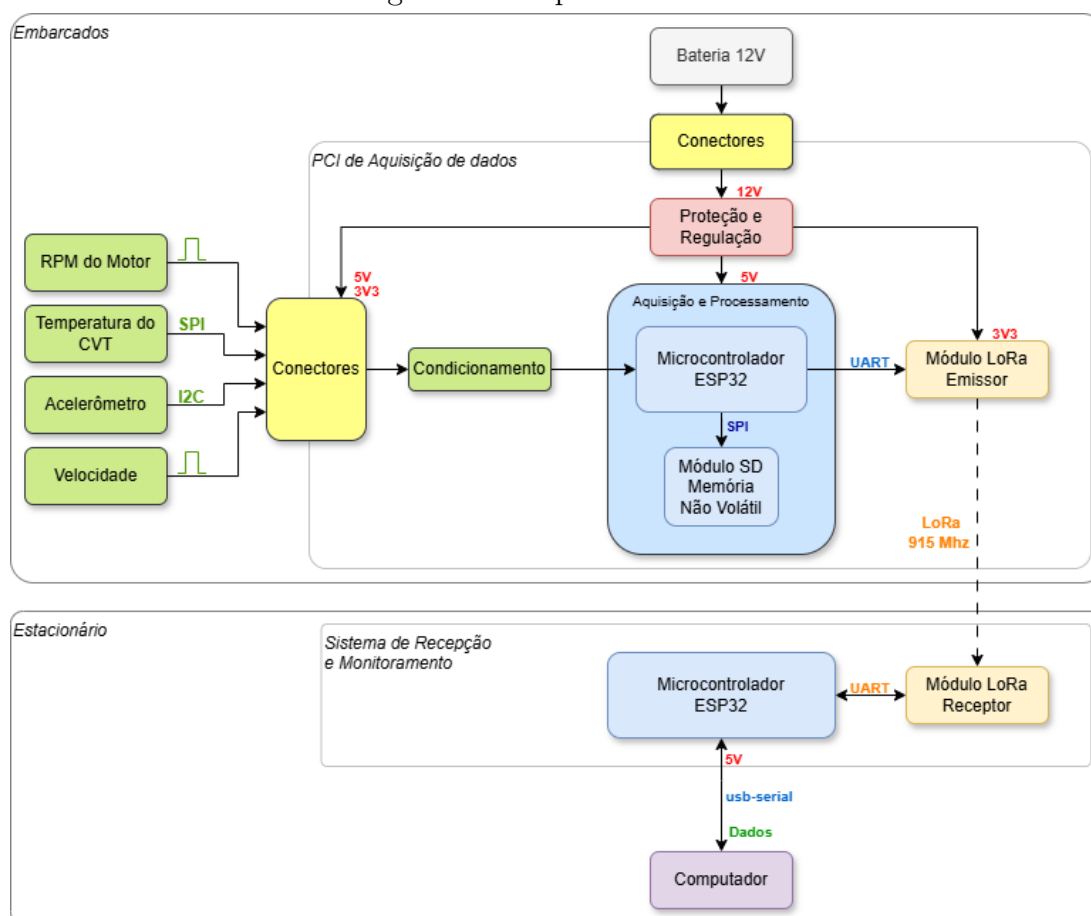
3 MATERIAIS E MÉTODOS

A seção tem como objetivo apresentar de forma técnica e objetiva as arquiteturas e funcionalidades do sistema projetado. Para isso, será evidenciado um panorama geral, seguido por seções que detalharão as principais decisões de hardware e software.

3.1 Arquitetura Geral do Sistema

No momento inicial, define-se a arquitetura de alto nível em diagramas de blocos, representado pela Figura 9, com o intuito de construir o esqueleto do projeto de acordo com os requisitos impostos.

Figura 9 – Arquitetura Geral



Fonte: Autoria própria.

O sistema proposto é composto por duas unidades principais: uma embarcada no veículo, responsável pela aquisição e transmissão dos dados, e outra estacionária, que atua como receptora.

A unidade embarcada realiza a leitura dos sensores através das interfaces digitais I2C, SPI e pulsos, o processamento das informações, por um microcontrolador que realiza a aquisição, organização e destina as informações à telemetria e armazenamento em memória não volátil (Cartão SD). Por estar embarcada, a alimentação da unidade é feita por uma bateria de lítio de 12 V, que passa por estágios de proteção e regulação antes de fornecer energia aos módulos. Para os sensores, a conexão no diagrama foi ocultada para preservar a clareza das ligações.

Com as variáveis de interesse adquiridas, a informação é transmitida via UART pelo microcontrolador até o módulo transmissor, que utiliza modulação LoRa na faixa de 915 MHz, para comunicar com o módulo receptor.

Nesta unidade receptora, há um segundo microcontrolador, que é responsável pela decodificação das mensagens recebidas e atua como uma ponte serial, conectando-se a um computador para que a equipe possa analisar os parâmetros em tempo real.

Por fim, por questões de desempenho no sistema e complexidade, a unidade embarcada é considerada a principal, sendo o maior desenvolvimento do projeto voltado para ela. Com isso, nas subseções seguintes, seus módulos de hardware e firmware são descritos com maior nível de detalhamento. Por sua vez, a unidade receptora, por possuir um sistema mais simples, é abordada de forma menos evoluída.

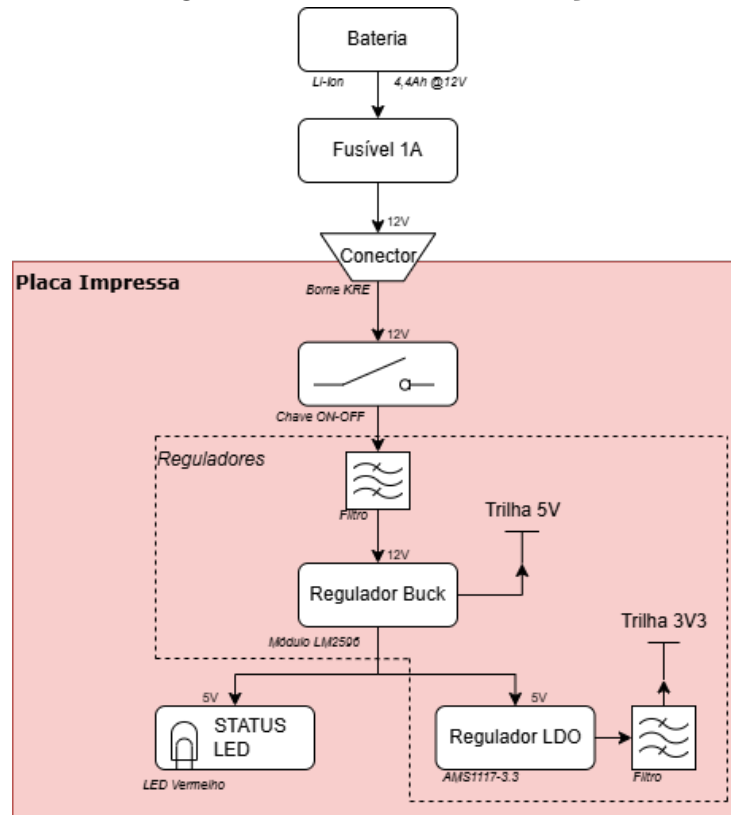
3.2 Materiais e Arquitetura de Hardware

O hardware do sistema embarcado é composto por uma Placa de Circuito Impresso (PCI) que integra o sistema de alimentação, a unidade de processamento (aquisição, organização dos dados dos sensores e armazenamento em cartão SD), o sistema de sensoriamento e o módulo de comunicação LoRa. Além disso, o sistema conta com sensores distribuídos pelo veículo e com uma unidade receptora dedicada à recepção dos dados transmitidos.

3.2.1 Sistema de Alimentação

O sistema de alimentação deve ser capaz de fornecer, de forma segura, a carga requisitada pelos componentes. Seu diagrama de blocos está apresentado na Figura 10.

Figura 10 – Bloco de Alimentação



Fonte: Autoria própria.

O sistema é alimentado por uma bateria de Li-Ion com capacidade nominal de 4,4 Ah e tensão nominal de 12 V, cuja carga é limitada por um fusível tipo fca de 1 A, antes de ser conectada à placa.

Na PCI, o acesso à alimentação é realizado por meio de um conector do tipo KRE, seguido de um interruptor liga/desliga responsável pela energização do sistema. Após a energização, a tensão passa por um capacitor cerâmico de 100 nF para desacoplamento e um capacitor de tântalo de 22 μ F para estabilidade, antes de ser aplicada ao módulo regulador chaveado buck LM2596.

Na saída do regulador, a tensão é ajustada para fornecer 5 V, alimentando o microcontrolador e dois sensores de efeito Hall, além de servir como entrada para o regulador linear AMS1117-3.3. Na próxima etapa, a tensão é convertida em 3,3 V pelo segundo regulador, passando por uma combinação de capacitores com os mesmos valores anteriormente utilizados, antes de alimentar os demais dispositivos embarcados.

Em seguida, com os módulos selecionados, foi realizada uma estimativa do consumo de potência do sistema com base nas especificações elétricas dos principais componentes utilizados. A Tabela 3 apresenta as tensões de operação, as correntes de pico e as respectivas potências máximas associadas às trilhas de 3,3 V e 5 V.

Tabela 3 – Estimativa de potência máxima consumida pelos componentes

Componente	Tensão (V)	I_{Pico} (mA)	P_{Pico} (mW)
Trilha de 3,3 V			
Módulo LoRa (EBYTE, 2020)	3,3	110	363
Módulo Cartão SD (CATALEX, s.d.)	3,3	200	660
Acelerômetro MPU6050	3,3	4,0*	13,2
Amplificador MAX6675 (MAXIN INTEGRATED, 2021)	3,3	1,5	4,95
<i>Total na trilha 3,3V</i>	<i>3,3</i>	<i>315,5</i>	<i>1.041,15</i>
Trilha de 5,0 V			
Microcontrolador ESP32 DevKit V1	5,0	250,0*	1.250
(x2) Sensor Hall - A3144 (ALLEGRO, s.d.)	5,0	18,0	90
LED de Status (VISHAY, s.d.)	5,0	15,0	75
Trilha de 3,3 V	5,0	347**	1.735**
<i>Total na trilha 5 V</i>	<i>5,0</i>	<i>630,0</i>	<i>3.150</i>

Fonte: Autoria própria.

Nota: * Valor estimado por falta de informações oficiais.

Nota: ** Rendimento para regulador linear é dado por V_{OUT}/V_{IN} . Portanto, $\eta_{AMS1117} = 0,66$.

Conforme apresentado acima, a trilha de 5 V apresenta um consumo máximo estimado de 3.150 mW. Considerando que essa tensão é obtida a partir de um regulador buck LM2596, com eficiência aproximada de 80%, é possível estimar a potência requerida na entrada do regulador, ou seja, na bateria de 12 V. Essa relação é expressa pela Equação 3.1.

$$P_{12V} = \frac{P_{5V}}{\eta_{LM2596}} = \frac{3.150 \text{ mW}}{0,8} = 3.937,5 \text{ mW} \quad (3.1)$$

A partir da potência demandada da bateria, a corrente total do sistema pode ser determinada por meio da Equação 3.2, permitindo a estimativa da autonomia do conjunto de alimentação.

$$I_{Total} = \frac{P_{12V}}{V_{Bateria}} = \frac{3.937,5 \text{ mW}}{12 \text{ V}} \approx 328 \text{ mA} \quad (3.2)$$

Com o consumo de corrente pelo sistema e considerando uma bateria com capacidade nominal de 4,4 Ah, a autonomia teórica do sistema pode ser estimada pela Equação 3.3.

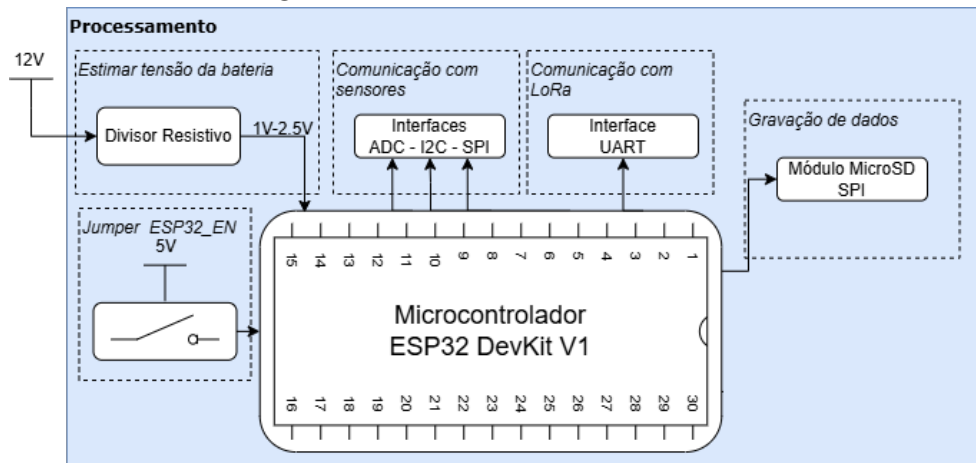
$$t_{\text{autonomia}} = \frac{C_{\text{bateria}}}{I_{\text{Total}}} = \frac{4,4 \text{ Ah}}{0,328 \text{ A}} \approx 13,4 \text{ h} \quad (3.3)$$

O resultado indica uma duração aproximada de 13,4 horas de operação contínua em condição de consumo máximo. Ressalta-se que esse valor representa uma estimativa ideal, podendo ser reduzido na prática devido a fatores como variações de carga, eficiência dos conversores e temperatura de operação.

3.2.2 Unidade de Processamento

A unidade de processamento é responsável pela aquisição dos dados provenientes dos sensores, seu tratamento e a distribuição das informações aos demais módulos do sistema. A Figura 11 ilustra o bloco de processamento.

Figura 11 – Bloco de Processamento



Fonte: Autoria própria.

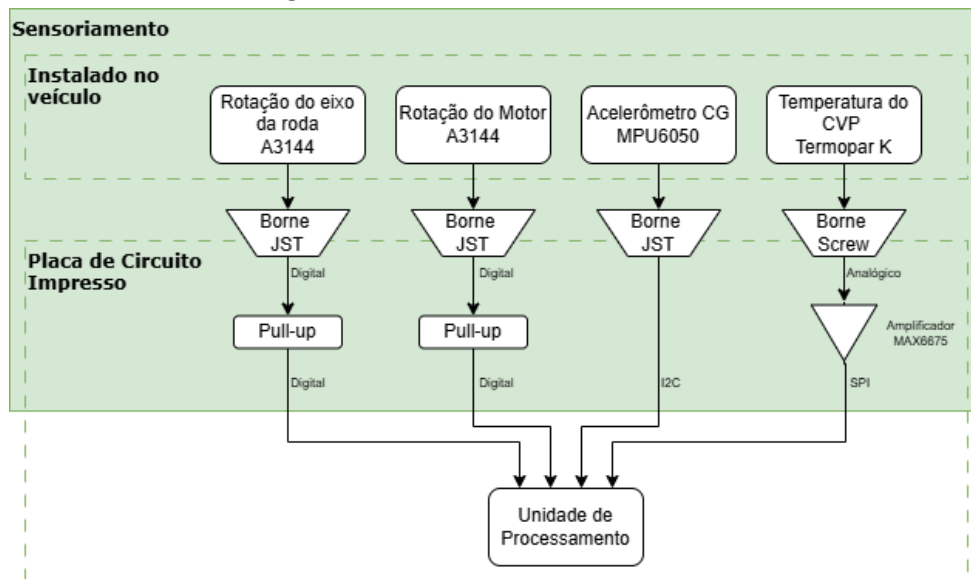
O sistema é baseado na placa ESP32 DevKit V1, que incorpora um microcontrolador de dois núcleos operando a 3,3 V. Esse dispositivo é responsável pela interface com diferentes protocolos de comunicação, incluindo ADC, I2C, SPI e UART. Além disso, os dados processados são transmitidos ao módulo LoRa e armazenados em um cartão MicroSD.

Adicionalmente, o circuito incorpora um sistema de estimativa da tensão da bateria por meio de um divisor resistivo conectado a uma entrada analógica, e um jumper na trilha de alimentação do microcontrolador, que permite isolar o ESP32 em momentos de depuração e configuração.

3.2.3 Sistema de Sensoriamento

O sistema de sensoriamento é projetado para estimar os parâmetros de velocidade do veículo, RPM do motor, temperatura da Caixa de Transmissão Variável (CVT) e aceleração nos eixos x, y e z. A Figura 12 apresenta o diagrama em blocos do sistema de sensoriamento, destacando a interligação entre os sensores, os circuitos de interface e a unidade de processamento.

Figura 12 – Bloco de Sensoriamento



Fonte: Autoria própria.

A medição da velocidade do veículo e do RPM do motor é realizada utilizando sensores de efeito Hall do modelo A3144, instalados próximos a ímãs permanentes acoplados aos respectivos eixos. Como esses sensores possuem saída do tipo coletor aberto, são empregados resistores de *pull-up* conectados entre a saída e a tensão de alimentação de 3,3 V, garantindo níveis lógicos compatíveis com as entradas digitais do ESP32. Os sinais resultantes são conectados a pinos configurados para detecção de borda, possibilitando a contagem de pulsos associada à rotação dos eixos monitorados.

