



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA
DE CONTROLE E AUTOMAÇÃO

Caio Judá da Cunha Melo

**Desenvolvimento de um Sistema Integrado de Monitoramento e Telemetria
Veicular Utilizando a Rede de Comunicação LoRa e o Protocolo CAN**

Recife
2025

Caio Judá da Cunha Melo

**Desenvolvimento de um Sistema Integrado de Monitoramento e Telemetria
Veicular Utilizando a Rede de Comunicação LoRa e o Protocolo CAN**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Engenharia de Controle e Automação da
Universidade Federal de Pernambuco,
como requisito parcial para obtenção do
grau de Bacharel.

Orientador(a): Prof. Dr. Douglas Contente Pimentel Barbosa

Recife
2025

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

Melo, Caio Judá da Cunha.

Desenvolvimento de um Sistema Integrado de Monitoramento e Telemetria Veicular Utilizando a Rede de Comunicação LoRa e o Protocolo CAN / Caio Judá da Cunha Melo. - Recife, 2025.

57 : il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa

(Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, , 2025.

Inclui referências, anexos.

1. Telemetria veicular. 2. LoRa. 3. Protocolo CAN. 4. Monitoramento em tempo real. 5. Veículos leves. 6. Internet das Coisas (IoT). I. Barbosa, Douglas Contente Pimentel. (Orientação). II. Título.

620 CDD (22.ed.)

Caio Judá da Cunha Melo

**Desenvolvimento de um Sistema Integrado de Monitoramento e Telemetria
Veicular Utilizando a Rede de Comunicação LoRa e o Protocolo CAN**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Engenharia de Controle e Automação da
Universidade Federal de Pernambuco,
como requisito parcial para obtenção do
grau de Bacharel.

Aprovado em: 16/08/2025.

BANCA EXAMINADORA

Prof. Dr Douglas Contente Pimentel Barbosa (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Artur Muniz Szpak Furtado (Examinador Interno)
Universidade Federal de Pernambuco

Eng. M. Sc. Néstor Iván Medina Giraldo (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Este trabalho é dedicado a todos que, de alguma forma, contribuíram para a realização deste projeto. Agradeço imensamente aos meus professores, colegas, amigos e familiares, pelo apoio, incentivo e orientação durante toda a jornada acadêmica. A todos que compartilharam seus conhecimentos e experiências, ajudando-me a crescer tanto pessoal quanto profissionalmente, minha eterna gratidão. Agradeço à Usina Petribu pelo financiamento e auxílio imprescindíveis para a concretização deste projeto.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema integrado de monitoramento e telemetria veicular voltado para veículos leves, utilizando a rede de comunicação LoRa (*Long Range*) e o protocolo CAN (*Controller Area Network*) com foco na análise de dados. A proposta visa oferecer uma solução de baixo custo, eficiente e com grande alcance para aplicações em frotas ou monitoramento individual de veículos, superando limitações de sistemas tradicionais baseados exclusivamente em redes móveis. O sistema é composto por módulos embarcados capazes de coletar dados diretamente da rede CAN dos veículos, tais como velocidade, rotação do motor e status de sensores, transmitindo essas informações via LoRa a uma central receptora. Esta, por sua vez, realiza o encaminhamento dos dados para um servidor responsável pela visualização e análise em tempo real. O uso da tecnologia LoRa permite a comunicação entre os veículos e a central mesmo em locais com baixa cobertura de internet, favorecendo a utilização em áreas urbanas e rurais. Os testes realizados demonstraram estabilidade na transmissão dos dados, baixo consumo de energia dos dispositivos e precisão nas informações capturadas. Como resultado, o sistema se mostra viável para aplicações de monitoramento contínuo, gestão de desempenho veicular e tomada de decisão preventiva quanto à manutenção dos veículos. O trabalho também discute as possibilidades de escalabilidade e integração com os possíveis sistemas propostos para a telemetria veicular.

Palavras-chave: Telemetria veicular; LoRa; protocolo CAN; monitoramento em tempo real, veículos leves, Internet das Coisas (IoT).

ABSTRACT

This work presents the development of an integrated vehicle monitoring and telemetry system designed for light vehicles, using the LoRa communication network and the CAN protocol with a focus on real-time data analysis. The proposed solution aims to offer a low-cost, efficient, and long-range alternative for fleet applications or individual vehicle monitoring, overcoming limitations of traditional systems based solely on mobile networks. The system consists of embedded modules capable of collecting data directly from the vehicle's CAN network—such as speed, engine RPM, and sensor status—and transmitting this information via LoRa to a receiving unit. This receiver then forwards the data to a server responsible for visualization and real-time analysis. The use of LoRa technology enables communication between vehicles and the central unit even in areas with limited internet coverage, supporting operation in both urban and rural settings. Tests conducted demonstrated stable data transmission, low power consumption, and high accuracy in the collected information. As a result, the system proves to be viable for continuous monitoring applications, vehicle performance management, and preventive maintenance decision-making. The study also discusses the possibilities of scalability and integration with the possible systems proposed for vehicle telemetry.

Keywords: Vehicle telemetry; LoRa; CAN protocol; real-time monitoring; light vehicles; Internet of Things (IoT).