O monitoramento da temperatura do CVT é realizado por meio de um termopar do tipo K, cuja interface com o microcontrolador é implementada utilizando o circuito integrado MAX6675. Esse dispositivo realiza a compensação de junta fria e a conversão do sinal do termopar em dados digitais, disponibilizados por meio de uma interface SPI.

Vale observar que, nas primeiras $N - 1$ leituras válidas, o sistema utiliza o valor instantâneo do termopar enquanto o *buffer* não é completamente preenchido, adotando a média apenas a partir da N -ésima amostra.

A medição das acelerações nos eixos x, y e z é realizada com o acelerômetro triaxial MPU6050, conectado ao ESP32 por meio da interface I2C. O dispositivo é instalado próximo ao centro de massa do veículo, minimizando a influência de vibrações locais e proporcionando uma estimativa mais representativa da dinâmica veicular.

A Tabela 4 apresenta um resumo dos sensores empregados no sistema, incluindo a grandeza medida, o modelo do dispositivo e o respectivo local de instalação no veículo.

Tabela 4 – Sensores utilizados no sistema de sensoriamento

Grandeza	Sensor/Interface	Local de Instalação
Velocidade	Sensor de efeito Hall A3144	Eixo da roda
RPM	Sensor de efeito Hall A3144	Eixo do motor
Temperatura	Termopar tipo K com MAX6675	Carcaça da CVT
Aceleração	Acelerômetro MPU6050	Próximo ao centro de massa

Fonte: Autoria própria.

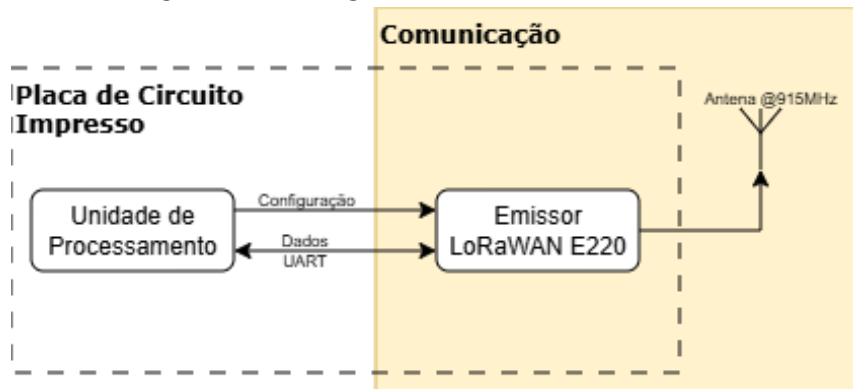
Os sensores são integrados ao sistema de forma a garantir a aquisição das grandezas de interesse, utilizando interfaces compatíveis com o ESP32 e coletando dados relevantes para a tomada de decisão da equipe.

3.2.4 Comunicação Sem Fio LoRa

O subsistema de comunicação sem fio é implementado utilizando o módulo LoRa E220-900T22D, operando na faixa de 915 MHz, com potência de transmissão configurável entre 10 e 22 dBm e alcance nominal de até 10 km, em condições ideais (EBYTE, 2020). O módulo é responsável pela transmissão dos dados adquiridos pela unidade de processamento para a estação receptora.

A Figura 13 apresenta o diagrama de hardware do subsistema de comunicação, evidenciando a integração entre o ESP32, o módulo LoRa e a antena externa.

Figura 13 – Diagrama subsistema transmissor



Fonte: Autoria própria.

A comunicação entre o ESP32 e o módulo LoRa é realizada por meio da interface UART, além dos pinos de configuração do dispositivo. Além disso, o módulo é alimentado com tensão de 3,3 V, provida pelos reguladores da placa.

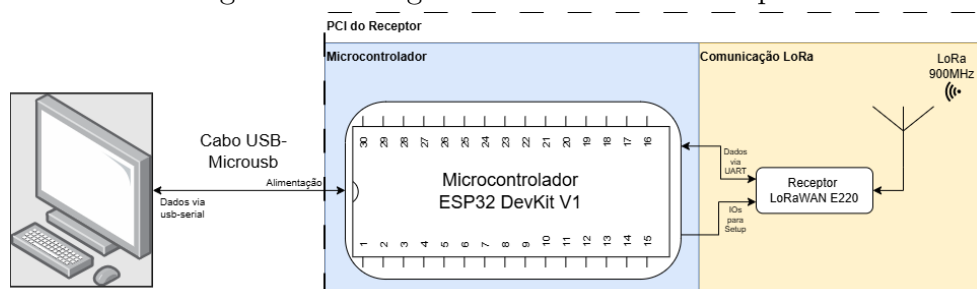
Para garantir a estabilidade da alimentação, utiliza-se um capacitor de desacoplamento de alta frequência cerâmico de 100 nF, posicionado o mais próximo possível do pino de alimentação do módulo LoRa.

A conexão com a antena externa é realizada por meio de um conector de radio-frequência do tipo SMA. São utilizadas duas antenas omnidirecionais compatíveis com as faixas de 868 MHz e 915 MHz, com ganho nominal de 6 dBi, sendo uma instalada no nó transmissor embarcado no veículo e a outra na estação receptora.

3.2.5 Bloco do Receptor

De maneira análoga à Seção 3.2.4, o bloco do receptor é composto por um microcontrolador ESP32 integrado a um módulo LoRa E220-900T22D, responsável pela recepção das mensagens transmitidas pelo nó embarcado no veículo. O diagrama de hardware do subsistema receptor é apresentado na Figura 14.

Figura 14 – Diagrama do subsistema receptor



Fonte: Autoria própria.

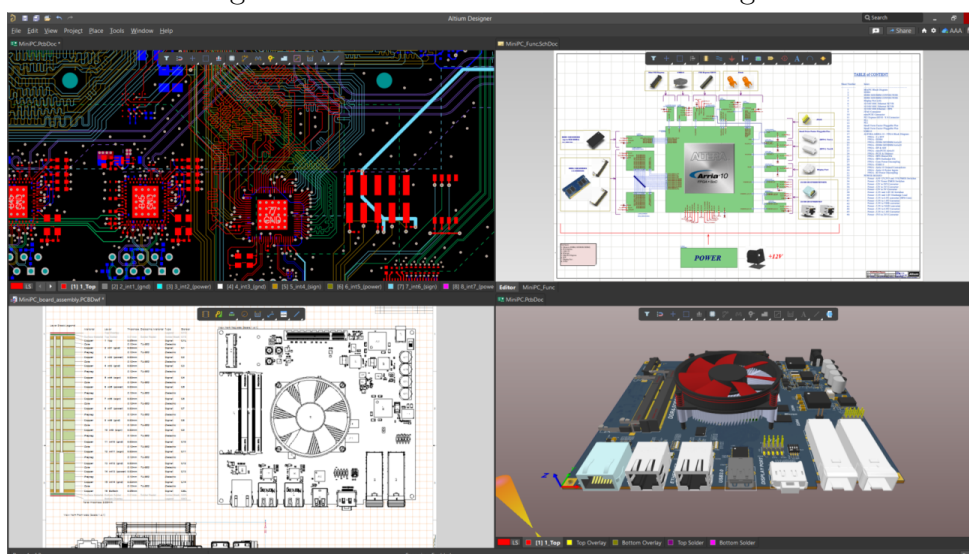
Nessa configuração, o computador é conectado ao microcontrolador por meio de um cabo USB–MicroUSB, responsável tanto pela alimentação do sistema quanto pela comunicação serial. As mensagens recebidas pelo módulo LoRa são transferidas ao ESP32 via interface UART e, posteriormente, encaminhadas ao computador para visualização e armazenamento.

Dessa forma, o subsistema de comunicação é implementado atendendo aos requisitos necessários para a transmissão dos dados. Os aspectos relacionados à configuração e ao protocolo de comunicação são apresentados na seção seguinte.

3.2.6 Ferramentas de Projeto de Hardware

O desenvolvimento do hardware do sistema, incluindo a elaboração dos esquemáticos e o projeto do layout da PCB, é realizado utilizando o software Altium Designer, exibido na Figura 15.

Figura 15 – Plataforma do Altium Designer



Fonte: (ALTIUM, 2024).

A ferramenta permite a captura esquemática dos circuitos, a associação dos componentes aos respectivos *footprints*, o roteamento das trilhas da placa e a visualização tridimensional do conjunto. Além disso, disponibiliza recursos de *Design Rule Check* (DRC), que contribuem para a verificação das regras de projeto e para a confiabilidade do circuito desenvolvido (ALTIUM, 2024).

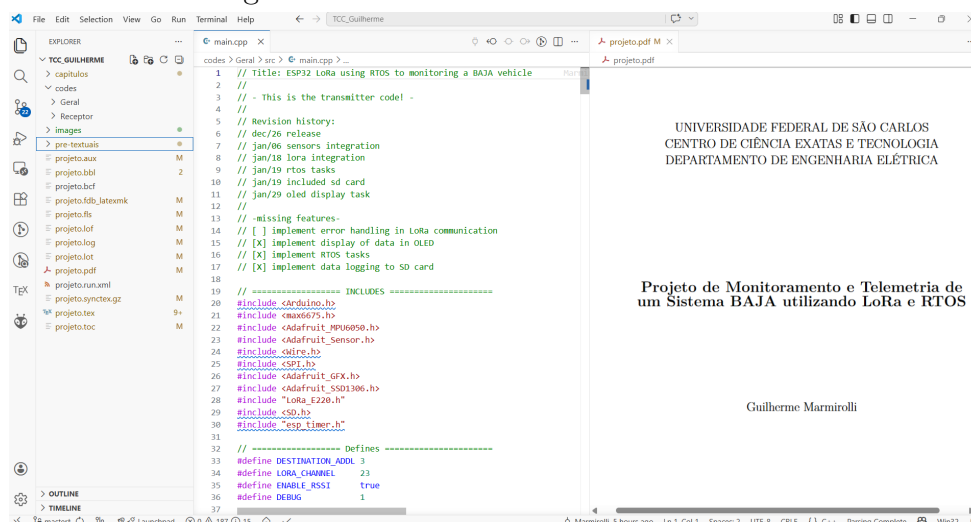
3.3 Arquitetura de Software

Esta seção apresenta os detalhes da arquitetura de software, abrangendo o ambiente de desenvolvimento, a estrutura do firmware embarcado na placa de aquisição e telemetria e a organização do software do receptor.

3.3.1 Ambiente de Desenvolvimento

O desenvolvimento do firmware é realizado no ambiente Visual Studio Code, que oferece recursos de edição, depuração e integração com extensões úteis ao projeto. A ferramenta é apresentada na Figura 16.

Figura 16 – Ambiente Visual Studio Code



Fonte: Autoria própria.

Nesse ambiente, ocorre a integração das etapas de escrita e programação de microcontroladores por meio de extensões específicas. O processo de compilação, incluindo o gerenciamento de dependências e o carregamento do código no microcontrolador, é realizado por meio da extensão *PlatformIO*. Dentro dessa plataforma, é possível adicionar bibliotecas que facilitam a interface com os dispositivos de hardware. A Tabela 5 apresenta as bibliotecas empregadas e suas respectivas funções.

Tabela 5 – Bibliotecas utilizadas no desenvolvimento do firmware

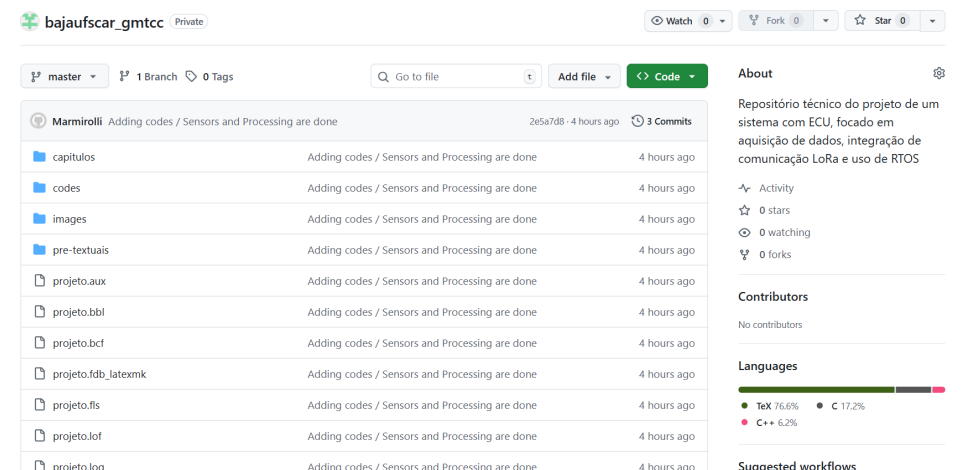
Biblioteca	Versão	Interface
adafruit/MAX6675 library	1.1.2	Amplificador MAX6675
adafruit/Adafruit MPU6050	2.2.6	Acelerômetro MPU6050
xreef/EByte LoRa E220 library	1.0.8	Comunicação LoRa

Fonte: Autoria própria.

Adicionalmente, utilizam-se as bibliotecas nativas do ambiente, como *Wire* e *SPI*, responsáveis pela comunicação com dispositivos baseados nos protocolos I2C e SPI, respectivamente. O sistema operacional de tempo real FreeRTOS (??), nativo do ESP32, também é empregado em parte do desenvolvimento para a organização das tarefas do firmware.

Para o controle de versões e armazenamento do código-fonte, utiliza-se o sistema Git, com hospedagem em repositório remoto na plataforma GitHub, conforme ilustrado na Figura 17.

Figura 17 – Repositório no GitHub



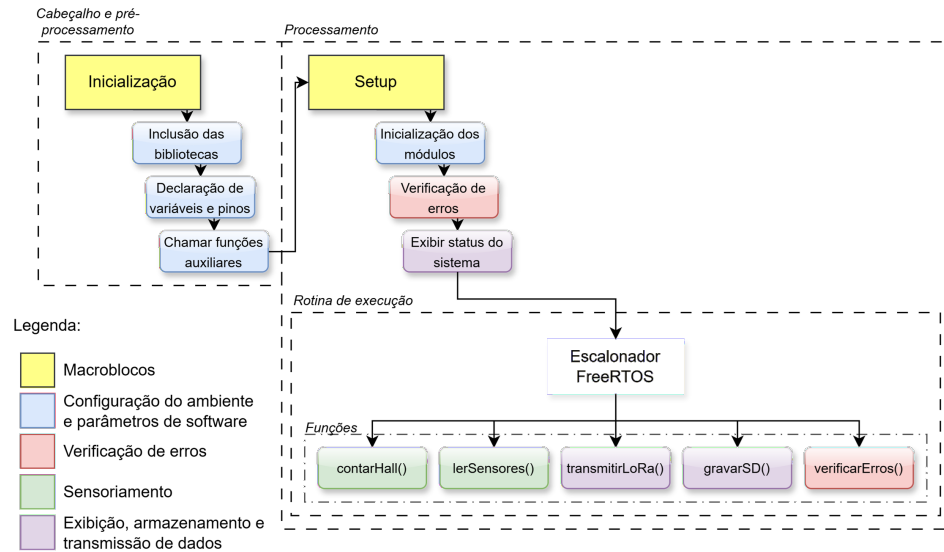
Fonte: Autoria própria.

O sistema possibilita o rastreamento das modificações e a organização do desenvolvimento, além de disponibilizar o código para futuros usuários.

3.3.2 Estrutura do Firmware Embarcado

A arquitetura de software baseia-se no FreeRTOS, responsável pelo escalonamento das tarefas da aplicação, nativo da plataforma do ESP32. A Figura 18 apresenta o diagrama de execução correspondente a esse modelo.

Figura 18 – Execução da aplicação no modelo baseado em RTOS

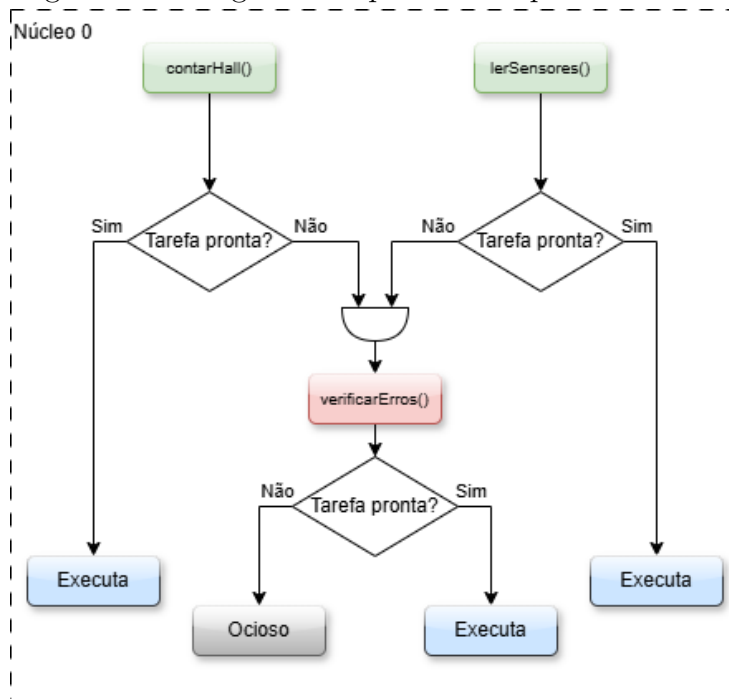


Fonte: Autoria própria.

A estrutura inicial da aplicação é semelhante ao modelo convencional, de super-lço, passando por rotinas de inicialização e configuração dos módulos a serem utilizados. Porém, no RTOS, as tarefas são organizadas em blocos de execução concorrentes, com prioridades definidas. O sistema operacional é responsável por gerenciar o escalonamento dessas tarefas, garantindo que as de maior prioridade sejam executadas quando prontas.

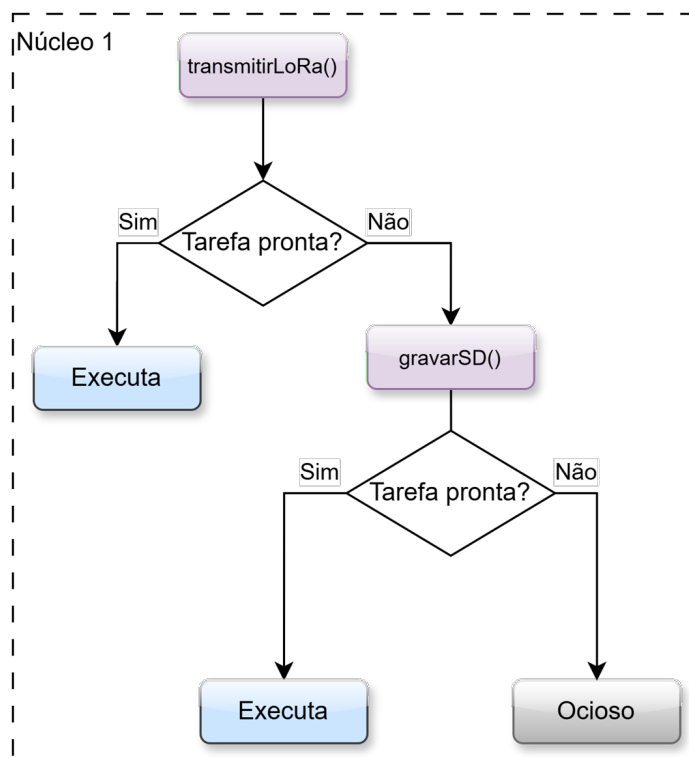
Além disso, o microcontrolador ESP32 apresenta arquitetura de dois núcleos de processamento, possibilitando a distribuição das tarefas entre eles, sem concorrência. Desse modo, as atividades relacionadas à aquisição, supervisão do sistema e ao processamento inicial dos sensores são atribuídas ao núcleo 0, enquanto as tarefas de comunicação e armazenamento de dados são alocadas no núcleo 1, como apresentado pelas Figuras 19 e 20.

Figura 19 – Diagrama de prioridades para o Núcleo 0



Fonte: Autoria própria.

Figura 20 – Diagrama de prioridades para o Núcleo 1



Fonte: Autoria própria.

No núcleo 0, as tarefas `contarHall` e `lerSensores` possuem a mesma alta prioridade, sendo executadas assim que estiverem prontas. Em caso de concorrência, utiliza-se o escalonamento *Round-Robin*. Caso nenhuma das duas estiverem prontas, a tarefa `verificarErros` é liberada a usar o núcleo, podendo ser interrompida caso alguma tarefa de maior prioridade fique pronta. Por fim, na ausência de tarefas prontas, o núcleo permanece em estado ocioso.

De maneira análoga, para o núcleo 1, a tarefa `transmitirLora` possui maior prioridade e é executada assim que estiver pronta, disponibilizando o núcleo para a tarefa `gravarSD` durante os intervalos entre as execuções. Por fim, o núcleo permanece ocioso quando não houver tarefas prontas.

3.4 Validação Experimental do Sistema

A validação experimental do sistema foi estruturada em três etapas sequenciais, partindo da verificação funcional em bancada até a simulação de operação em ambiente dinâmico. Em cada etapa, foram avaliadas diferentes funcionalidades, garantindo a operação correta dos módulos antes da integração do sistema.

O primeiro ensaio, realizado em bancada, destinou-se a confirmar a operação básica do sistema: comunicação LoRa, recepção serial e gravação no cartão SD. O transmissor e o receptor foram posicionados a uma distância de 3 m em ambiente livre de obstáculos, e o sistema foi mantido em operação durante 100 s. Monitorou-se a recepção dos dados no receptor e a gravação correta no cartão SD, garantindo o acionamento de todos os sensores ao menos uma vez. Os dados coletados serviram ainda como referência para comparações posteriores.

Para a validação da aquisição dos sensores, utilizou-se apenas o módulo transmissor, com gravação de dados no cartão SD. A Tabela 6 apresenta os sensores e suas respectivas configurações de teste.

Tabela 6 – Configuração de Teste dos Sensores

Medição	Referência A	Referência B
Temperatura	Gelo	Água fervente
RPM	Veículo comercial (CAN)	N/A
Velocidade	Veículo comercial (CAN)	N/A
Aceleração (3 eixos)	Gravidade	N/A

Fonte: Autoria própria.

O procedimento consistiu na aplicação de estímulos conhecidos e na comparação dos resultados com valores e tempos de resposta obtidos por instrumentos de referência (exceto para o acelerômetro), com o objetivo de caracterizar o funcionamento de cada sensor

Em seguida, a comunicação LoRa foi realizada em campo, avaliando o desempenho do enlace em função da distância por meio da análise de RSSI e da taxa de perda de pacotes. O teste foi conduzido em cenário sem obstáculos, semelhante ao da competição, em seis pontos de distância: 20, 50, 100, 200, 500 e 1000 m, estimados por GPS, com o receptor fixo em um ponto de referência e o transmissor movendo-se até cada posição definida e coletando dados por 120 s.

Definidos os procedimentos experimentais empregados na validação, o capítulo seguinte apresenta o desenvolvimento do sistema, detalhando as soluções de hardware e software implementadas, bem como a integração dos módulos que compõem a arquitetura de telemetria proposta.

4 DESENVOLVIMENTO

Este capítulo descreve o desenvolvimento do sistema de telemetria embarcada, abrangendo o projeto da placa eletrônica, o firmware e a configuração do receptor. São apresentadas as principais decisões de projeto e as soluções adotadas, com ênfase nos aspectos que impactam o desempenho e a confiabilidade do sistema.

4.1 Desenvolvimento da Placa Embarcada

A placa embarcada concentra em um único hardware o sistema de alimentação, a unidade de processamento, entrada de sensores e o módulo de comunicação. O projeto seguiu boas práticas de PCI, com planejamento de posição dos componentes, atenção aos capacitores de desacoplamento e organização dos planos de terra.

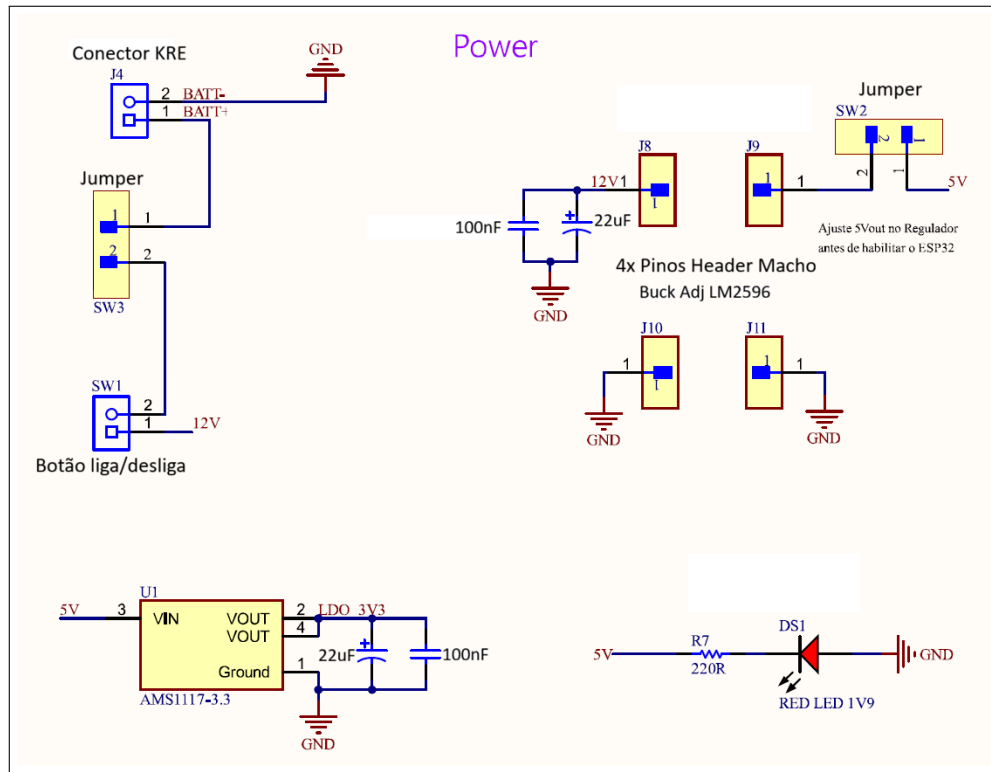
Durante o projeto, pelo uso constante de módulos comerciais como forma de barateio de projeto, muitas das vezes são utilizadas barras de pinos para solda do módulo na placa, atuando como um conector placa-a-placa e permitindo o uso dos componentes. Devido a isso, o esquemático apresentado pode não conter o módulo em si, mas sim o conector utilizado.

4.1.1 Sistema de Alimentação

O sistema de alimentação opera a partir de uma bateria de Lí-Ion 12 V com capacidade de 4,4 Ah, escolhida por robustez e custo, que distribui energia a dois barramentos: 5 V, destinado ao microcontrolador e aos sensores Hall, e 3,3 V, utilizado pelo módulo LoRa, acelerômetro, termopar e módulo SD.

A arquitetura adotada combina um conversor *buck* para a geração de 5 V com um regulador linear em cascata para 3,3 V, conforme a Figura 21. Essa abordagem equilibra eficiência e qualidade de tensão: o conversor chaveado absorve a maior parte da queda de tensão com alta eficiência, enquanto o regulador linear fornece uma saída de baixo ruído para os dispositivos mais sensíveis.

Figura 21 – Esquemático do bloco de alimentação da placa embarcada



Fonte: Autoria própria.

Nota: O LED está invertido propositalmente, pois seu *footprint* apresenta orientação invertida. Desta forma, a indicação visual na placa permanece correta.

A entrada de 12 V é realizada por um conector KRE, selecionado pela robustez mecânica e facilidade de conexão em campo, e passa por um jumper e uma chave liga/desliga. O jumper tem papel de redundância e segunda opção de controle de alimentação, visto que o botão de liga/desliga é mantido fixo no receptáculo e é substituído por uma ligação direta durante testes de bancada.

Em seguida, um estágio de filtragem, composto por um capacitor de tântalo de 22 μ F e um cerâmico de 100 nF, antecipa o barramento de 12 V na entrada do primeiro regulador, atenuando ruídos de alta frequência e transientes de tensão.

A conversão para 5 V é realizada pelo regulador chaveado LM2596 (TEXAS INSTRUMENTS, 2023), de custo acessível e amplamente utilizado em protótipos, que suporta tensão de entrada de até 40 V e corrente de saída de até 3 A com eficiência tipicamente superior a 80%, sendo adequado para a tensão de bateria utilizada. A saída é disponibilizada por meio de um jumper de configuração, que permite ajustar e verificar a tensão antes de habilitar a alimentação ao restante do circuito. Por fim, um LED com resistor limitador de 220 Ω sinaliza a presença de tensão nesse barramento.

No primeiro regulador, a tensão de 5 V é convertida para 3,3 V através regulador linear AMS1117 (ADVANCED MONOLITHIC SYSTEMS, s.d.). Com queda de tensão típica de 1,3 V e corrente máxima de 1 A, o componente atende à demanda do barramento com margem adequada, além de ser amplamente utilizado e possuir um preço acessível. Dois capacitores de desacoplamento, 100 nF e 22 μ F, foram posicionados na saída conforme recomendado pelo fabricante.

A Tabela 7 apresenta as correntes estimadas de pico frente aos limites especificados em *datasheet*.

Tabela 7 – Correntes estimadas e especificações dos reguladores

Regulador	Corrente estimada de pico	Limites do datasheet
LM2596	$I_{IN} = 0,328 \text{ A} / I_{OUT} = 0,630 \text{ A}$	$I_{IN} \leq 7,5 \text{ A} / I_{OUT} \leq 3 \text{ A}$
AMS1117	$I_{IN} = 0,347 \text{ A} / I_{OUT} = 0,315 \text{ A}$	$I_{IN} \leq 1,5 \text{ A} / I_{OUT} \leq 1 \text{ A}$

Fonte: Autoria própria.

Nota: Os valores de tensão de entrada e saída são 12 V e 5 V, para o LM2596 e 5 V, e 3,3 V para o AMS1117.

Desta maneira, o sistema de alimentação é capaz de fornecer energia estável e confiável para todos os componentes da placa embarcada, garantindo a integridade do sistema mesmo em situação de consumo máximo.

4.1.2 Unidade de Processamento

A unidade de processamento é baseada no módulo ESP32 DevKit V1. O microcontrolador conta com dois núcleos operando a até 240 MHz e suporte nativo a RTOS, o que permite gerenciar e distribuir as tarefas entre os núcleos por meio de um escalonador. Adicionalmente, dispõe de interfaces SPI, I2C e UART necessárias para integrar todos os periféricos do projeto.

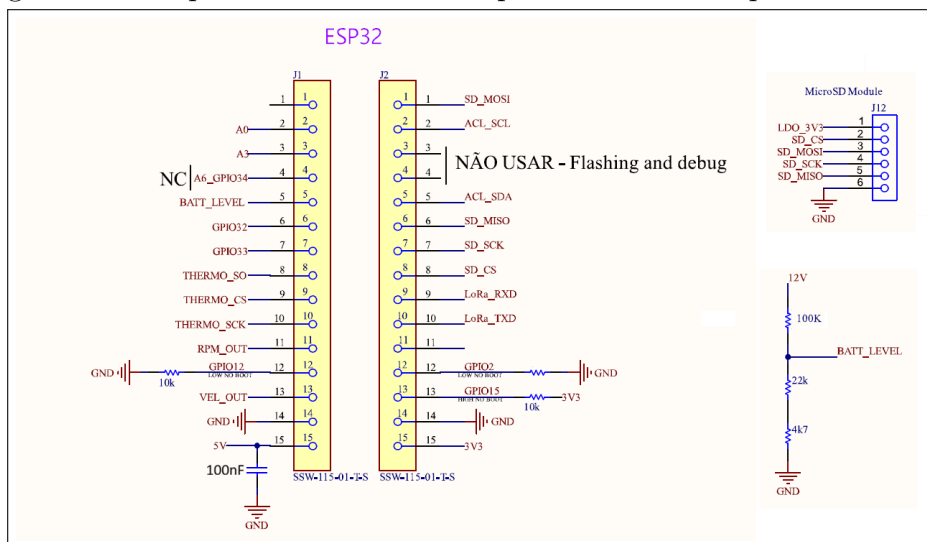
O armazenamento local de dados é realizado por um módulo de cartão microSD via protocolo SPI, que registra as informações de telemetria em arquivos texto. Esse recurso opera de forma independente da comunicação sem fio, garantindo a persistência dos dados mesmo em caso de falha no enlace LoRa e permitindo análise de eventuais falhas.

Para o monitoramento da bateria, foi implementado um divisor resistivo com resistores de 100 k Ω , conectado à linha 12V, e 22 k Ω mais 4,7 k Ω formando 26,7 k Ω , conectados ao terra, configurando assim uma redução da tensão num fator de aproximadamente 4,75. Com a bateria completamente carregada (12,6 V), a tensão no pino analógico do ESP32 é de aproximadamente 2,65 V, dentro do limite máximo de 3,3 V suportado pela entrada. O

ADC do ESP32 possui resolução de 12 bits, mas apresenta não-linearidade, o que limita a precisão das leituras (EMBARCADOS, 2019). Para os fins deste projeto, a estimativa do nível de bateria é suficiente dentro dessa restrição.

A Figura 22 apresenta o esquemático do bloco de processamento.

Figura 22 – Esquemático do bloco de processamento da placa embarcada



Fonte: Autoria própria.

A conexão do ESP32 na placa é realizada por meio de dois conectores do tipo header fêmea 15 pinos posicionados 28mm entre si. Desta forma, é possível conectar e desconectar o módulo facilmente. Além disso, um capacitor de desacoplamento de 100 nF foi adicionado próximo à entrada de alimentação de 5 V do módulo, para atenuar possíveis ruídos do regulador chaveado.

Para as conexões de sensores e periféricos, foram utilizados os pinos GPIO disponíveis, conforme o mapeamento apresentado na Tabela 8.

Tabela 8 – Mapeamento de pinos do ESP32 e suas conexões

Pino ESP32	Conectado com	Observação
GPIO5	SD_CS	Seleção do chip SPI
GPIO13	VEL_IN	Entrada digital de velocidade
GPIO14	RPM_IN	Entrada digital de RPM
GPIO16	LoRa_TXD	Canal UART com o módulo LoRa
GPIO17	LoRa_RXD	Canal UART com o módulo LoRa
GPIO18	SD_SCK	Interface SPI padrão (clock)
GPIO19	SD_MISO	Interface SPI padrão (dados)
GPIO21	ACL_SDA	Interface I2C padrão (dados)
GPIO22	ACL_SCL	Interface I2C padrão (clock)
GPIO23	SD_MOSI	Interface SPI padrão (dados)
GPIO25	THERMO_SO	Saída serial
GPIO26	THERMO_CS	<i>Chip Select</i>
GPIO27	THERMO_CLK	Sincronismo com o módulo
GPIO32	LoRa_M0	Pino de configuração
GPIO33	LoRa_M1	Pino de configuração
GPIO35	BATT_LEVEL	Entrada apenas leitura
GPIO36	LoRa_State	<i>Feedback</i> do módulo

As conexões visam a otimização de aproveitamento das interfaces padrões disponíveis no microcontrolador, facilitando a implementação do firmware.