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura Geral do Sistema.	26
Figura 2 - Pinos Conector OBD2.	30
Figura 3 – Conector OBD2 físico.	30
Figura 4 - Módulo GPS NEO-6M.	32
Figura 5 - Módulo Transceptor CAN SN65HVD230.	33
Figura 6 - Pinos Lora32 TTGO.	34
Figura 7 - Ligação do Transmissor.	34
Figura 8 – Invólucro físico do circuito.	36
Figura 9 - Código para sincronização da tarefa CAN.	40
Figura 10 - Parte superior PCB	42
Figura 11 - Parte Inferior PCB.	43
Figura 12 - PCB idealizada.	44
Figura 13 - Envio e recepção via LoRa.	45
Figura 14 - Confirmação de recepção (ACK).	45
Figura 15 - Ambiente para verificação dos vizinhos.	46
Figura 16 - Verificação dos vizinhos.	46
Figura 17 - Transmissão ponto a ponto.	47
Figura 18 - Ganho de relevo da maior distância.	48
Figura 19 - Maior distância alcançada entre veículo e central de monitoramento.	48
Figura 20 - Coleta de dados CAN.	49
Figura 21 - Coleta de dados CAN, 2.	50
Figura 22 - Zoom da conexão.	51
Figura 23 - Mensagem coletada 1.	52
Figura 24 - Mensagem coletada 2.	52

LISTA DE TABELAS

Tabela 1 - Demais Componentes	27
Tabela 2 – Módulos LoRa.	28
Tabela 3 - Detalhamento OBD2	31

LISTA DE ABREVIATURAS E SIGLAS

3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
ACK	Acknowledgment
ADC	Analog to Digital Converter
API	Application Programming Interface
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CAN FD	Controller Area Network Flexible Data rate
CSS	chirp spread spectrum
ECU	Electronic Control Unit
ESP IDF	Espressif IoT Development Framework
ESP32	Espressif Systems Protocol 32 bit
GPS	Global Positioning System
GND	Ground
I2C	Inter Integrated Circuit
ID	Identifier
IDE	Integrated Development Environment
IoT	Internet of Things
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
MAC	Media Access Control
MQTT	Message Queuing Telemetry Transport
OBD	On Board Diagnostics
PA	Power Amplifier
PCB	Printed Circuit Board
PID	Parameter Identifier
PWM	Pulse Width Modulation
RAM	Random Access Memory

RSSI	Received Signal Strength Indicator
RX	Receiver
SAE	Society of Automotive Engineers
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SNR	Signal-to-Noise Ratio
TCP/IP	Transmission Control Protocol Internet Protocol
TX	Transmitter
UART	Universal Asynchronous Receiver Transmitter
VCC	Voltage at the Common Collector
VHF	Very High Frequency
Wi-Fi	Wireless Fidelity

LISTA DE SÍMBOLOS

V	Volts
A	Ampère
MHz	Mega Hertz
mA	Miliampère
μ A	Micro
GHz	Gigahertz
bps	Bits por segundo
°C	Graus Celsius
kB	Quilobyte
dBm	Decibel-milliwatts
h	Horas

Sumário

1	INTRODUÇÃO	13
1.1	OBJETIVOS	16
1.1.1	Geral	16
1.1.2	Específicos	17
1.2	ORGANIZAÇÃO DO TRABALHO.....	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	MICROCONTROLADOR ESP32	19
2.2	CAN (<i>CONTROLLER AREA NETWORK</i>).....	20
2.3	GPS (<i>GLOBAL POSITIONING SYSTEM</i>)	21
2.4	LORA (<i>LONG RANGE</i>).....	21
2.5	REDE EM MALHA (<i>MESH</i>)	22
2.6	PROTOCOLO DE TRANSMISSÃO ENTRE MICROCONTROLADOR E SERVIDOR	24
3	METODOLOGIA	25
3.1	ARQUITETURA E COMPONENTES.....	25
3.2	LIGAÇÕES DO SISTEMA	29
4	DESENVOLVIMENTO DO TRABALHO	37
4.1	COMUNICAÇÃO CAN AO USAR O MICROCONTROLADOR	37
4.2	COMUNICAÇÃO ENTRE 2 MICROCONTROLADORES VIA LORA	38
4.3	ENVIO DE DADOS CAN DO TRANSMISSOR AO RECEPTOR VIA LORA	39
4.4	CONFECÇÃO DA PLACA DE CIRCUITO IMPRESSO	41
5	RESULTADOS.....	42
5.1	CONFECÇÃO DA PLACA DE CIRCUITO IMPRESSO	42
5.2	COMUNICAÇÃO ENTRE 2 MICROCONTROLADORES VIA LORA	44
5.3	ENVIO DE DADOS CAN DO TRANSMISSOR AO RECEPTOR VIA LORA	46
5.4	COMUNICAÇÃO CAN AO USAR O MICROCONTROLADOR	49
6	CONCLUSÕES E PROPOSTAS DE CONTINUIDADE.....	53
7	REFERÊNCIAS.....	55

1 INTRODUÇÃO

Atualmente, empresas do setor sucroalcooleiro dependem de prestadores de serviço terceirizados para realizar a telemetria veicular. Esse sistema é essencial para acompanhar, em tempo real, informações operacionais dos veículos, como posição geográfica, nível de combustível, rotação do motor, velocidade, tempo de operação, entre outros dados críticos para a gestão da frota.

A comunicação dos equipamentos de telemetria é feita utilizando a rede móvel 3G (*Third Generation*), 4G (*Fourth Generation*), 5G (*Fifth Generation*), em conjunto com sistemas de transmissão via satélite. No entanto, a operação enfrenta desafios significativos no que se refere à estabilidade e confiabilidade da comunicação de dados. Especialmente no estado de Pernambuco, estão localizadas as plantações de cana-de-açúcar, o relevo possui características irregulares, com muitas elevações, morros e áreas de densa vegetação, o que resulta na formação de zonas de sombra, onde o sinal da rede celular se torna instável ou até inexistente. Essas falhas de cobertura comprometem diretamente a eficiência do monitoramento da frota, dificultando não só a coleta precisa dos dados em tempo real, mas também a troca de informações essenciais entre os veículos e o centro de controle da usina. Com isso, ocorrem perdas na visibilidade operacional, dificultando a tomada de decisões rápidas e impactando na logística, produtividade e controle dos custos operacionais.

Outrossim, problemas semelhantes ocorrem em diversas aplicações críticas que dependem de monitoramento remoto confiável. Um exemplo recente é o desenvolvimento de um sistema autônomo de drones para rastreamento rápido e robusto da fauna em terrenos complexos (CHEN, 2023).

De forma complementar, estudos analisaram o efeito das árvores de um pomar na comunicação ponto a ponto via LoRa entre sensores e um gateway, mostrando que a vegetação reduz significativamente a qualidade do sinal e a taxa de entrega dos dados, especialmente em maiores distâncias, o que ressalta a necessidade de ajustes para garantir a eficiência em ambientes densamente arborizados (AHMED, 2024).