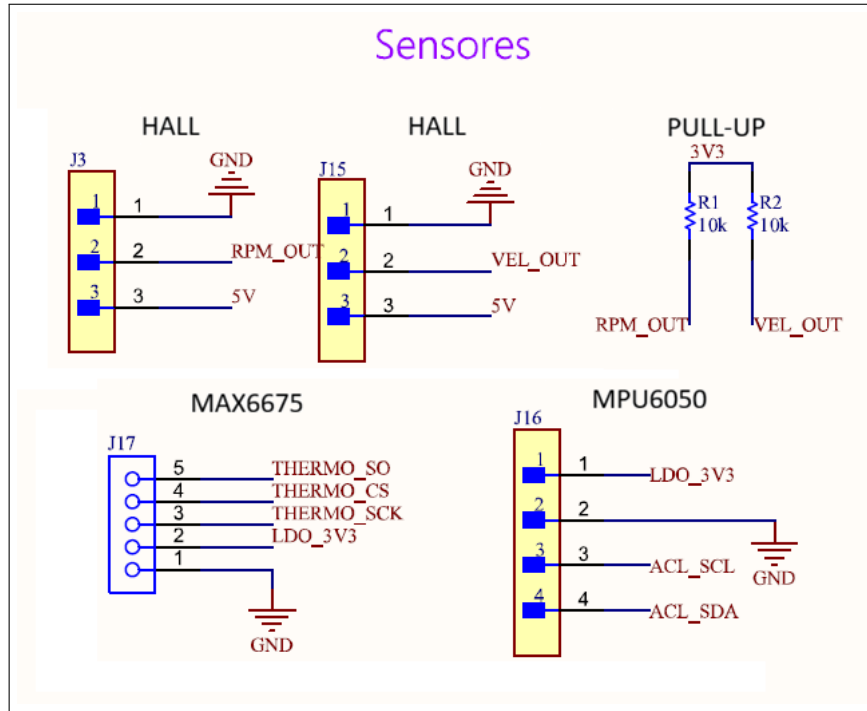
Pinos como GPIO2, GPIO12 e GPIO15 foram polarizados com resistores de *pull-up* e *pull-down* externos para garantir os níveis lógicos corretos durante a inicialização, assegurando a inicialização desejada. Além disso, os pinos GPIO1 e GPIO3, reservados para gravação e depuração via UART, foram mantidos propriamente desutilizados durante a aplicação. Pinos não citados não foram utilizados e não interferem no sistema.

Com o sistema de processamento definido, passa-se à etapa de desenvolvimento do sistema de sensoriamento.

4.1.3 Sistema de Sensoriamento

O sistema de sensoriamento é composto por sensores de RPM, velocidade, aceleração e temperatura. Pelo fato da maioria dos sensores estarem posicionados pelo veículo, o esquemático apresentado pela Figura 23, contém apenas os circuitos de conectores.

Figura 23 – Esquemático do bloco de sensoriamento da placa embarcada



Fonte: Autoria própria.

Os sensores Hall A3144 (ALLEGRO, s.d.) são alimentados com 5 V, satisfazendo sua faixa de operação, e fornecem uma saída digital que é diretamente conectada à entrada GPIO do ESP32. Para adquirir o valor de RPM, um dos sensores é posicionado dentro do motor, em local definido pelo fabricante.

O segundo sensor Hall é posicionado no semi-eixo traseiro do carro, que gira com a mesma variação angular que o pneu. Desta forma, ao utilizar a Equação 2.3, com raio efetivo do pneu de $R = 0,30$ m e com $N = 1$ imã, e partir da aquisição da frequência, estima-se a velocidade em:

$$v = \frac{2\pi \cdot 0,30}{1} \cdot f \quad (4.1)$$

Simplificando:

$$v \approx 1,8849 \cdot f \quad (4.2)$$

onde v é dado em m/s e f em Hz.

Para a conexão dos sensores à placa, foram utilizados conectores JST, que impossibilitam a conexão invertida, facilitam a manutenção e são amplamente acessíveis. Ademais, por apresentarem saída em coletor aberto, resistores de *pull-up* de $10\text{k}\Omega$ garantem nível lógico alto quando o sensor está inativo, evitando flutuações de sinal.

A medição de aceleração é realizada pelo sensor MPU6050, capaz de medir a aceleração linear nos eixos X, Y e Z. O dispositivo possui conversores A/D de 16 bits e faixa de medição configurável, o que o torna adequado tanto para medições de baixa dinâmica quanto para aplicações com maiores níveis de aceleração (INVENSENSE, 2013). Para o projeto, a faixa escolhida foi $\pm 4g$. Além disso, apresentando baixo custo, o módulo possui sensores MEMS e interface de comunicação via I2C. Para a aquisição adequada dos dados, o sensor é posicionado atrás do banco do motorista, região estimada como próxima ao centro de massa do veículo. Por fim, sua conexão à placa é realizada por meio de conector JST.

A temperatura do CVT é medida por um termopar Tipo K, cuja leitura é condicionada pelo circuito integrado MAX6675 (MAXIN INTEGRATED, 2021), responsável pela compensação da junta fria e pela conversão do sinal para formato digital, disponibilizado ao microcontrolador via interface SPI. O módulo amplificador é diretamente soldado na placa e conta com dois conectores parafusos para a ligação com os termopares. A partir desse conector, o sinal é conduzido por meio de um cabo blindado até a sonda de temperatura, posicionada no interior da Caixa de CVT, onde se concentra a condição térmica a ser aferida.

A escolha do termopar Tipo K se deve principalmente à sua robustez mecânica, associada à construção blindada da sonda. Adicionalmente, esse tipo de sensor é adequado à faixa de temperatura esperada na CVT, que pode atingir até 150 °C. Embora apresente uma resposta dinâmica mais lenta, tal característica não compromete a aplicação, uma vez que a variação de temperatura nesse subsistema ocorre de forma gradual.

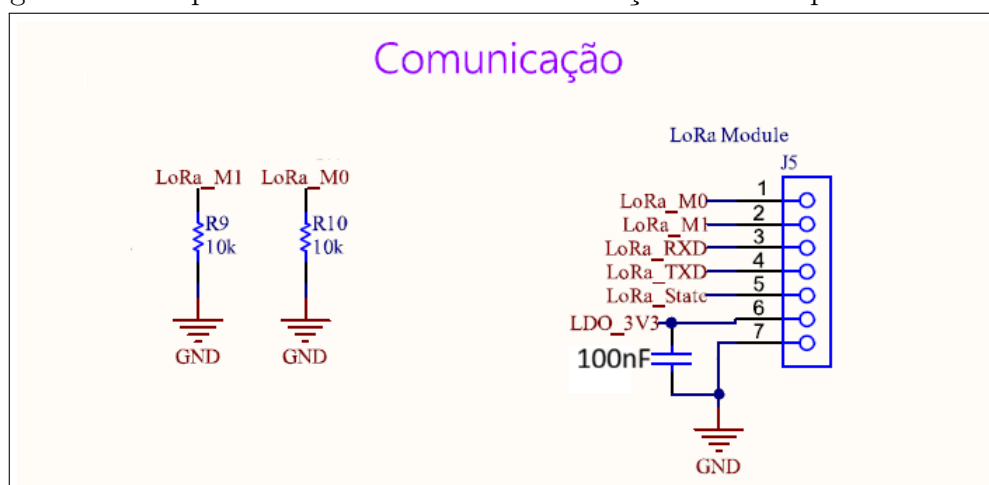
4.1.4 Módulo de Comunicação LoRa

A comunicação sem fio é realizada pelo módulo E220-900T22D (EBYTE, 2020), que opera na faixa de 915 MHz e utiliza a modulação LoRa. A escolha pelo módulo se justifica por atender à distância requisitada durante competições (aproximadamente 500 m), além da compatibilidade com a tensão de 3,3 V e do baixo consumo de energia.

No hardware, o módulo é conectado ao microcontrolador via UART, pelas linhas TXD e RXD. Os pinos de configuração M0 e M1 são mantidos em nível lógico baixo por resistores de *pull-down* de 10 k Ω , selecionando o modo de operação normal na inicialização. A alimentação em 3,3 V é filtrada por um capacitor de desacoplamento de 100 nF posicionado próximo ao pino de alimentação do módulo, minimizando ruídos de alta frequência.

A Figura 24 apresenta o esquemático do bloco de comunicação.

Figura 24 – Esquemático do bloco de comunicação LoRa da placa embarcada



Fonte: Autoria própria.

O módulo foi posicionado o mais distante possível do circuito de alimentação, de modo a reduzir a interferência proveniente do conversor chaveado. Além disso, o posicionamento adotado proporciona um caminho mais livre para a conexão da antena.

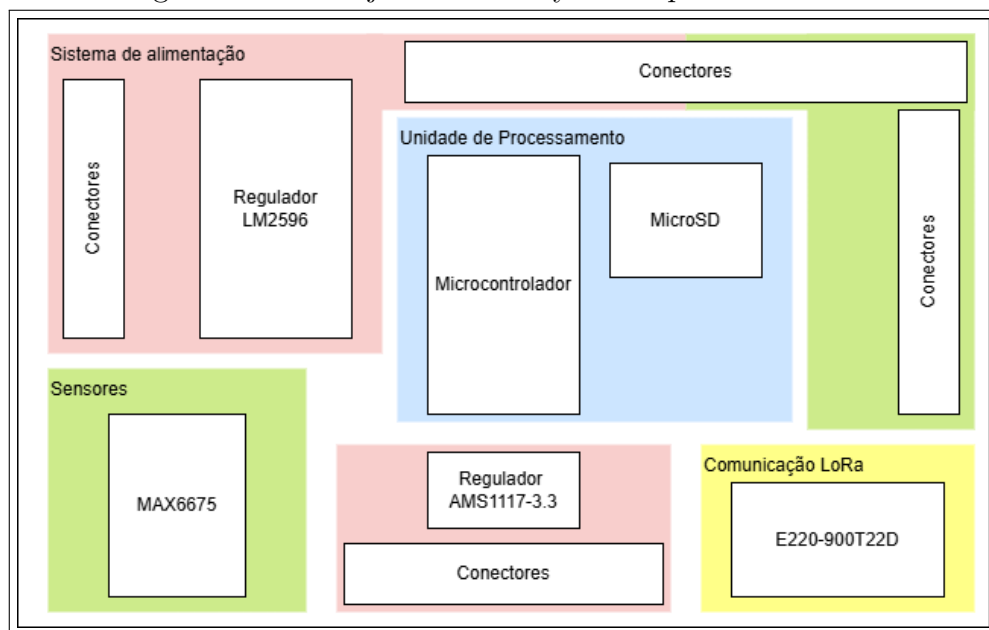
A antena é conectada ao módulo por meio de um conector SMA, garantindo uma interface mecânica robusta e baixa perda de sinal. Foi utilizada uma antena omnidirecional, com polarização vertical e ganho nominal de 6 dBi. Esse tipo de antena favorece o aumento do alcance na direção horizontal, sendo adequado para comunicação em ambientes abertos.

4.1.5 Considerações de Layout

Ao iniciar o desenvolvimento do layout, foram considerados critérios relacionados à disposição dos componentes e à mitigação de interferências eletromagnéticas, fatores que impactam diretamente o desempenho do sistema.

Com base nesses critérios, foi realizado o planejamento inicial de posicionamento dos componentes, apresentado na Figura 25.

Figura 25 – Planejamento de layout da placa embarcada



Fonte: Autoria própria.

O microcontrolador assume papel central no sistema e, por essa razão, foi posicionado no centro da placa. Ao lado, o módulo microSD foi colocado de forma a reduzir o comprimento das trilhas da interface SPI. O módulo LoRa, por sua vez, foi posicionado na borda, afastado do circuito de alimentação e, como utiliza interface UART, possui mais resistência a distância de trilhas. Por fim, os demais componentes foram organizados de modo a otimizar o aproveitamento do espaço disponível na placa.

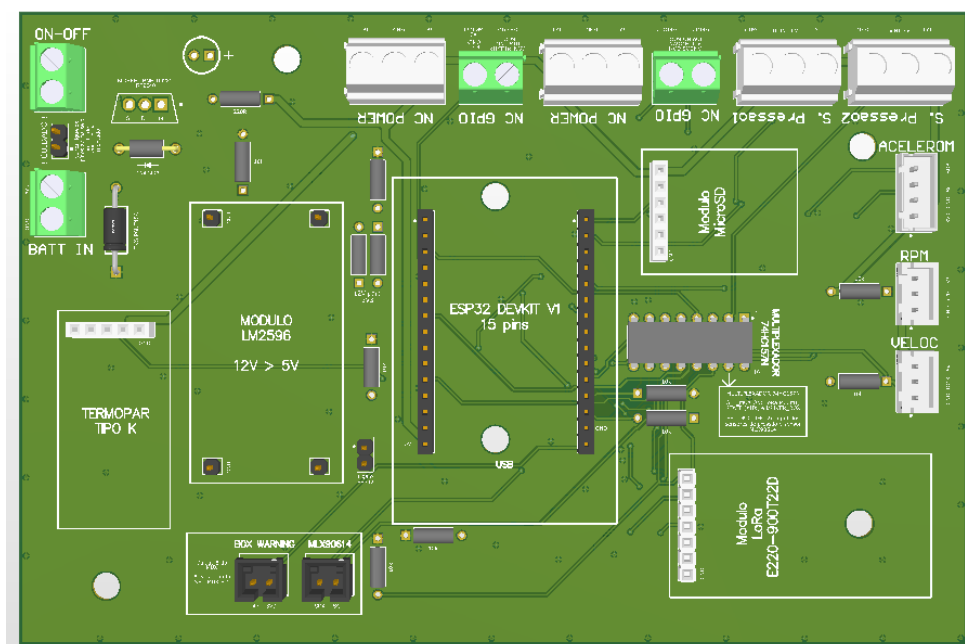
Ademais, alguns pinos de GPIO não utilizados, bem como linhas de alimentação (5 V, 3,3 V e terra), foram disponibilizados por meio de conectores do tipo KRE, permitindo futuras expansões sem necessidade de reprojetar a placa.

Com o planejamento das posições finalizadas, e estabelecido o uso de apenas duas camadas para baretear o processo, foram definidas as diretrizes de roteamento e distribuição de energia da placa. As larguras das trilhas foram dimensionadas com base na Tabela 7 e adicionando margem segura em relação aos valores necessários, resultando em 1 mm para a linha de 12 V e 0,5 mm para 5 V e 3,3 V, enquanto sinais digitais e de sensores utilizam trilhas de 0,254 mm. Para o terra, foram adotados planos de referência em ambas as faces da placa.

A implementação final da placa pode ser visualizada por meio das vistas tridimensionais apresentadas nas Figuras 26 e 27¹, que evidenciam o posicionamento dos componentes, conectores e a organização geral do sistema.

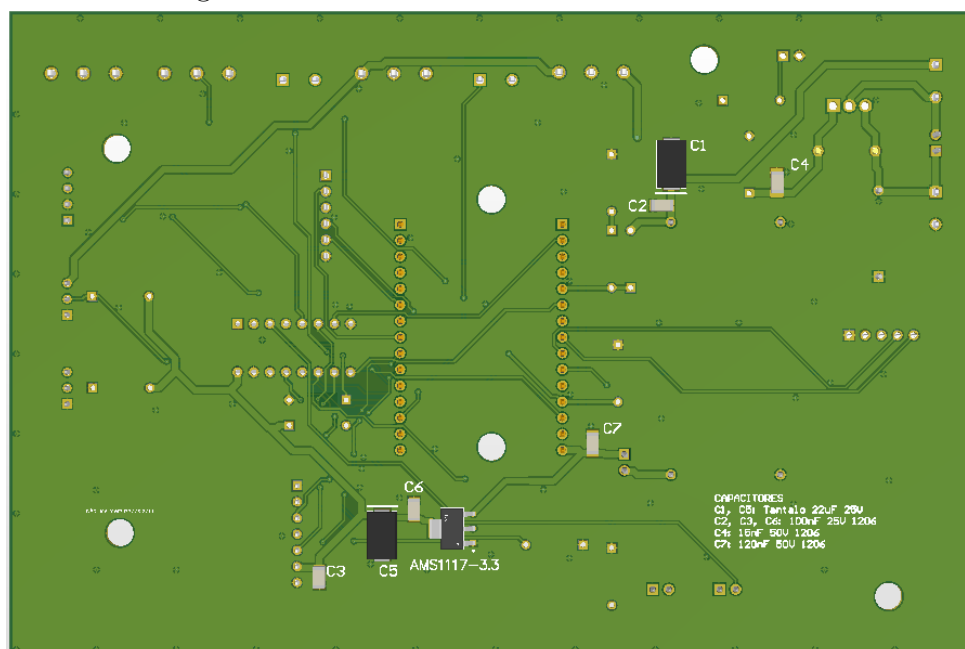
¹Parte dos circuitos implementados na placa não foi considerada nesta análise, por não integrar a versão final validada do sistema ou por não impactar os resultados experimentais apresentados.