Na mesma linha, pesquisas em ambientes controlados, como estufas agrícolas, evidenciaram que fatores internos como temperatura e umidade também podem interferir na transmissão dos dados via LoRaWAN (*Long Range Wide Area Network*). Nesse sentido, foram testados diferentes invólucros, desde os mais fechados até os

mais abertos, e aquele com melhor circulação de ar reduziu as interferências, evidenciando a importância de adaptar as soluções às condições específicas do ambiente para garantir a confiabilidade do monitoramento (SINGH, 2020).

Ampliando a discussão, observa-se que as redes LPWAN (*Low-Power Wide-Area Network*), caracterizadas por baixo consumo de energia e grande alcance, têm ganhado relevância devido à sua eficiência, custo reduzido e cobertura extensa. Essas redes superam tecnologias convencionais de curto alcance e redes celulares em aplicações que incluem cidades inteligentes, comunicação M2M (*Machine-to-Machine*), saúde, transporte, agricultura e rastreamento de ativos. O estudo aborda os objetivos de projeto, diferentes arquiteturas como topologias em estrela e malha, as principais tecnologias existentes, além de analisar a coexistência com outras redes, comparar seus benefícios e limitações, identificar oportunidades de mercado e discutir desafios para avanços futuros (CHILAMKURTHY, 2022).

Diante desses desafios, o avanço de soluções abertas e adaptáveis tem se mostrado promissor. Nesse contexto, o *OPEnS Hub* (DEBELL, 2017) apresenta uma alternativa eficiente para coleta e transmissão de dados em tempo real, mesmo em áreas remotas e de difícil acesso. Utilizando tecnologias como rádio LoRa e nRF (*Nordic Radio Frequency*) para comunicação de longa distância e alta taxa de dados, além de suportar conexões via Ethernet, Wi-Fi ou redes celulares, o sistema envia informações diretamente para planilhas do Google por meio de serviços gratuitos, facilitando o acesso remoto. Testado com sucesso em uma rede de sensores aplicação em uma rede de sensores no estado do Oregon comprovou sua viabilidade como uma solução acessível e eficaz para enfrentar os desafios da comunicação em ambientes complexos.

Com base na análise desses estudos, a ideia de utilizar uma rede *mesh* — composta por vários nós interligados para ampliar a cobertura e a confiabilidade — ganhou destaque e será adotada como base para a solução proposta.

Nesse sentido, assim como os veículos da indústria sucroalcooleira enfrentam zonas de sombra na comunicação, os pesquisadores também enfrentavam dificuldades para localizar animais em terrenos irregulares, onde os métodos tradicionais de telemetria VHF (*Very High Frequency*) apresentavam limitações. Para contornar esses desafios, a solução proposta combina:

- Medições híbridas (RSSI – *Received Signal Strength Indicator* e SNR – *Signal to Noise Ratio*) para compensar limitações individuais de cada tecnologia;
- Algoritmos adaptativos que priorizam o método mais eficiente conforme as condições do terreno;
- Restrições operacionais para manter distância segura dos alvos (evitando interferência).

Essa abordagem demonstrou ganhos de até 9 vezes em velocidade de localização e precisão de 2 a 40 metros mesmo em áreas com obstruções, superando métodos manuais. Para a empresa do setor agroindustrial, adaptações desses princípios poderiam inspirar soluções como:

- Sistemas de comunicação redundantes (celular + satélite + redes em malha);
- Processamento local de dados em veículos para transmissão diferida.

Com o avanço da eletrônica embarcada e das tecnologias de comunicação sem fio, os sistemas de monitoramento e telemetria veicular vêm ganhando destaque em aplicações que exigem coleta e análise de dados em tempo real, como controle de frotas, segurança veicular e manutenção preditiva. Nesse contexto, este trabalho propõe o desenvolvimento de um sistema integrado de telemetria para veículos leves, capaz de coletar dados diretamente da rede interna do veículo por meio do protocolo CAN (*Controller Area Network*) e transmiti-los utilizando a tecnologia de rádio LoRa (*Long Range*), conhecida por sua baixa taxa de transmissão, mas alta eficiência energética e longo alcance, especialmente em áreas rurais.

A arquitetura do sistema contempla a utilização do microcontrolador ESP32 (*Espressif Systems Protocol 32 bit*), escolhido por sua conectividade Wi-Fi (*Wireless Fidelity*), suporte à comunicação SPI (*Serial Peripheral Interface*) e compatibilidade com uma vasta gama de bibliotecas para manipulação do barramento CAN e protocolos de rede. A alimentação elétrica é proveniente da própria bateria do veículo (12 V), sendo realizada a conversão para 3,3 V por meio de um regulador de tensão, a fim de garantir a integridade dos componentes sensíveis. O projeto, inicialmente desenvolvido em protoboard, foi posteriormente transferido para PCB (*Printed Circuit Board*), otimizando o espaço e conferindo maior confiabilidade à solução.

Complementando a coleta de dados operacionais do veículo, módulos de rastreamento com suporte a GPS (*Global Position System*) são integrados ao sistema para registrar a localização geográfica em tempo real. As informações são então transmitidas via protocolo MQTT (*Message Queuing Telemetry Transport*) a um servidor, seguindo o modelo de publicação/assinatura que viabiliza a comunicação assíncrona e escalável entre os dispositivos.

Estudos recentes demonstram que a utilização de microcontroladores como o ESP32, aliados ao protocolo MQTT, permite não apenas a transmissão eficiente dos dados do barramento CAN para servidores remotos, mas também a implementação de funcionalidades avançadas de diagnóstico veicular, como a leitura de códigos de falha e parâmetros operacionais em tempo real, seguindo padrões como o OBD-II (NIKOLOV e GOTSEVA, 2004). Isso amplia significativamente as possibilidades de aplicação da solução proposta, viabilizando não só o monitoramento de desempenho, mas também a identificação precoce de falhas, mesmo em locais de difícil acesso ou baixa conectividade.

Assim, o projeto visa oferecer uma solução de baixo custo, escalável e de fácil adaptação a diferentes tipos de veículos leves, contribuindo para o fortalecimento de tecnologias nacionais voltadas à mobilidade inteligente, à gestão eficiente de dados veiculares e ao uso racional dos recursos em áreas rurais e remotas.

1.1 Objetivos

1.1.1 Geral

Desenvolver um sistema de monitoramento e telemetria veicular para veículos leves, utilizando o protocolo CAN e a comunicação LoRa, com foco na transmissão, análise de dados em tempo real em curtas e longas distâncias e armazenamento dos mesmos.

1.1.2 Específicos

- Estabelecer a comunicação entre um microcontrolador e o sistema eletrônico do veículo utilizando o barramento CAN, por meio da configuração adequada de um transceptor - ou controlador - CAN (SN65HVDV230), da definição dos parâmetros de comunicação e da integração com o veículo para leitura de dados relevantes, como rotações do motor, velocidade, ou códigos de falha;

- Estabelecer a comunicação sem fio entre dois microcontroladores por meio da tecnologia LoRa, configurando um deles como transmissor e o outro como receptor, com foco em garantir a confiabilidade na troca de dados, ajustando parâmetros como potência de saída, fator de espalhamento, banda de frequência e técnicas de verificação de integridade ,como CRC (*Cyclic Redundancy Check*) – um método utilizado para detectar erros em dados transmitidos – e ACK (*Acknowledgment*) – uma confirmação de que os dados foram recebidos com sucesso. Além disso, validar a comunicação em diferentes distâncias e ambientes;

- Transmitir dados do veículo para um microcontrolador receptor via LoRa, avaliando o desempenho em curtas (até 1 km) e longas distâncias (acima de 1 km). O objetivo é garantir integridade dos dados, baixo consumo de energia e comunicação estável, ajustando potência, antena e modulação conforme o ambiente e a distância;

- Projetar uma PCB compacta que integre o microcontrolador, o transceptor CAN, o módulo LoRa e outros componentes eletrônicos, garantindo robustez, baixo consumo e compatibilidade com o ambiente automotivo. O layout deve seguir boas práticas de sinal e alimentação, prever conectores adequados, proteção contra surtos e facilitar testes e produção.

1.2 Organização do Trabalho

Inicialmente, o Capítulo 1 (Introdução), referente à Introdução, apresenta o contexto geral do projeto, destacando a relevância do monitoramento veicular em áreas remotas, a motivação que levou ao desenvolvimento da solução proposta, o objetivo geral e os objetivos específicos que orientam o trabalho, bem como a descrição da estrutura dos capítulos que compõem o documento.

O Capítulo 2 (Fundamentação teórica) reúne os conceitos e tecnologias essenciais para o desenvolvimento do sistema citando as características do microcontrolador ESP32, o funcionamento do protocolo CAN utilizado na comunicação veicular, a tecnologia LoRa empregada na transmissão de dados sem fio em longas distâncias, além de outros tópicos relevantes da eletrônica embarcada que sustentam a base teórica do projeto.

O Capítulo 3 (Metodologia) descreve as etapas adotadas no desenvolvimento do projeto, incluindo os critérios para seleção dos componentes eletrônicos, a definição da arquitetura do sistema de comunicação, os procedimentos de montagem e conexão dos módulos físicos, bem como as abordagens utilizadas para validação funcional durante o processo.

O Capítulo 4 (Desenvolvimento) detalha a implementação prática do sistema proposto, apresentando a integração entre os módulos de comunicação CAN e LoRa com o microcontrolador ESP32, a estruturação do código-fonte utilizado, as configurações específicas de hardware, e os ajustes realizados tanto no software quanto na montagem dos circuitos eletrônicos para garantir o correto funcionamento da solução.

O Capítulo 5 (Resultados) apresenta os testes experimentais realizados em diferentes cenários de operação, tanto em curta quanto em longa distância, incluindo a coleta e interpretação dos dados transmitidos, a análise do desempenho da comunicação entre os módulos, e a verificação da confiabilidade e estabilidade do sistema em condições reais de uso.

O Capítulo 6 (Conclusão e propostas de continuidade) expõe desfechos obtidos com a realização do trabalho, refletindo sobre os objetivos alcançados, as limitações técnicas ou operacionais identificadas durante o desenvolvimento, sugerindo possíveis aprimoramentos na arquitetura ou nos métodos utilizados e indicando caminhos viáveis para a continuidade ou expansão da pesquisa em futuros projetos acadêmicos ou aplicações práticas.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados os conceitos e tecnologias fundamentais que sustentam o desenvolvimento deste trabalho. Serão abordados aspectos técnicos do microcontrolador ESP32, protocolo CAN, sistemas de posicionamento GPS, tecnologia de comunicação LoRa e sua aplicação em redes em malha, além do protocolo MQTT para transmissão de dados entre dispositivos embarcados e servidores. A compreensão desses tópicos é essencial para contextualizar a proposta de monitoramento veicular remota e as escolhas técnicas adotadas no projeto.

2.1 Microcontrolador ESP32

O ESP32 é um microcontrolador de alta performance desenvolvido pela Espressif Systems, amplamente utilizado em aplicações de Internet das Coisas (IoT - *Internet of things*), automação e sistemas embarcados com boa conectividade, recursos integrados, alta capacidade de processamento e versatilidade (BABIUCH, FOLTÝNEK e SMUTNÝ, 2019). Ele possui um processador dual-core Xtensa LX6 operando até 240 MHz, com até 520 kB de RAM e suporte à memória flash externa. Destaca-se pela integração nativa de múltiplas interfaces de comunicação, como Wi-Fi (IEEE 802.11 b/g/n) e Bluetooth (*Bluetooth 4.2 Classic* e *Bluetooth Low Energy* – BLE) e diversos protocolos industriais, incluindo funcionalidades como UART (*Universal Asynchronous Receiver/Transmitter*), I2C (*Inter-Integrated Circuit*), PWM (*Pulse Width Modulation*), ADC (*Analog-to-Digital Converter*) e CAN, ampliando sua versatilidade para comunicação com sensores e atuadores. Ademais, é altamente eficiente energeticamente, com modos de economia que o tornam ideal para sistemas remotos ou que dependem de fontes limitadas, como baterias veiculares e painéis solares.