Figura 26 – Vista superior 3D da PCI embarcada



Fonte: Autoria própria.

Figura 27 – Vista inferior 3D da PCI embarcada



Fonte: Autoria própria.

A placa desenvolvida possui dimensões de 152 mm x 100 mm, atendendo às restrições de espaço do veículo. Além disso, apresenta furos mecânicos para fixação e representações gráficas para facilitar a montagem.

Por fim, a placa foi encomendada por meio de serviço de prototipagem comercial, pelo custo de 1 dólar por unidade, fora taxas e fretes. O processo de montagem foi realizado manualmente pelo autor.

Com a implementação do hardware concluída, passa-se à etapa de desenvolvimento do firmware embarcado, responsável por controlar os periféricos, realizar a aquisição de dados e gerenciar a comunicação do sistema.

4.2 Desenvolvimento de Firmware Embarcado

O firmware desenvolvido é responsável pela aquisição, processamento, armazenamento e transmissão dos dados de telemetria do sistema. Para atender aos diferentes requisitos temporais dessas funções, faz-se necessária uma arquitetura baseada em gerenciamento de tarefas.

Nesse contexto, foi selecionado o FreeRTOS, recurso nativo do ESP32, para implementar um sistema de gerenciamento de tarefas, realizando o controle direto sobre a execução e temporização das funções. Essa abordagem viabiliza o uso de tarefas de diferentes períodos e prioridades.

O código foi estruturado e disponibilizado pelo autor na plataforma GitHub, através do repositório Baja UFSCar - Repositório do Projeto (MARMIROLI, 2026). Além disso, as subseções a seguir apresentam a organização das tarefas e os principais módulos implementados no sistema.

4.2.1 Arquitetura de Tarefas

O firmware foi projetado para ter uma arquitetura modular e eficiente, sendo estruturado a partir da definição das tarefas isoladas, seleção de núcleo de execução, priorização e integração. Essa medida permite o controle e manutenção de forma simplificada, além de permitir maior organização das tarefas em periodicidade, elevando a previsibilidade do sistema.

Como critério de distribuição de execuções entre os núcleos, tarefas de temporização previsíveis foram separadas daquelas com latência variável. As tarefas `lerSensores`, `contarHall` e `verificarErros` são executadas no núcleo 0, enquanto `transmitirLoRa` e `gravarSD` são executadas no núcleo 1.

Além da distribuição entre núcleos, foi necessário definir a prioridade para cada tarefa, de acordo com sua criticidade no sistema. Essa atribuição influencia diretamente o comportamento do escalonador preemptivo, garantindo que tarefas mais sensíveis a tempo tenham precedência de execução. A Tabela 9 apresenta essa organização.

Tabela 9 – Distribuição das tarefas FreeRTOS nos núcleos do ESP32

Núcleo	Tarefa	Função	Prioridade
0	<code>lerSensores</code>	Leitura de MPU6050 e Termopar	Maior
0	<code>contarHall</code>	Contagem de pulsos RPM e Velocidade	Maior
0	<code>verificarErros</code>	Monitoramento de falhas e reinicialização	Menor
1	<code>transmitirLoRa</code>	Composição e envio do pacote LoRa	Maior
1	<code>gravarSD</code>	Gravação de dados no cartão SD	Menor

Fonte: Autoria própria.

No núcleo 0, as tarefas `lerSensores` e `contarHall` receberam maior prioridade por dependerem de janelas de amostragem e eventos físicos, nos quais atrasos podem causar perda de informação. A tarefa `verificarErros` foi posicionada em nível intermediário, garantindo detecção de falhas sem interferir nas tarefas críticas. Já no núcleo 1, as tarefas `transmitirLoRa` e `gravarSD` possuem menor prioridade por operarem sobre dados já adquiridos e tolerarem maior latência.

Uma vez definida a organização das tarefas e suas prioridades, torna-se necessário integrar o sistema. Para isso, foi adotada uma estrutura global denominada de `SystemData`, que concentra os dados de temperatura, velocidade, RPM e aceleração, simplificando o acesso pelas múltiplas tarefas que usam dessas informações.

Entretanto, o uso de uma estrutura compartilhada introduz o risco de acessos concorrentes, podendo resultar em inconsistências nos dados. Para evitar esse problema, foi adotado um mecanismo de exclusão mútua, implementado por meio de `portMUX_TYPE` do FreeRTOS. Nesse mecanismo, quando uma tarefa tenta acessar um recurso já em uso, ela aguarda até que o acesso seja liberado.

Essa abordagem é adequada porque as operações envolvem apenas leitura e escrita de poucas variáveis, sendo executadas rapidamente. Assim, o tempo de espera fica limitado à duração dessas operações, não impactando de forma significativa a execução das demais tarefas e garantindo acesso seguro aos dados compartilhados.

Por fim, para a tarefa de `verificarErros`, foram utilizados *Event Groups*, um mecanismo do FreeRTOS baseado em um conjunto de *flags* compartilhadas entre tarefas. Na prática, quando um módulo entra em estado de erro, a tarefa supervisora é notificada e exibe a informação no monitor serial. Dessa forma, possibilita-se a detecção de falhas de maneira mais rápida e eficiente.

Com a estrutura de tarefas, prioridades e mecanismos de sincronização definidos, apresentam-se a seguir os aspectos e decisões relacionados à aquisição de sensores.

4.2.2 Aquisição de Sensores

A aquisição de dados foi estruturada em duas tarefas independentes no sistema: uma responsável pela leitura do sensor inercial e do termopar, e outra dedicada ao processamento de pulsos dos sensores Hall.

Os processamentos dos dados dos sensores MPU6050 e Termopar são executados na denominada `taskSensores`, descrita pelo Algoritmo 1. A junção é realizada para que os dados inferidos pelos sensores sejam atualizados nas variáveis globais ao mesmo momento, reduzindo rotinas de escritas concorrentes ao longo do código.

Porém, como os sensores de efeito Hall utilizam interrupções e proteções específicas (`portENTER_CRITICAL` e `portEXIT_CRITICAL`), optou-se por separar sua aquisição em uma tarefa distinta, descrita pelo Algoritmo 2.

Algoritmo 1: Fluxo geral da tarefa de aquisição de sensores

```

Inicializar marcadores de tempo da tarefa; Inicializar parâmetros de
amostragem;
Enquanto tarefa ativa faça
    Obter tempo atual;
    Se tempo atual - tempo da última leitura  $\geq 300\text{ ms}$  então
        Ler temperatura do termopar;
        Se a leitura for inválida então
            Sinalizar erro (Event Group);
        Senão
            Publicar temperatura na variável global (Mutex);
    Se (tempo atual - tempo da última leitura  $\geq$  período de amostragem) então
        Ler aceleração instantânea;
        Se a leitura for inválida então
            Sinalizar erro (Event Group);
        Senão
            Acumular leitura na janela estatística;
            Se amostras de aceleração  $== 5$  então
                Calcular máximo;
                Publicar máximo na variável global (Mutex);
    Tempo da última leitura = tempo atual;
    Aguardar 50 ms;

```

Embora os sensores MPU6050 e termopar compartilhem a tarefa, cada módulo possui um controle temporal independente. Dessa forma, a tarefa é executada periodicamente a cada 50 ms, enquanto a realização de cada leitura é condicionada a definições internas de

tempo, permitindo que sensores com requisitos temporais distintos compartilhem a mesma tarefa.

Para o acelerômetro, a aquisição é realizada a cada executar da tarefa, totalizando uma aquisição a cada 50 ms. Todavia, para evitar a perda de dados úteis, criou-se uma estrutura de pacotes, em que a cada cinco aquisições (250 ms), forma-se um pacote. A cada pacote formado, o valor máximo obtido dentro dele é registrado na variável global. Optou-se pela abordagem de pacotes para não sobrecarregar à escrita, e o uso de máximo para garantir a captura de picos de aceleração, que são justamente os dados mais rápidos e que devem ser preservados para a análise posterior.

O termopar, embora tratado na mesma tarefa, é amostrado em intervalos maiores, compatíveis com o tempo de conversão do módulo MAX6675 (MAXIN INTEGRATED, 2021). Para isso, a leitura é condicionada a um controle de tempo interno, resultando em uma taxa efetiva de 300 ms.

Algoritmo 2: Fluxo geral da tarefa de cálculo de RPM e Velocidade

Se *interrupção de pulso RPM ou Velocidade* **então**

└ Incrementar contador global de pulsos;

Enquanto *tarefa ativa* **faça**

└ Acesso protegido na leitura das interrupções (*portENTER_CRITICAL*);

└ Copiar contadores de pulsos das variáveis globais em variáveis locais;

└ Zerar contadores de pulsos das variáveis globais;

└ Sair da seção crítica (*portEXIT_CRITICAL*);

└ Obter tempo atual;

Se *houve pulsos no intervalo* **então**

└ $rpm = \frac{\text{Pulsos}}{\text{Intervalo} \cdot \text{Quantidade de imãs}} \cdot 60$;

Senão

└ $rpm = 0$;

Se *houve pulsos no intervalo* **então**

└ $velocidade = \frac{\text{Pulsos} \cdot (2 \cdot \pi \cdot R)}{\text{Intervalo}} \cdot 3.6$;

Senão

└ $velocidade = 0$;

└ Atualizar variáveis globais de RPM e velocidade (*Mutex*);

└ Aguardar 250 ms;

Na segunda tarefa, os valores de RPM e velocidade são obtidos a partir da contagem de pulsos gerados pelos sensores. Cada pulso recebido incrementa um contador global por meio de rotinas de interrupção, permitindo registrar eventos independentemente da execução da tarefa.

A tarefa periódica, executada a cada 250 ms, realiza a leitura desses contadores de forma protegida, desabilitando novas interrupções, e calcula a quantidade de pulsos ocorridos desde a última execução. A partir dessa variação, são estimados os valores de RPM e velocidade, considerando a duração fixa da janela de tempo. Para a velocidade, é aplicado o fator de conversão previamente definido na Equação 4.2.

A separação entre a contagem de pulsos e seu processamento permite registrar eventos de forma contínua, enquanto o cálculo é feito em intervalos definidos, evitando a perda de pulsos e mantendo a consistência das medições.

4.2.3 Monitoramento de Saúde do Sistema

Uma das grandes dificuldades do BAJA, e da eletrônica embarcada em geral, é identificar falhas de forma rápida, poupando tempo de manutenção durante as provas. Pensando nisso, foi implementado um mecanismo de monitoramento de saúde para acompanhar o funcionamento dos principais subsistemas, como comunicação LoRa, armazenamento em cartão SD e sensores. O Algoritmo 3 apresenta o fluxo geral dessa tarefa.

Algoritmo 3: Fluxo geral do monitoramento de saúde do sistema

```

Definir conjunto de erros monitorados (sensores, LoRa e SD);
Inicializar estado anterior dos erros;
Enquanto tarefa ativa faça
    Checar evento de erro utilizando EventGroup a cada 500 ms;
    Se erro de sensores foi reportado então
        Registrar ocorrência de falha de sensores;
        Limpar bit de erro de sensores após tratamento;
    Se erro de LoRa foi reportado então
        Registrar falha de comunicação LoRa;
    Senão se erro de LoRa foi recuperado então
        Registrar recuperação da comunicação LoRa;
    Se erro de SD foi reportado então
        Registrar falha de operação do cartão SD;
    Senão se erro de SD foi recuperado então
        Registrar recuperação da operação do cartão SD;
    Aguardar 50 ms;

```

A cobertura do monitoramento é feita por subsistema e inclui o estado de inicialização e resultado da última operação, permitindo assim distinguir falhas pontuais de condições persistentes. A partir desses dados, uma tarefa supervisora avalia periodicamente

mente o estado global do sistema, reportando o módulo e o momento da falha na porta serial, no LoRa e no cartão SD.

Para os sensores, o registro de falha é feita de forma pontual, com marcação de erro e liberação do sinal. Já para o LoRa e o cartão SD, o monitoramento é por estado. Essa implementação permite que seja analisável o período exato em que os sistemas de comunicação e gravação ficaram em falha, revelando quantos pacotes de dados foram perdidos. Para os sensores, essa informação não é relevante, visto que os erros tendem a ser não persistentes e a perda de algumas leituras não impactam significativamente a análise posterior.

4.2.4 Armazenamento em Cartão SD

O bloco de gravação é responsável por armazenar dados provenientes dos sensores para posterior análise e serve como redundância aos dados obtidos por telemetria. Sendo assim, é uma tarefa definida com menor prioridade e, além disso, possui estruturas que reduzem o tempo de uso do núcleo, que são naturalmente extensos devido a operações de escrita, como apresentado no Algoritmo 4.

Algoritmo 4: Tarefa de Gravação em Cartão SD (*taskSDLogging*)

Declarar períodos e tamanho de lote;

Enquanto *tarefa ativa* **faça**

Se *SD não inicializou* **ou** *arquivo texto não reabre após 2s* **então**

 Reportar erro de SD;

 Aguardar manutenção;

Para *cada módulo (LoRa, Hall, MPU e Termopar)* **faça**

 Verificar estado de erro;

Se *houver transição de estado na variável de erro* **então**

 Gravar evento de falha ou recuperação no cartão SD;

 Acesso protegido nas variáveis globais (mutex);

 Copiar dados para variáveis locais;

 Sair do mutex;

 Armazenar linha (*tempo, temperatura, rpm, velocidade, a_x, a_y, a_z*);
no pacote;

Se *pacote ≥ 5* **ou** *tempo da tarefa ≥ 1 s* **então**

Se *gravação do lote bem-sucedida* **então**

 Executar *flush* e marcar SD como saudável;

Senão

 Fechar arquivo, descartar lote e reportar erro;

 Aguardar 200 ms;

No código, a cada 200ms, o algoritmo rastreia se as condições para operação estão satisfeitas (acesso ao arquivo texto) e verifica o funcionamento dos demais módulos, registrando falhas e recuperações. Em seguida, é feita a leitura protegida dos dados globais utilizando mutex e cópia dos valores para variáveis locais, guardando-os em um pacote. Por fim, quando o pacote atinge 5 registros ou o tempo da tarefa chega a 1 segundo, os dados são gravados em lote no cartão SD. Isso é feito para amenizar o uso de processamento necessário para a gravação, que envolve o sistema de arquivos e o driver do cartão, ocupando o núcleo por menos tempo.

4.2.5 Comunicação LoRa

A comunicação LoRa necessita de dois códigos diferentes: um para configuração e outro para execução em si, como instruído pela biblioteca E220 utilizada. Nesse viés, o pri-

meiro passo é executar a configuração com os parâmetros desejados, conforme apresentado na Tabela 10.

Tabela 10 – Parâmetros de configuração do módulo embarcado

Parâmetro	Valor adotado
Endereço do transmissor	0x02
Canal de operação	23
Baudrate UART	9600 bps
Taxa de transmissão	4,8 kbps
Tamanho do pacote	64 bytes
Potência de transmissão	22 dBm
Transmissão fixa	Habilitada
Habilitar RSSI	Habilitado

Fonte: Autoria própria.

Os valores foram selecionados visando a melhor qualidade de sinal no contexto da competição, com interferências e distância de 500 m. Nesse viés, o endereço e o canal são utilizados para estabelecer identidade e situar os módulos na comunicação, e são definidos pelos primeiros 3 bytes da mensagem quando o módulo está em modo Transmissão Fixa, filtrando o sinal e removendo interferências provenientes de dispositivos de outras equipes.

Para parâmetros de desempenho, a taxa de transmissão, tamanho do pacote e potência de transmissão delimitam todo o escopo. A primeira, taxa de transmissão, define a velocidade que o módulo transmite os dados, sendo inversamente proporcional ao atraso de envio dos pacotes e ao alcance máximo. O segundo, tamanho do pacote, é escolhido o mais próximo do tamanho utilizado, no caso, de 64 bytes, para evitar fragmentação no envio. E, por último, a potência é correlacionada à intensidade do sinal enviado, sendo selecionada a maior possível, ao custo de consumo.

Ademais, parâmetros como RSSI foram habilitados para medir intensidade do sinal e o baudrate da interface UART utiliza valor padrão, sendo apresentada apenas pela importância de ser semelhante ao do receptor, para evitar congestionamento de dados.

Após a etapa inicial de configuração, na execução principal, a transmissão de dados é realizada por uma tarefa periódica com intervalo de 250 ms, responsável por enviar ao receptor um conjunto reduzido de variáveis do sistema. A cada ciclo, é obtida uma cópia consistente dos dados globais, utilizada para compor a mensagem a ser transmitida, evitando inconsistências causadas por atualizações concorrentes. O fluxo em alto nível dessa tarefa é apresentado no Algoritmo 5.