Em comparação, Arduino (como o Arduino Uno) é geralmente baseado em microcontroladores AVR de 8 bits, com menor capacidade de processamento (16 MHz) e memória limitada, possuindo interfaces mais simples e sem conectividade nativa Wi-Fi ou Bluetooth (BANZI, 2022). Isso o torna mais adequado para projetos educativos ou prototipagem simples, mas menos eficiente para aplicações IoT

complexas ou que exigem conectividade sem fio. Por outro lado, os STM32 são microcontroladores de 32 bits da *STMicroelectronics*, com arquitetura ARM Cortex-M, oferecendo desempenho variável (até centenas de MHz), ampla gama de periféricos e maior eficiência energética em aplicações industriais (PAKDEL, 2020). Eles proporcionam maior flexibilidade para sistemas embarcados complexos, mas geralmente exigem maior conhecimento técnico para configuração e programação em comparação com o ESP32.

A popularidade do ESP32 também decorre da ampla comunidade de desenvolvedores e do suporte a diversas plataformas de desenvolvimento, como Arduino IDE (BHOOKYA, 2022), PlatformIO (DEHROUYEH, 2024), MicroPython (PLAUSKA, LIUTKEVIČIUS e JANAČIČIŪTĖ, 2023) e ESP-IDF (*Espressif IoT Development Framework*) (HOANG, VAN e NGUYEN, 2019), facilitando a prototipagem e implementação de soluções embarcadas.

2.2 CAN (Controller Area Network)

O CAN é um protocolo de comunicação serial amplamente utilizado na indústria automotiva, com alta confiabilidade e baixa latência (RODRÍGUEZ, RAÚL e INOUE, 2018), para a troca de dados entre ECUs (*Electronic Central Units*). Existem variações desse protocolo, como o CAN clássico e o CAN FD (*Flexible Data-rate*), que ampliam a capacidade e velocidade da comunicação conforme a complexidade dos sistemas veiculares. No entanto, o foco deste trabalho está nos veículos leves, que tipicamente utilizam o padrão OBD-II (*On-Board Diagnostics II*) como interface de diagnóstico e monitoramento. O OBD-II, obrigatório em veículos leves produzidos a partir da década de 1990, geralmente utiliza o barramento CAN como meio físico para a comunicação entre a ECU e dispositivos externos. Por meio do OBD-II, é possível acessar diversos parâmetros do veículo utilizando os chamados comandos PID (*Parameter IDs*), que são códigos padronizados que solicitam informações específicas, como velocidade, rotação do motor, temperatura do fluido de arrefecimento e nível de combustível. Esses comandos são essenciais para sistemas de telemetria, manutenção preditiva e diagnóstico veicular. A utilização do barramento CAN via OBD-II em veículos leves possibilita o desenvolvimento de soluções eficientes para coleta de dados em tempo

real, contribuindo para a gestão de frotas, monitoramento remoto e otimização do desempenho veicular com baixo custo e alta precisão.

2.3 GPS (*Global Positioning System*)

O GPS é um sistema de satélites que permite determinar a localização precisa de um dispositivo em tempo real, em qualquer lugar do mundo. Em projetos embarcados, como a telemetria veicular, utilizam-se módulos GPS — dispositivos compactos que recebem sinais dos satélites e transmitem as coordenadas para microcontroladores, geralmente via interfaces como UART, SPI ou I2C. Esses módulos são fundamentais para o rastreamento e monitoramento de veículos, permitindo o acompanhamento de rotas e contribuindo para a segurança, além de facilitar a gestão eficiente de frotas quando integrados a sistemas embarcados como o ESP32.

2.4 LoRa (*Long Range*)

O LoRa é uma tecnologia de comunicação sem fio desenvolvida para transmitir dados a longas distâncias com baixo consumo energético. Utilizando a modulação CSS (*chirp spread spectrum*), essa tecnologia opera em ambientes onde a infraestrutura tradicional de redes é limitada ou inexistente, como áreas rurais, zonas industriais e regiões remotas. Funcionando em frequências não licenciadas na banda MHz, o LoRa pode alcançar distâncias superiores a 100 quilômetros, na teoria, em campo aberto (JOVALEKIC, 2018). Além do alcance, destaca-se pela resistência a interferências e obstáculos físicos, como relevo acidentado e vegetação densa, condições comuns em ambientes rurais.

No contexto da telemetria veicular, o LoRa permite a transmissão confiável de dados operacionais e informações de localização mesmo em locais com pouca ou nenhuma cobertura de redes móveis, configurando-se como uma solução eficiente para monitoramento remoto de frotas, coleta em tempo real e integração com sistemas de gestão. Complementarmente, o protocolo LoRaWAN (*Long Range Wide Area Network*) define as normas de comunicação e gerenciamento para redes em larga escala, garantindo segurança, escalabilidade e o controle eficaz dos dispositivos

conectados. A combinação do LoRa com microcontroladores, como o ESP32, possibilita o desenvolvimento de sistemas de baixo custo, baixa energia (SEMTECH, 2015) e alta confiabilidade, essenciais para aplicações em veículos leves em áreas rurais.

Experimentos realizados por (KOYANAGI, 2018) comprovam a viabilidade de transmissões estáveis em curtas distâncias (até 6.5 km), enquanto soluções com repetidores LoRa ampliam o alcance para além desse valor (MESHTASTIC, 2025). A referência ao fórum Meshtastic se justifica pelo fato de a tecnologia ainda ser relativamente recente, com pouca exploração formal sobre distâncias máximas em artigos científicos ou conferências. Dessa forma, a informação concreta mais próxima dos resultados desejados — relacionada ao alcance com uso de repetidores — foi encontrada nesse fórum técnico, baseado em testes práticos realizados pela comunidade especializada.

Em comparação com outras tecnologias LPWA, como SigFox e NB-IoT, observa-se que cada solução apresenta vantagens distintas. SigFox é eficiente para AMI em áreas urbanas e rurais, oferecendo boa cobertura, baixo número de gateways e sinais estáveis, embora fatores de posicionamento e ambientais precisem ser considerados na implementação real (PURNAMA e NASHIRUDDIN, 2022). A tecnologia não licenciada LoRa, por sua vez, apresenta benefícios em duração da bateria, capacidade e custo, enquanto a tecnologia licenciada NB-IoT se destaca por oferecer melhor qualidade de serviço, menor latência, maior confiabilidade e maior alcance (SINHA, WEI e HWANG, 2017). Neste projeto, o LoRa foi escolhido principalmente pela maior experiência do usuário com esta tecnologia, garantindo uma implementação mais ágil, confiável e de baixo custo.

2.5 Rede em malha (*Mesh*)

Uma rede em malha é uma arquitetura de comunicação na qual os dispositivos, ou nós, atuam não apenas como transmissores, mas também como retransmissores de dados, criando múltiplos caminhos alternativos para o destino. Essa característica aumenta a robustez da rede, pois a comunicação pode ser mantida mesmo quando alguns nós falham ou quando não há conexão direta entre origem e destino. Essa

abordagem é especialmente útil em ambientes com obstáculos físicos ou distribuição geográfica dispersa, onde a dependência de um único ponto central (como um *gateway*) pode limitar a cobertura (AKYILDIZ, WANG e WANG, 2005).

No contexto de tecnologias como o LoRa, que originalmente opera em topologia estrela, a implementação de uma malha via software pode superar limitações de alcance. Algoritmos de roteamento inteligente permitem que os nós retransmitam mensagens entre si, estendendo a área de cobertura sem aumentar a potência de transmissão individual. Estudos empíricos, como os conduzidos por (KOYANAGI, 2018), demonstram que, embora a comunicação direta entre nós em LoRa seja limitada a distâncias curtas (aproximadamente 6,5 km em condições ideais), a introdução de repetidores em uma malha pode ampliar significativamente o alcance total da rede (MESHTASTIC, 2025).

Essa estratégia é respaldada por pesquisas que analisam a capacidade de redes sem fio em cenários distribuídos. Trabalhos (GUPTA e KUMAR, 2020) destacam que, em redes *ad hoc*, a eficiência da comunicação depende não apenas da potência de transmissão, mas também da densidade de nós e da forma como o tráfego é distribuído. Eles mostram que, em ambientes com muitos dispositivos, a capacidade individual de cada nó diminui, mas a utilização de retransmissões em malha pode otimizar o uso do canal compartilhado, equilibrando carga e alcance. Essa abordagem é particularmente relevante para aplicações em IoT, onde a escalabilidade e a confiabilidade são críticas.

Além disso, a adoção de malhas em LoRa, como o módulo LoRaMESH da Radioenge, pode mitigar problemas de interferência e congestionamento, já que a comunicação é distribuída por múltiplos caminhos (RADIOENGE, 2021). Isso está alinhado com a observação de que, em redes sem fio, a capacidade total é limitada pela necessidade de compartilhamento do meio entre nós vizinhos (GUPTA e KUMAR, 2020). Assim, a malha não apenas estende o alcance, mas também melhora a resiliência em ambientes dinâmicos.

Em contrapartida, o ZigBee também é uma tecnologia de comunicação sem fio, mas projetada principalmente para curtas distâncias (FARAHANI, 2008), sendo ideal para redes de sensores e automação residencial, porém menos adequada para redes de longo alcance ou cenários distribuídos em larga escala, onde soluções como LoRaMESH se destacam.

A escolha das tecnologias de rede sem fio para este trabalho considerou critérios específicos de alcance e custo. Entre ZigBee e Meshtastic, o critério determinante foi a distância de comunicação, sendo o Meshtastic preferido por permitir maior cobertura (MESHTASTIC, 2025). Já entre Meshtastic e o módulo LoRaMESH da Radioenge, o custo foi determinante: enquanto a solução desse apresentava valores proibitivos para aplicação em larga escala, aquele - por ser uma plataforma de código aberto - ofereceu a combinação ideal de acessibilidade financeira e flexibilidade técnica para este projeto.

2.6 Protocolo de transmissão entre microcontrolador e servidor

O MQTT é um protocolo de comunicação leve, desenvolvido para atender às demandas de sistemas que operam em redes com baixa largura de banda, alta latência ou conexões instáveis — características comuns em aplicações de Internet das Coisas. Baseando-se em um modelo de comunicação assíncrona mediado por um broker, ele permite que dispositivos publiquem mensagens sem precisar conhecer diretamente os destinatários, que por sua vez recebem apenas os dados de interesse previamente definidos. Essa abordagem promove escalabilidade, economia de energia e redução no tráfego de rede, tornando-se ideal para plataformas embarcadas como o ESP32, frequentemente utilizadas na coleta de dados sensoriais e transmissão para servidores ou serviços em nuvem (AL-FUQAHA, 2015).

Em uma análise mais profunda, o modelo de comunicação adotado pelo MQTT é fundamentado no paradigma *publish/subscribe*, que se destaca pelo desacoplamento entre emissores e receptores em três dimensões: tempo, espaço e sincronização. Esse tipo de arquitetura favorece sistemas distribuídos mais flexíveis e escaláveis, permitindo que publicadores e assinantes interajam de forma assíncrona e independente (EUGSTER, 2003). Esse nível de desacoplamento é particularmente vantajoso em ambientes IoT, onde a conectividade pode ser instável e os dispositivos possuem recursos limitados.

Assim, a utilização do MQTT não é apenas tecnicamente viável, mas também respaldada por uma base teórica sólida que reforça sua eficácia em aplicações sensíveis à conectividade e à eficiência energética.