Algoritmo 5: Fluxo geral da tarefa de transmissão LoRa (*taskLoRa*)

```

Inicializar contador de sequencia de pacotes;
Enquanto tarefa ativa faça
    Se módulo LoRa indisponível então
        Sinalizar falha de comunicação;
        Aguardar recuperação;
    Copiar variáveis globais para variáveis locais utilizando mutex;
    Montar pacote telemetria (seq, t, temperatura, rpm, velocidade);
    Enviar pacote de telemetria na serial para acompanhamento;
    Enviar pacote via LoRa ao endereço/canal configurados;
    Se envio bem-sucedido então
        Atualizar estado de saúde do LoRa para normal;
        Limpar sinalização de erro de LoRa;
    Senão
        Marcar módulo LoRa como indisponível;
        Atualizar estado de falha e sinalização de erro;
    Se modo de depuração habilitado então
        Exibir detalhes adicionais da transmissão;
    Incrementar contador sequencial;
    Se tempo de transmissão < 250 ms então
        Aguardar tempo restante do período;

```

A cada ciclo de transmissão, a tarefa obtém uma cópia atômica dos dados compartilhados utilizando o mecanismo mutex, garantindo uma leitura consistente. Com essa leitura em mãos, é formado um pacote contendo sequência e tempo percorrido (calculados internamente para verificação de erros e temporização), e temperatura, RPM e velocidade, da variável global. Com isso, o pacote é transmitido ao receptor via chamada `sendFixedMessage`, que utiliza os parâmetros de endereçamento e canal configurados na Tabela 10.

A transmissão é realizada de forma não confirmada, sem mecanismo de paridade ativo. Porém, o método de verificação de funcionamento segue o último envio concluído, contestando que o módulo funciona.

Após a definição dos módulos de aquisição, processamento e comunicação, bem como dos recursos de apoio, a seção seguinte apresenta o desenvolvimento do bloco receptor, responsável pela recepção e visualização dos dados transmitidos.

4.3 Desenvolvimento do Receptor e Suporte de Logs

De maneira análoga ao bloco embarcado, o bloco receptor requer que o módulo LoRa seja configurado antes de atuar. Para isso, a configuração contida na Tabela 11 é aplicada.

Tabela 11 – Parâmetros de configuração do módulo receptor

Parâmetro	Valor adotado
Endereço do receptor	0x03
Canal de operação	23
Baudrate UART	9600 bps
Taxa de transmissão	4,8 kbps
Tamanho do pacote	64 bytes
Potência de transmissão	22 dBm
Transmissão fixa	Habilitada
Habilitar RSSI	Habilitado

Fonte: Autoria própria.

Os valores selecionados são casados com os utilizados no transmissor, por recomendação do fabricante, mudando apenas o endereço do módulo.

Para a rotina principal, o bloco receptor possui o firmware baseado em superlaço, escolhido pela simplicidade da funcionalidade desejada, sendo essa a responsabilidade pela recepção das mensagens transmitidas pelo sistema embarcado e disponibilização dessas informações em interface serial. Seu funcionamento é baseado na leitura contínua do módulo LoRa e no registro das mensagens recebidas em formato textual. O Algoritmo 6 apresenta a lógica do laço desenvolvido.

A cada pacote válido, são extraídas informações como tamanho da mensagem, RSSI e conteúdo da mensagem, que são então enviados pela interface serial. Esse formato permite o registro direto dos dados em ferramentas de monitoramento ou arquivos de log, facilitando a análise posterior do desempenho do enlace.

Além disso, são calculadas métricas simples de comunicação, como número de pacotes e volume de dados recebidos por segundo. Essas informações permitem avaliar, de forma direta, a taxa efetiva de transmissão e possíveis perdas de pacotes ao longo dos testes.

Algoritmo 6: Execução em laço do bloco receptor

```

Inicializar porta serial;
Inicializar módulo LoRa;
Se módulo LoRa não responde então
  | Reportar falha;
Senão
  | Ler configuração atual do módulo LoRa;
  | Registrar tempo inicial do sistema;
  | Reportar receptor pronto;
Enquanto laço ativo faça
  | Obter tempo atual;
  | Se LoRa disponível então
  | | Receber pacote;
  | | Se recepção bem-sucedida então
  | | | Incrementar contadores de pacotes e bytes recebidos;
  | | | Extrair campos do pacote: sequência, tempo, temperatura, RPM e
  | | | velocidade;
  | | | Registrar RSSI do pacote recebido;
  | | | Exibir campos extraídos e RSSI na porta serial;
  | | Senão
  | | | Mostrar na serial que houve erro na comunicação;
  |

```

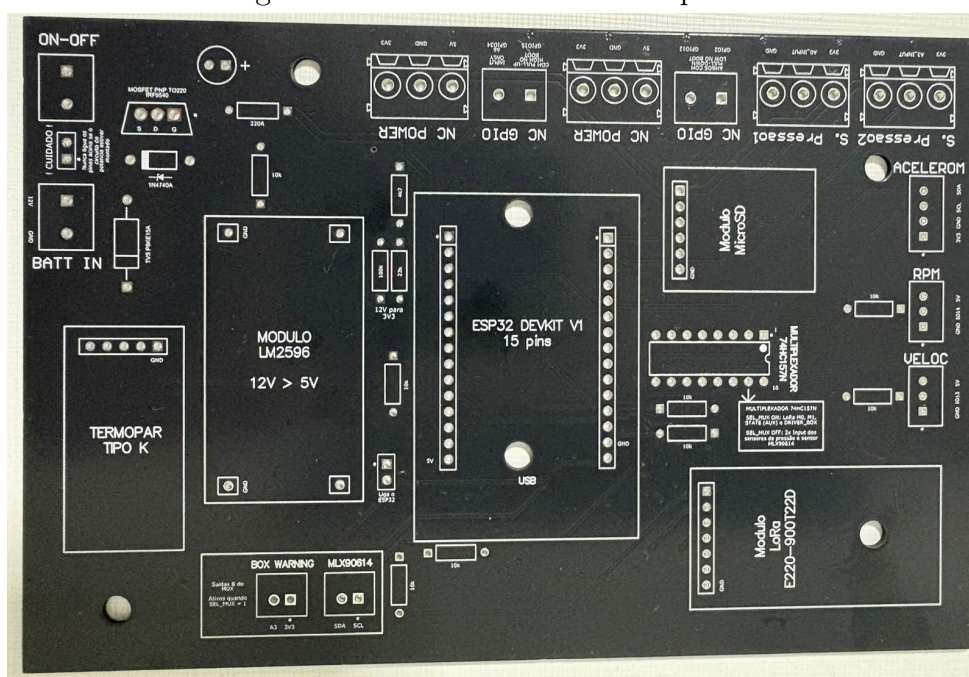
5 Resultados Obtidos

Este capítulo apresenta os resultados obtidos ao longo do desenvolvimento e da validação experimental do sistema. Inicialmente, apresenta-se o resultado da fabricação e montagem da placa de circuito impresso. Em seguida, são discutidos os resultados obtidos e os testes descritos na Seção 3.4, realizados de forma progressiva desde a verificação em bancada até a validação do sistema completo.

5.1 Placa de Circuito Impresso

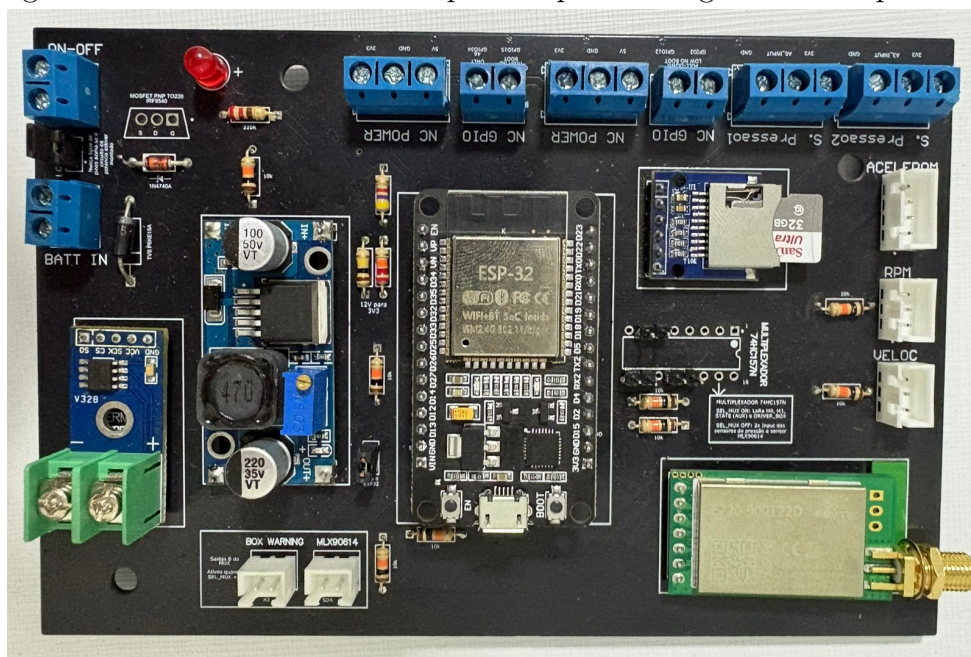
A placa de circuito impresso foi fabricada por serviço de prototipagem comercial e montada manualmente pelo autor. As Figuras 28 e 29 apresentam, respectivamente, a placa após a fabricação e após a montagem dos componentes.

Figura 28 – Placa de Circuito Impresso



Fonte: Autoria própria.

Figura 29 – Placa de Circuito Impresso após montagem dos componentes



Fonte: Autoria própria.

A placa resultante apresenta dimensões de 152 mm $\times$ 100 mm, com furos mecânicos para fixação e marcações gráficas que auxiliam na montagem e manutenção.

Após a montagem, o sistema passou por testes de continuidade e verificação de alimentação, confirmando a integridade das conexões. Além disso, a corrente de entrada foi medida em condições de operação, indicando um consumo médio de 90 mA, que está abaixo ao valor previsto na Equação 3.2, pela estimativa inicial considerar valores de pico.

Com o sistema fisicamente pronto, os ensaios experimentais foram conduzidos conforme descrito na Seção 3.4, buscando validar a operação integrada dos módulos, o desempenho do LoRa em condições controladas e representativas de uso.

5.2 Validação Experimental do Sistema

5.2.1 Verificação Funcional em Bancada

O primeiro ensaio teve como objetivo confirmar a operação integrada dos principais módulos do sistema em condição controlada. O transmissor e o receptor foram posicionados a 3 m de distância, sem obstáculos e em operação contínua por 100 s.

Durante os testes, foi necessário ajustar a configuração do enlace LoRa devido à ocorrência de atrasos periódicos na transmissão. Na configuração inicial, observava-se uma latência próxima de 3 s a cada quatro pacotes enviados. Esse comportamento

indicava acúmulo de mensagens na fila interna do módulo, causado pela combinação entre baixa taxa de transmissão aérea e pacotes relativamente grandes para o intervalo de envio adotado pelo sistema.

Na prática, novos pacotes eram gerados antes da conclusão completa da transmissão anterior, provocando o engavetamento das mensagens e o aumento progressivo da latência.

Para eliminar o problema, a taxa de transmissão aérea foi ajustada para 2,4 kbps e o tamanho do pacote reduzido para 32 bytes, junto com o conteúdo da mensagem. Após a alteração, o enlace passou a operar de forma estável, sem os atrasos observados anteriormente, uma vez que o tempo necessário para transmissão de cada pacote foi reduzido.

Com os ajustes realizados, o teste foi devidamente conduzido e a Tabela 12 apresenta os indicadores medidos.

Tabela 12 – Resultados do ensaio de Verificação em Bancada

Critério	Limite	Resultado	Situação
(Sistema) Contagem de reinicializações	0	0	Aprovado
(Sistema) Perda de pacotes (P_{perda})	$\leq 1\%$	0	Aprovado
(LoRa) Taxa de recepção (R_{rx})	$\approx 4pkt/s$	4,011 pkt/s	Aprovado
(LoRa) RSSI	N/A	-46 dBm	Referência
(SD) Linhas gravadas no microSD	$\approx 5pkt/s$	5 pkt/s	Aprovado

Fonte: Autoria própria.

Todos os critérios de aceitação foram atendidos, indicando que o sistema operou de forma integrada e estável durante o período de teste, após o ajuste do SF. Durante o teste, todos os sensores foram brevemente estimulados, possibilitando validar a aquisição, gravação e comunicação dos dados e estrutura de firmware.

Além disso, o valor referência para RSSI foi estabelecido em -46 dB, indicando o valor em condições máximas ideais.

Por fim, o uso do Indicador de Força do Sinal Recebido durante o teste serviu para validar a temporização do sistema escolhido. Na Tabela 13 é apresentado estatísticas do tempo de execução da tarefa gravarSD(), para o primeiro teste.

O sistema apresentou atraso de 1ms em 10 de 500 pacotes, portanto, conclui-se que o sistema possui elevada estabilidade temporal, até mesmo para tarefas consideradas lentas, como a escrita em cartão SD.

Tabela 13 – Estatísticas de tempo de execução da tarefa `gravarSD()`

Métrica	Valor
Média	250,02 (ms)
Desvio Padrão	0,14 (ms)
Mínimo	250,00 (ms)
Máximo	251,00 (ms)
Amostras	500

Fonte: Autoria própria.

5.2.2 Validação dos Sensores

O ensaio dos sensores avaliou as leituras de cada sensor frente a estímulos conhecidos e/ou instrumentos de referência. Os testes foram conduzidos individualmente em bancada, visando observar a resposta durante transientes e em regime permanente.

5.2.2.1 Medição de Temperatura

Para a validação do conjunto MAX6675 e termopar Tipo K, foram realizados dois testes com estímulos distintos: um teste de baixa temperatura, utilizando gelo e água como referência próxima a 0 °C, e um teste de alta temperatura, utilizando água fervente como referência de aproximadamente 100 °C.

Em ambos os testes, as leituras do sistema foram comparadas com as medidas obtidas por um multímetro digital MS-10 (MINIPA, 2020) equipado também com termopar Tipo K. O multímetro apresenta precisão de ± 3 °C, resolução de 0,1 °C e uma sistema de aquisição calibrado e independente da aplicação desenvolvida, tornando-se adequado para a comparação.

Os termopares utilizados, apesar de serem do mesmo tipo, apresentavam junções de medição de tamanhos diferentes, o que impacta diretamente no tempo de resposta térmica. Na Figura 30 está apresentada a junção de medição do termopar utilizado no sistema, enquanto na Figura 31 está apresentada a junção de medição do termopar utilizado com o multímetro.

Figura 30 – Termopar tipo K utilizado com o módulo MAX6675



Fonte: (ELETROGATE, s.d.)

Figura 31 – Termopar tipo K utilizado com o multímetro MS-10



Fonte: (ELETROGATE, s.d.)

É possível observar que a junção de medição do termopar utilizado com o módulo, que é significativamente maior do que a junção do termopar utilizado com o multímetro, impacta diretamente no tempo de resposta térmica.

Por fim, em todos os testes, os termopares foram devidamente posicionados próximos e imersos na mesma profundidade, garantindo que ambos os sensores estivessem sujeitos às mesmas condições térmicas. E, as leituras provenientes do ESP32 foram registradas no cartão SD, enquanto as leituras do multímetro foram gravadas em vídeo e, posteriormente, transcritas para comparação.

5.2.2.1.1 Baixa temperatura

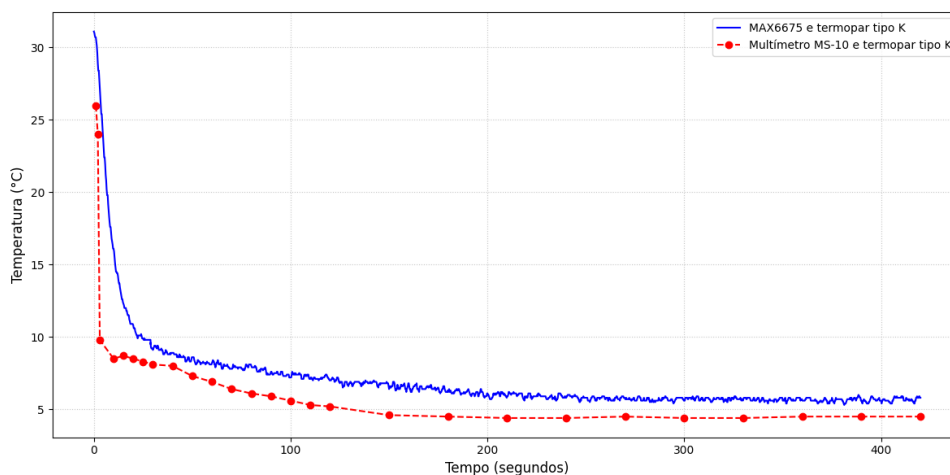
O teste em baixa temperatura foi conduzido como um degrau térmico: iniciando com os termopares fixados em um copo vazio, é despejado um segundo copo contendo gelo e água, de forma a criar um ambiente térmico próximo a 4 °C. As Figuras 32 e 33 apresentam, respectivamente, a configuração do ensaio e a comparação entre as leituras do sistema e do multímetro.

Figura 32 – Configuração utilizada para o ensaio



Fonte: Autoria própria.

Figura 33 – Comparação entre instrumentos em baixa temperatura.



Fonte: Autoria própria.

Percebe-se que o sistema apresentou um comportamento coerente com o estímulo térmico, com erro de aproximadamente 2 °C em relação ao multímetro após estabilização. Porém, respondendo de forma relativamente lenta, com atraso de 19 s para atingir 10 °C e 100 s para atingir a temperatura de regime.