3 METODOLOGIA

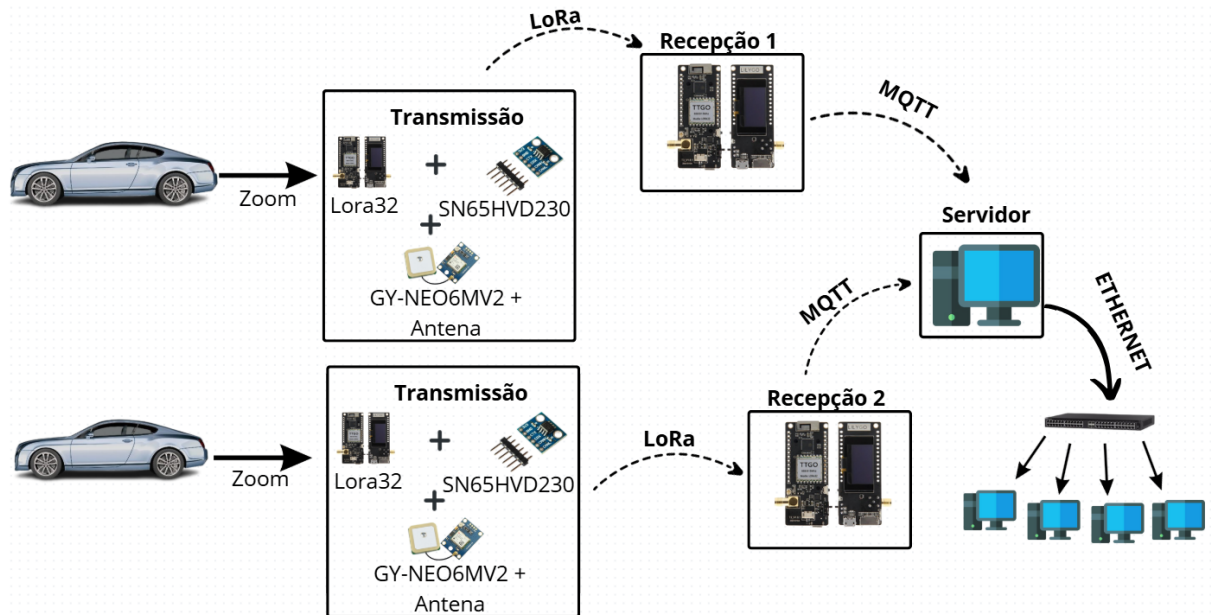
Este capítulo apresenta os procedimentos adotados para o desenvolvimento do sistema de telemetria veicular proposto. Inicialmente, será apresentada a arquitetura geral do sistema, detalhando seus principais blocos e fluxos de informação. Em seguida, serão descritos os componentes eletrônicos selecionados para a implementação, com justificativas técnicas para suas escolhas. Por fim, serão abordadas as etapas de montagem, integração dos módulos e os procedimentos para validação funcional do sistema.

3.1 Arquitetura e componentes

Este capítulo apresenta os procedimentos adotados para o desenvolvimento do sistema de telemetria veicular proposto. Inicialmente, será apresentada a arquitetura geral do sistema, detalhando seus principais blocos e fluxos de informação. Em seguida, serão descritos os componentes eletrônicos selecionados para a implementação, com justificativas técnicas para suas escolhas. Ressalta-se que as referências utilizadas para cada componente serão provenientes, em sua maioria, de fontes eletrônicas, uma vez que o trabalho envolve módulos e placas personalizadas fornecidas por fabricantes que disponibilizam as informações técnicas exclusivamente em seus sites oficiais. Por fim, serão abordadas as etapas de montagem, integração dos módulos e os procedimentos para validação funcional do sistema.

Inicialmente foi planejado a arquitetura geral do sistema presente na Figura 1 na qual aparece o veículo e um “zoom” dos equipamentos de transmissão que ficarão embarcados, a rede de transmissão que será utilizada, os receptores e o servidor que ficará ligado e transmitindo as informações para onde desejar.

Figura 1 - Arquitetura Geral do Sistema.



Fonte: Autor, 2025.

Por conseguinte, para a montagem do sistema de telemetria veicular foram utilizados diversos componentes que formam a arquitetura do projeto, cada um com funções específicas relacionadas à coleta, processamento e transmissão de dados. Entre os módulos embarcados, inclui-se o microcontrolador, responsável pelo gerenciamento da comunicação interna entre os componentes e pelo controle do fluxo de informações. Os transceptores LoRa compõem a camada de comunicação de longo alcance, permitindo a transmissão dos dados coletados para a central receptora. Sensores auxiliares foram incorporados para captar parâmetros adicionais provenientes do veículo, enquanto os módulos de alimentação fornecem energia estável aos dispositivos, garantindo a continuidade das operações.

A Tabela 1 apresenta uma descrição detalhada de cada componente utilizado, contemplando informações como identificação, função e características técnicas relevantes. O conector OBD2, por sua vez, não está listado na tabela, pois se configura como elemento externo, utilizado exclusivamente para a interface entre o sistema de telemetria e a rede de comunicação CAN dos veículos, possibilitando a conexão direta com os barramentos internos do automóvel."

Tabela 1 - Demais Componentes.

Equipamento	SN65HVD230	GPS NEO-6M	Conversor Buck	ESP32
Descrição	Transceptor CAN para comunicação em rede CAN	Módulo GPS com antena cerâmica integrada	Conversor abaixador de tensão	Microcontrolador com Wi-Fi, Bluetooth e suporte a LoRa via SPI
Tensão de operação	3.3 V	3.3 V a 5 V	Entrada: 5 V a 30 V Saída: 5 V	3.3 V
Corrente máxima	370 μ A	50 mA	3 A	Tipicamente entre 160 mA a 260 mA, pico de até 500 mA
Interface de comunicação	CAN	UART (9600 bps, típico)	-	SPI, UART, I2C, CAN (via transceptor externo), Wi-Fi, Bluetooth
Temperatura operacional	-40 °C a +85 °C	-40 °C a +85 °C	-40 °C a +85 °C	-40 °C a +125 °C
Precisão	-	Até 2.5 metros	-	-
Outras características	Baixo consumo, adequado para 3.3 V	Inclui EEPROM para salvar dados de configuração	Vida útil: 30.000 h	Dual-core 240 MHz, até 520 kB RAM, 16 MB flash externa, Wi-Fi 802.11 b/g/n, Bluetooth BLE/Classic

Fonte: Adaptada de (INSTRUMENTS, 2002), (U-BLOX, 2011), (INSTRUMENTS, 2013), (ESPRESSIF, 2019).