O erro observado é esperado e aceitável, considerando as tolerâncias (+/- 2 °C) estabelecidas pelo fabricante do MAX6675 (MAXIN INTEGRATED, 2021). Já o atraso é justificado pelo diferente tamanho entre as junções de medição dos termopares do módulo e do multímetro.

Finalizado o teste em baixa temperatura, o sistema foi submetido ao teste de alta temperatura.

5.2.2.1.2 Alta temperatura

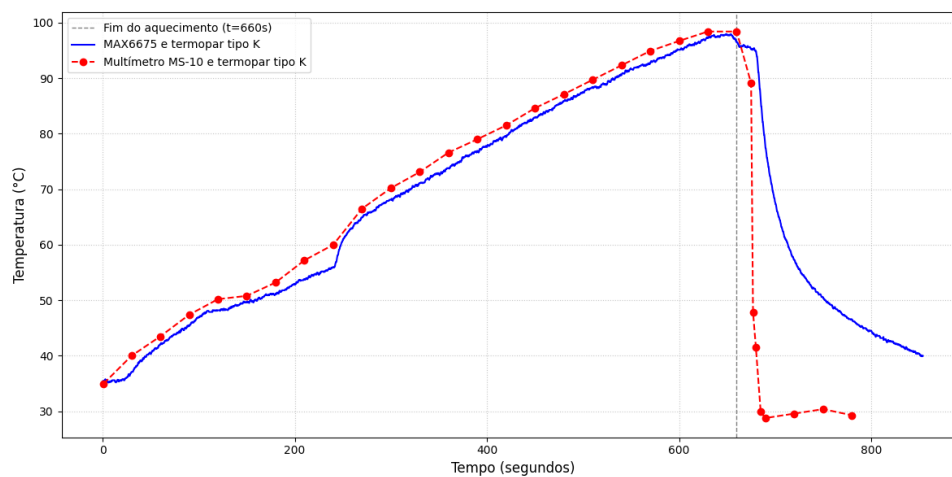
Nesse ensaio, foi reproduzido uma rampa térmica: iniciando com os termopares fixados em uma caneca de metal, cobertos com água em temperatura ambiente, em seguida, liga-se o fogão e aguarda-se até que a água atinja o ponto de ebulição, criando um ambiente térmico próximo a 100 °C. Ao chegar em ebulição, o fogão foi desligado. As Figuras 34 e 35 apresentam, respectivamente, a configuração do ensaio e a comparação entre as leituras do sistema e do multímetro.

Figura 34 – Configuração do ensaio em alta temperatura.



Fonte: Autoria própria.

Figura 35 – Comparação entre instrumentos em alta temperatura.



Fonte: Autoria própria.

A temperatura de ebulição, dada a pressão de 1,016 atm, é de aproximadamente 99,6 °C. O pico, observado no multímetro, foi de 98,4 °C e no sistema, de 97,8 °C. Com isso, conclui-se que o sistema apresentou um comportamento coerente com o estímulo térmico, com erro de aproximadamente 1,8 °C em relação à referência e 0,6 °C em relação ao multímetro. Além disso, apresentou resposta temporal perceptivelmente atrasada à rampa com atraso de 20 s para atingir 50 °C e 25 s para atingir 97 °C, em relação ao multímetro. De maneira análoga ao teste em baixa temperatura, o erro observado é esperado, dentro dos padrões estabelecidos pelo fabricante.

Considerando que o comportamento esperado para a caixa CVT, local onde o sensor será instalado, sofrerá variações térmicas em forma de rampa positiva, conclui-se que o sensor é ideal ao projeto, apresentando resposta adequada e precisão dentro dos limites toleráveis, além de possuir encapsulamento mais resistente contra desgastes mecânicos quando comparado a outras opções disponíveis na mesma faixa de custo.

5.2.2.2 Sensores Hall A3144

O objetivo do ensaio é validar a leitura dos sensores de efeito Hall A3144 (ALLEGRO, s.d.), utilizados para medir aquisições de RPM e velocidade. Para isso, foi utilizado um automóvel comercial leve, de forma a gerar estímulos reais e fornecer saída comparável.

Para aquisição de dados da referência, foi utilizado um cabo adaptador OBD2 para DB9, juntamente com um KVASER USB (KVASER, s.d.).

As leituras realizadas por CAN foram processadas e armazenadas em um arquivo de texto através de um programa desenvolvido em Python, enquanto as leituras do sistema foram registradas no cartão SD.

5.2.2.2.1 Rotações por minuto

Para medir o RPM, é necessário realizar a leitura através de medição indireta, visto que é inviável a aquisição diretamente do eixo do motor. Portanto, o sistema realiza aquisições na polia do virabrequim, onde é possível obter um sinal de frequência na razão de 1:1 com a rotação do motor.

Para gerar variações no campo magnético detectado pelo sensor, foram fixados três ímãs na polia do virabrequim, de forma a gerar três pulsos por rotação. A Figura 36 apresenta a instalação do sensor e dos ímãs na polia.

Figura 36 – Instalação do sensor e ímãs na polia do virabrequim



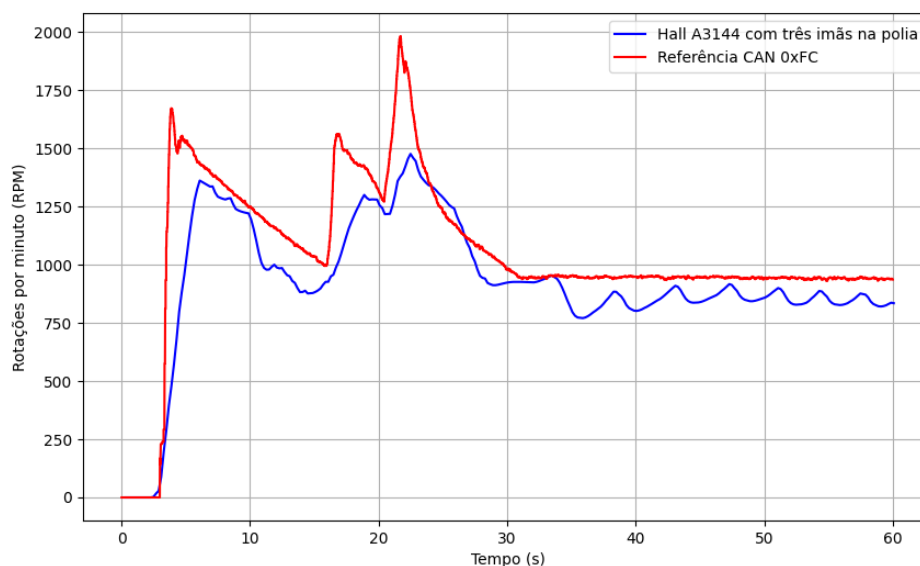
Fonte: Autoria própria.

Na etapa, é necessário garantir que os ímãs estejam equidistantes entre si, para que o intervalo entre os pulsos permaneça constante, evitando leituras artificialmente adiantadas ou atrasadas.

Além disso, a quantidade de ímãs implica diretamente na resolução da medição, visto que a frequência do sinal gerado para uma mesma rotação é proporcional à quantidade de ímãs utilizados. Por fim, o sensor deve ser posicionado de forma a garantir a detecção adequada dos pulsos, sendo assim, foi posicionado a aproximadamente 5 mm de distância da polia, garantindo a detecção dos pulsos sem risco de contato mecânico.

Após a configuração do sistema, com o carro imóvel, a placa de aquisição foi iniciada e o motor do carro foi ligado, e a Figura 37 apresenta a comparação entre os resultados obtidos.

Figura 37 – Comparação entre leituras de RPM do sistema e via CAN



Fonte: Autoria própria

O sistema apresentou uma resposta coerente à referência, porém com atraso à estímulos rápidos e oscilação em regime. A principal causa das oscilações são os posicionamentos dos ímãs que, com pequenas variações angulares, geram variação na interpretação dos pulsos, resultando em leituras inconsistentes. Em uma instalação definitiva, elementos magnéticos são fixados pelos próprios fabricantes, garantindo o posicionamento correto e a estabilidade das leituras.

Para o atraso observado, a causa está relacionada tanto à estratégia de aquisição: o cálculo da rotação é realizado a partir da contagem de pulsos em intervalos discretos, o que reduz a sensibilidade a variações rápidas. Como consequência, os picos rápidos são suavizados, fazendo com que a resposta medida apresente menor amplitude em comparação à referência.

Em conclusão, o sistema é capaz de medir o RPM de forma aceitável dentro dos limites do projeto proposto, dada a coerência em relação à referência e ao limite de variação proposto.

5.2.2.2.2 Velocidade

Para medir a velocidade, foi utilizado um sensor Hall A3144 com um ímã fixado no semieixo, permitindo a detecção de pulsos proporcionais à rotação da roda. A Figura 38 apresenta a instalação do sensor e do ímã no semieixo.

Figura 38 – Instalação do sensor Hall A3144 e ímã no semieixo

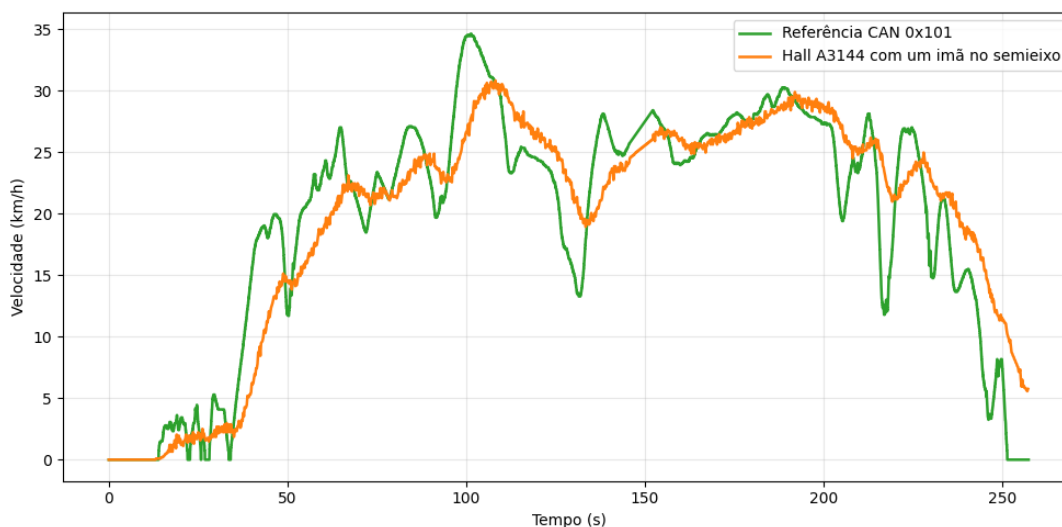


Fonte: Autoria própria

No modelo do carro utilizado, o acesso ao semieixo é simples, facilitando o processo de instalação do ímã e do sensor. Por ser um eixo com raio pequeno, o intervalo entre os pulsos detectados pelo sensor será menor, sendo assim não é necessário utilizar múltiplos ímãs. Além disso, o sensor foi posicionado a aproximadamente 1 cm do ímã, utilizando um suporte de metal para garantir estabilidade. Na prática, o conjunto do motor somado com a suspensão do veículo, gera vibrações que podem afetar a medição, porém não suficientemente para impossibilitar a aferição da velocidade.

Em seguida, após configurar o sistema e rodar com o carro por alguns minutos, foram adquiridos os dados de velocidade estimados, podendo ser comparados à referência via CAN, como apresentado pela Figura 39.

Figura 39 – Comparação entre referência e dado obtido para velocidade



Fonte: Autoria própria

Os dados aferidos apresentam resposta atrasada em 5 a 10 s em relação à referência, devido à estratégia de medir os pulsos por intervalos. Porém, o comportamento geral é coerente com a referência e funciona para sistemas como o proposto, onde o requisito é monitoramento e estudo de desempenho. Além disso, o sistema apresenta ruído de medição, devido à variação na distância entre ímã e sensor causada pela suspensão, ao movimento do carro.

Nesse viés, o uso dos sensores A3144, nas configurações para RPM e para velocidade, mostram-se aceitáveis para o projeto. Todavia, a construção de suportes robustos, que garantam estabilidade, bem como a boa fixação do ímã na polia e no semieixo, trariam ganhos significativos em relação à ruídos e oscilações. Aliado a isso, a estratégia de aquisição, que atualmente é baseada em contagem de pulsos por intervalos, pode ser substituída por contagem por eventos, trazendo respostas mais rápidas, porém a custo de processamento.

5.2.2.3 Aceleração

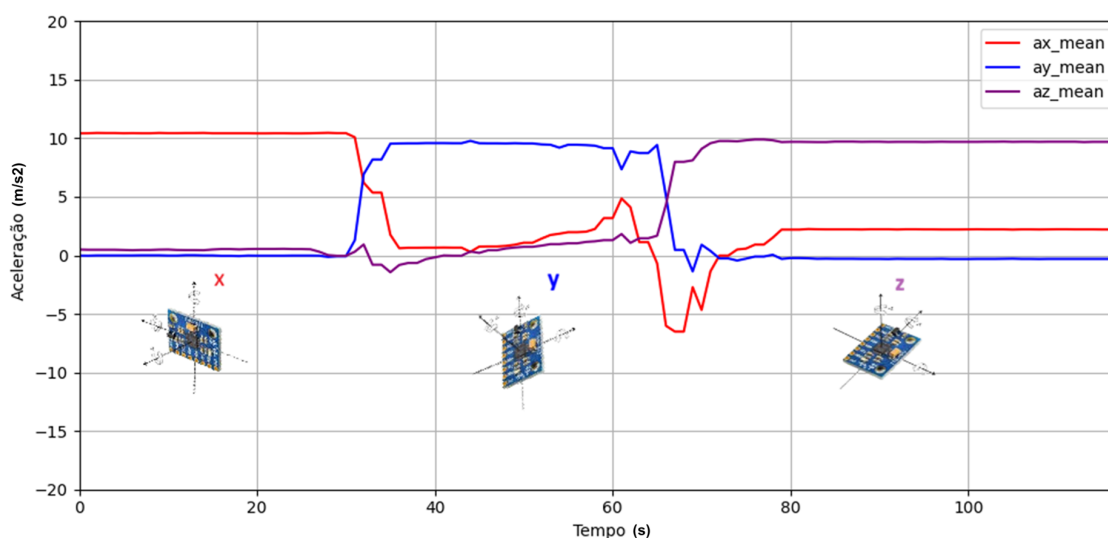
Com o objetivo de verificar o funcionamento do sensor, foi realizado um ensaio baseado na variação de orientação em relação ao vetor gravitacional. A ideia central é explorar o fato de que, em repouso, a única aceleração medida deve ser a da gravidade, projetada nos eixos do sensor conforme sua orientação, sem a necessidade de um instrumento ou valores de referência.

O procedimento consistiu em posicionar o módulo sobre uma superfície plana e, em seguida, rotacioná-lo manualmente de forma que cada eixo, individualmente, ficasse

alinhado com a direção da gravidade. Dessa forma, espera-se que o eixo alinhado apresente leitura próxima de 1 g , enquanto os demais permaneçam próximos de zero.

A Figura 40 apresenta os dados registrados durante o ensaio. Para facilitar a interpretação, são indicados a orientação do sensor durante o intervalo de leitura, permitindo associar diretamente cada trecho do sinal à orientação correspondente.

Figura 40 – Resposta dos eixos de aceleração durante o teste de orientação.



Fonte: Autoria própria.

O comportamento observado segue o esperado: em cada intervalo estático, um dos eixos apresenta valor próximo de 1 g , enquanto os outros dois permanecem próximos de 0 g .

Durante as transições entre orientações, os três eixos passam a apresentar variações simultâneas. Isso ocorre porque, nesse momento, o sensor está sujeito não apenas à gravidade, mas também às acelerações associadas ao movimento de rotação imposto manualmente.

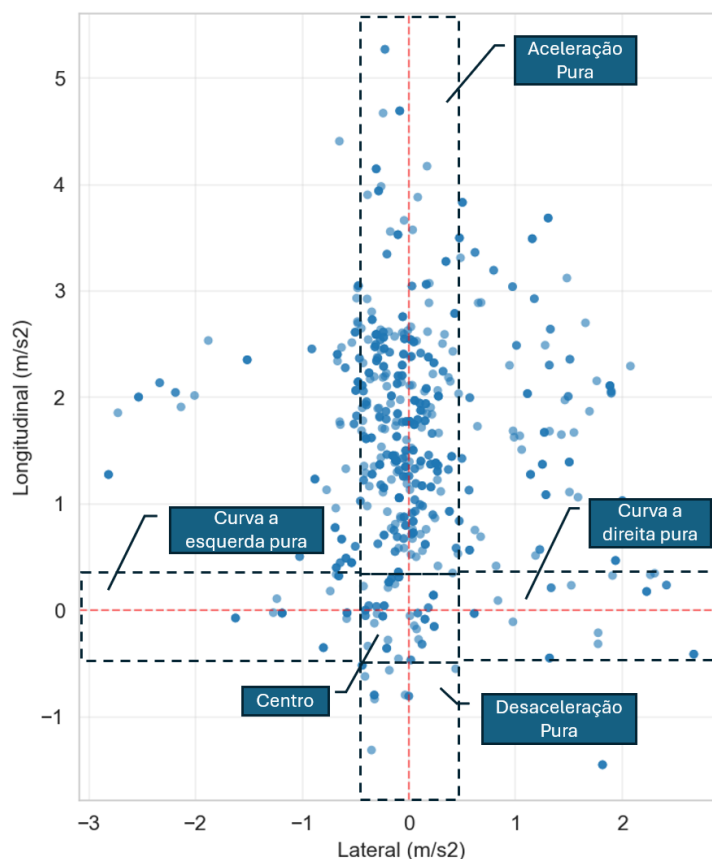
Esse resultado reforça que o sensor está respondendo de forma coerente do ponto de vista físico, indicando que as medições são consistentes com a projeção do vetor gravitacional nos diferentes eixos (NXP SEMICONDUCTORS, 2020).

Com isso, conclui-se que o sensor de aceleração está operando adequadamente, apresentando leituras coerentes com as orientações impostas e com o comportamento esperado.

No próximo teste, o acelerômetro foi testado em conjunto com a comunicação LoRa. Para isso, o módulo foi posicionado dentro do carro, enquanto o receptor recebia informações de aceleração. Para o teste, o sensor foi posicionado próximo ao centro de

massa, próximo ao câmbio, com seus eixos orientados de forma que o eixo X representasse a aceleração longitudinal e o eixo Y a aceleração lateral. Após locomover o carro por alguns minutos, é possível gerar o Diagrama GG, apresentado na Figura 41, que é um gráfico de dispersão que relaciona as acelerações laterais e longitudinais.

Figura 41 – Diagrama GG obtido a partir dos dados de aceleração



Fonte: Autoria própria.

No gráfico, foram adicionados alguns limitadores para facilitar a interpretação, que é: no teste, o carro possui um caráter de aceleração positiva, com curvas leves a direita e curvas mais acentuadas a esquerda. Isso se deve ao percurso escolhido, que é uma estrada rural em reta e com pequenos desvios.

Esse tipo de análise ajuda a equipe a compreender o comportamento do carro durante a prova, identificado qual o maior ponto de estresse e serve como base para corrigir problemas de alinhamento e suspensão.

5.2.3 Avaliação do Enlace LoRa em Campo

O ensaio dois caracterizou o desempenho do enlace LoRa em função da distância, avaliando o RSSI e a taxa de perda de pacotes em seis pontos: 20, 50, 100, 200, 500 e

1000 m. A Figura 42 apresenta o local onde o ensaio foi conduzido, mantendo o receptor fixo enquanto o transmissor se deslocava em linha reta. Em cada ponto de medição, o conjunto permanecia parado por dois minutos para estabilização e coleta dos dados, seguindo posteriormente para o próximo ponto.

Figura 42 – Vista satélite do local do ensaio de avaliação do enlace LoRa



Fonte: Autoria própria.

O local selecionado para o ensaio consistia em um campo aberto e de visada direta, buscando reproduzir condições semelhantes às encontradas em competições BAJA. Além disso, a via utilizada apresentava leve inclinação, porém sem interferências próximas capazes de causar bloqueio relevante do sinal. Por fim, a antena do transmissor foi posicionada a 1,5 m do solo, enquanto a do receptor a 2 m, simulando a instalação proposta para o sistema.

Durante o período de realização dos testes, as condições climáticas permaneceram estáveis, com céu parcialmente nublado, temperatura média de 25 °C e umidade relativa do ar de 44% (INMET, 2025).

Os resultados foram organizados na Tabela 14, onde é apresentada a relação entre pacotes recebidos, taxa de perda de dados e RSSI do sinal, para cada distância testada.

Tabela 14 – Resultados do LoRa em função da distância

Distância (m)	Pacotes recebidos (pkt/s)	Pacotes perdidos (%)	RSSI (dBm)
Referência	4	0	-46
20	3,98	0,71	-48
50	3,95	0,73	-55
100	3,94	0,80	-67
200	3,91	1,27	-75
500	3,81	4,91	-85
1000	3,52	11,68	-92

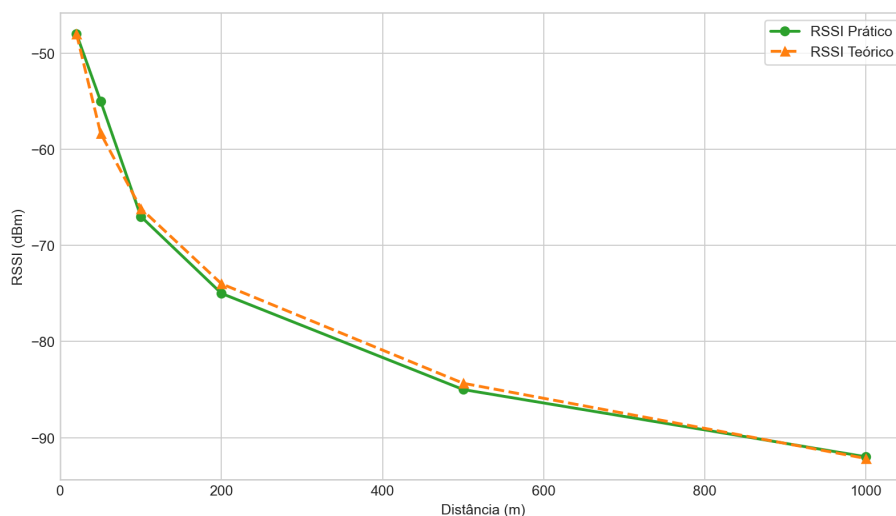
Fonte: Autoria própria.

É importante destacar que "Pacotes recebidos" são considerados qualquer informação recebida pelo receptor durante o período de amostragem, enquanto "Pacotes perdidos" representam a porcentagem de pacotes que não foram recebidos ou não são válidos.

Ademais, os resultados indicam a degradação do sinal em relação à distância, causando perda de pacotes significativa a 1000 m. Como tentativa de melhorar o desempenho nesse quesito, poderiam ser utilizadas configurações com maior SF, porém isso implicaria em redução da taxa de transmissão aérea, o que não satisfaz o projeto. Adicionalmente, estratégias como o uso de antenas de maior ganho e redução do tamanho do pacote enviado, satisfariam maiores distâncias. Contudo, dentro do projeto, os resultados observados são aceitáveis, visto que o requisito de 500 m é atendido, apresentando taxa de perda de pacotes inferior a 5% e RSSI de -85 dB, indicando um enlace satisfatório e com margem de segurança.

Por fim, para a validação da força do sinal RSSI obtido nos testes, foi feito um gráfico comparativo entre os dados práticos e teóricos, calculados através da Equação 2.5, que está apresentado na Figura 43. O valor de n , utilizado no cálculo, foi encontrado utilizando regressão linear dos dados conhecidos de RSSI na equação descrita, resultando em $n = 2,6$, o que é coerente com ambientes rurais de visada livre (MACEDO et al., 2019).

Figura 43 – Comparação entre valores teóricos e experimentais de RSSI



Fonte: Autoria própria.

5.2.4 Síntese dos Resultados

Os ensaios cobriram os principais módulos de forma progressiva, e os resultados foram, em geral, satisfatórios. A verificação em bancada confirmou a operação integrada do sistema sem perdas de pacotes. Na validação dos sensores, o termopar operou dentro das tolerâncias do fabricante, os sensores Hall responderam de forma rastreável, com as limitações mecânicas discutidas, e o acelerômetro se comportou conforme esperado. O resultado mais relevante do ponto de vista do projeto, contudo, foi o enlace LoRa: a taxa de perda inferior a 5% a 500 m em campo aberto valida a hipótese central do trabalho e demonstra viabilidade para aplicação em competição.

6 CONCLUSÃO

6.1 Trabalhos futuros

Os resultados experimentais apontam para melhorias que podem ser exploradas em trabalhos futuros. Para os sensores Hall, a construção de suportes mecânicos robustos e a fixação definitiva dos ímãs na polia e no semieixo devem reduzir as oscilações observadas nas leituras de RPM e velocidade. Adicionalmente, a substituição da estratégia de aquisição por contagem baseada em eventos, em vez de intervalos discretos, possibilitaria respostas mais rápidas a transientes, a custo de maior uso de processamento. No enlace LoRa, o uso de antenas de maior ganho e a redução do tamanho do pacote transmitido são alternativas diretas para estender o alcance além de 500 m sem comprometer a taxa de transmissão. O ajuste do fator de perda de percurso a partir das medições experimentais, conforme discutido na análise do modelo teórico, também pode auxiliar no dimensionamento de futuras configurações. Por fim, a integração definitiva do sistema no veículo BAJA UFSCar, incluindo a fixação da placa, o roteamento dos chicotes e a operação durante uma prova completa, representa o passo natural de validação em condições reais de competição.

6.2 Considerações finais

Este trabalho desenvolveu e validou experimentalmente um sistema embarcado de monitoramento e telemetria para veículos BAJA SAE, com comunicação LoRa e gerenciamento por FreeRTOS. A questão central era saber se o alcance de 500 m, requisito imposto pelo formato das competições, seria viável com módulos comerciais de baixo custo. Os resultados de campo responderam afirmativamente. O hardware foi projetado, fabricado e montado em uma placa de circuito impresso de 152 mm x 100 mm, integrando os subsistemas de alimentação, processamento, sensoramento e comunicação sem fio. O firmware foi estruturado sobre FreeRTOS com distribuição de tarefas entre os dois núcleos do ESP32, garantindo separação entre as tarefas de aquisição, com requisitos temporais previsíveis, e as tarefas de comunicação e armazenamento, com maior tolerância a latência. A validação experimental confirmou o funcionamento integrado do sistema. Os sensores apresentaram leituras coerentes com os estímulos aplicados e dentro das tolerâncias especificadas. O enlace LoRa atendeu ao requisito de cobertura de 500 m em campo aberto, com taxa de perda inferior a 5%, estabelecendo uma base sólida para operação em competições BAJA

SAE. As principais limitações identificadas referem-se à sensibilidade dos sensores Hall a variações mecânicas de instalação e à redução da qualidade do enlace a 1000 m, onde a taxa de perda de pacotes superou 11%. Ambas possuem caminhos de solução mapeados e não comprometem a aplicação dentro do escopo definido. O sistema desenvolvido representa uma solução de baixo custo, com autonomia estimada superior a 13 horas em consumo de pico, adequada ao contexto de equipes universitárias com restrições orçamentárias, e extensível a futuras melhorias de hardware e firmware.

Referências

- ADVANCED MONOLITHIC SYSTEMS. **AMS1117**. s.d. Disponível em: <<http://www.advanced-monolithic.com/pdf/ds1117.pdf>>. Acesso em: 15 abr. 2026.
- ALLEGRO. **3144 - SENSITIVE HALL-EFFECT SWITCHES FOR HIGH-TEMPERATURE OPERATION**. s.d. Disponível em: <<https://www.alldatasheet.com/datasheet-pdf/pdf/55092/ALLEGRO/A3144.html>>. Acesso em: 5 abr. 2026.
- ALTIVUM. **Por que o Altium Designer pode ajudar a projetar PCBs de forma intuitiva e inteligente**. 2024. Disponível em: <https://arquivos.saebrasil.org.br/2026/BajaNacional/RATBSB_emenda_07.pdf>. Acesso em: 12 abr. 2026.
- AMERINI, I. **Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication**. 2016. Disponível em: <https://www.researchgate.net/figure/MEMS-accelerometer-how-it-works_fig2_308074268>. Acesso em: 5 abr. 2026.
- ANFAVEA. **Anuário Anfavea 2025**. 2025. Disponível em: <https://anfavea.com.br/site/wp-content/uploads/2025/04/DIGITAL-ANUARIO-2025altafinal_compressed-1.pdf>. Acesso em: 11 abr. 2026.
- BARR, M. **Programming Embedded Systems in C and C++**. Sebastopol, EUA: O'Reilly Media, 1999.
- CATALEX. **Micro SD Card Micro SDHC Mini TF Card Adapter Reader**. s.d. Disponível em: <<https://cdn.awsli.com.br/945/945993/arquivos/Datasheet-MicroSD-Module.pdf>>. Acesso em: 11 abr. 2026.
- DAWOUD, D. S. **Digital System Design — Use of Microcontroller**. Aalborg, Dinamarca: River Publishers, 2010.
- DENARDIN, G. W.; BARRIQUELLO, C. H. **Sistemas operacionais de tempo real e sua aplicação em sistemas embarcados**. São Paulo: Blucher, 2019.
- DESMARAIS, R.; BREUER, J. **How to Select and Use the Right Temperature Sensor**. 2001. Disponível em: <<https://www.fiercesensors.com/components/how-to-select-and-use-right-temperature-sensor>>. Acesso em: 4 abr. 2026.
- EBYTE. **E220-900T22D User Manual**. 2020. Disponível em: <https://www.popolony2k.com.br/wp-content/uploads/2023/02/E220-900T22D_UserManual_EN_v1.0.pdf>. Acesso em: 11 abr. 2026.
- ELETROGATE. **Eletrogate - Soluções em Energia**. s.d. Disponível em: <[https://www.eletrogate.com/?utm_source=google&utm_medium=cpc&utm_campaign=\[MC4\]_\[G\]_\[Search\]_Institucional&utm_content=666365205786&utm_term=eletrogate&gad_source=1&gad_campaignid=1843381086&gclid=CjwKCAjwqubPBhBOEi](https://www.eletrogate.com/?utm_source=google&utm_medium=cpc&utm_campaign=[MC4]_[G]_[Search]_Institucional&utm_content=666365205786&utm_term=eletrogate&gad_source=1&gad_campaignid=1843381086&gclid=CjwKCAjwqubPBhBOEi)>.

wAzgZX2p23qAR_Yg4qCr11AH9wQ86UQf574nIC0DpCKyR2a94I6d0kE6TsTRoCPZkQAvD_BwE>. Acesso em: 11 abr. 2026.

EMBARCADOS. **ESP32 - Analisando e corrigindo o ADC interno**. 2019. Disponível em: <<https://embarcados.com.br/esp32-adc-interno/>>. Acesso em: 4 abr. 2026.

INMET. **Dados Meteorológicos - Estação Meteorológica Automática de São Paulo**. 2025. Disponível em: <<https://portal.inmet.gov.br/dadoshistoricos>>. Acesso em: 11 abr. 2026.

INSTITUTO NCB. **Como funcionam os sensores de Efeito Hall (ART1050)**. s.d. Disponível em: <<https://www.newtoncbraga.com.br/index.php/como-funciona/6640-como-funcionam-os-sensores-de-efeito-hall-art1050>>. Acesso em: 5 abr. 2026.

INVENSENSE. **MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.0**. 2013. Disponível em: <<https://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/RM-MPU-6000A.pdf>>. Acesso em: 5 abr. 2026.

KOPETZ, H. **Real-Time Systems: Design Principles for Distributed Embedded Applications**. 2. ed. New York, EUA: Springer, 2011.

KVASER. **Kvaser USBcan Pro 2xHS v2**. s.d. Disponível em: <https://pim.kvaser.com/var/assets/Datasheet/73-30130-00752-9/00752-9_Kvaser%20USBcan%20Pro%202xHS%20v2_A4.pdf>. Acesso em: 10 abr. 2026.

LORINCZ, J. et al. A comprehensive analysis of lora network wireless signal quality in indoor propagation environments. **J. Sens. Actuator Netw**, v. 111, n. 6, 2025.

MACEDO, R. da S. et al. Modelagem inteligente de perda de percurso utilizando transceptores lora. **XXXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS**, 2019.

MARMIROLI, G. **Baja UFSCar - Repositório do Projeto**. 2026. Disponível em: <<https://github.com/guimarmirolli/bajaufscar001>>. Acesso em: 4 abr. 2026.

MAXIN INTEGRATED. **MAX6675**. 2021. Disponível em: <<https://www.analog.com/media/en/technical-documentation/data-sheets/max6675.pdf>>. Acesso em: 4 abr. 2026.

MINIPA. **Multímetro Digital MS-10**. 2020. Disponível em: <<https://pt.scribd.com/document/678422553/Ms-10-1300>>. Acesso em: 11 abr. 2026.

MONTAGNY, S. **LoRa-LoRaWAN and Internet of Things for beginners**. Chambéry, FR: Université Savoie Mont Blanc, 2022.

NAIK, A. S.; REDDY, S. K.; MANDELA, G. R. A systematic review on implementation of internet-of-things-based system in underground mines to monitor environmental parameters. **Journal of The Institution of Engineers (India): Series D**, v. 105, n. 2, 2024.

NXP SEMICONDUCTORS. **High-Precision Calibration of a Three-Axis Accelerometer**. 2020. Disponível em: <<https://www.nxp.com/docs/en/application-note/AN4399.pdf>>. Acesso em: 5 abr. 2026.

SAE BRASIL. **REGULAMENTO BAJA SAE BRASIL (RATBSB) – EMENDA 7**. 2025. Disponível em: <https://arquivos.saebrasil.org.br/2026/BajaNacional/RATBSB_emenda_07.pdf>. Acesso em: 4 abr. 2026.

TEXAS INSTRUMENTS. **LM2596**. 2023. Disponível em: <<https://www.ti.com/lit/ds/symlink/lm2596.pdf>>. Acesso em: 15 abr. 2026.

VISHAY. **TLDR5800**. s.d. Disponível em: <<https://www.vishay.com/docs/83004/tldr5800.pdf>>. Acesso em: 11 abr. 2026.

WILSON, J. **Sensor Technology Handbook**. Burlington, EUA: Newnes, 2005.