Para a comunicação, optou-se pela utilização de módulos LoRa integrados ao microcontrolador ESP32, buscando facilitar a integração do sistema e acelerar o desenvolvimento. Nesse contexto, vale salientar que o termo “Lora32” é o nome dado

pela fabricante à plaquinha que reúne a ESP32 com o módulo LoRa. Assim sendo, foram estudadas duas placas comerciais amplamente utilizadas, apresentadas na Tabela 2: a Lora32 da TTGO, que utiliza o chip SX1276 da *Semtech*, e a Lora32 da *Heltec*, equipada com o chip SX1262. A escolha dessas placas deve-se à combinação do poder de processamento do ESP32 com a capacidade de comunicação LoRa, reunidas em um único módulo compacto. Apesar do SX1262 apresentar consumo energético reduzido, optou-se pela Lora32 da TTGO, com chip SX1276, devido à maior sensibilidade para recepção de sinais atribuída a este modelo, o que pode favorecer a estabilidade da comunicação em ambientes rurais e com obstáculos naturais.

Vale salientar que o PA (*Power Amplifier*) amplifica a potência do sinal interno, dentro do transmissor, antes dele chegar na antena. Já a potência máxima transmitida é a potência do sinal que sai pela antena para o ar, ou seja, o nível máximo de energia irradiada no ambiente.

Tabela 2 – Módulos LoRa.

Parâmetros	LORA32 HELTEC	LORA32 TTGO
Potência do transmissor	21 ±1 dBm	20 dBm
Sensibilidade	-134 dBm	-148 dBm
PA de alta eficiência	+14 dBm	+14 dBm
frequência	470~510 MHz, 863~928 MHz	470~510 MHz, 863~928 MHz

Fonte: Adaptada de (AUTOMATION, 2023), (LILYGO, 2025).

Além dos módulos LoRa, diversos outros componentes foram necessários para a montagem do sistema de telemetria veicular. Esses elementos, essenciais para o funcionamento integrado do projeto, estão listados na Tabela 2 – Módulos LoRa., que apresenta os detalhes de cada componente, excetuando-se o conector OBD2 que faz a interface entre o sistema apresentado e a comunicação veicular.

Em relação à comunicação em malha, optou-se por utilizar uma solução já consolidada e de código aberto, o protocolo Meshtastic (MESSINA, SANTORO e SANTORO, 2024). Essa escolha permitiu aproveitar uma infraestrutura robusta para o roteamento e retransmissão de mensagens entre os dispositivos LoRa, garantindo maior alcance e resiliência da rede. A principal contribuição do projeto foi entender o funcionamento da API, especificamente os pontos onde os dados são transmitidos e recebidos, para então implementar a lógica necessária para enviar, receber e publicar essas informações no protocolo MQTT. Dessa forma, foi possível integrar a rede em malha LoRa ao sistema de monitoramento centralizado de forma eficiente, sem a necessidade de desenvolver um protocolo de malha do zero.

Para o correto funcionamento do protocolo MQTT em dispositivos embarcados como o ESP32, alguns requisitos mínimos devem ser atendidos:

- Conectividade de rede: O dispositivo deve possuir interface Wi-Fi ou Ethernet para se conectar ao broker MQTT. No caso do ESP32, o módulo Wi-Fi integrado é suficiente para essa finalidade;
- Capacidade de processamento: O microcontrolador precisa ter capacidade para executar um cliente MQTT leve, como a biblioteca Eclipse “*Paho MQTT*” ou a “*PubSubClient*”, que são compatíveis com os recursos do ESP32 (dual-core de até 240 MHz);
- Memória RAM: É necessário que o dispositivo tenha pelo menos 100 kB de RAM (*Random Access Memory*) disponível para gerenciar buffers de mensagens, pilha de rede TCP/IP (*Transmission Control Protocol Internet Protocol*) e executar o cliente MQTT sem travamentos.

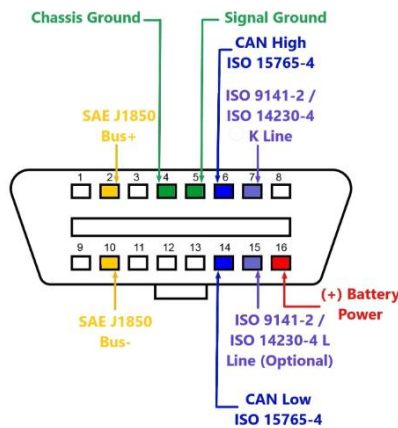
3.2 Ligações do sistema

O processo de montagem física do sistema iniciou-se com a prototipagem em protoboard, permitindo testes rápidos de conexão e funcionamento dos componentes. Nessa perspectiva, foram observados cuidados específicos para evitar interferências eletromagnéticas, como o uso de trilhas curtas para sinais sensíveis, a separação adequada entre linhas de alimentação e sinal, e o uso de capacitores de

desacoplamento próximos aos componentes críticos. Além disso, foi acrescentado um fusível para proteção elétrica visando a segurança dos módulos e do veículo.

A Figura 2 - Pinos Conector OBD2 apresenta a conexão com o conector OBD2 do veículo, responsável por fornecer os dados do barramento CAN e a alimentação elétrica de 12V.

Figura 2 - Pinos Conector OBD2.



Fonte: Retirado de (WEIS, 2024).

A Figura 3 mostra o conector físico do OBD2, enquanto a Tabela 3 destrincha cada pino. Vale salientar que ISO significa *International Organization for Standardization* e SAE, *Society of Automotive Engineers*.

Figura 3 – Conector OBD2 físico.



Fonte: Retirado de (ALIEXPRESS, 2025).

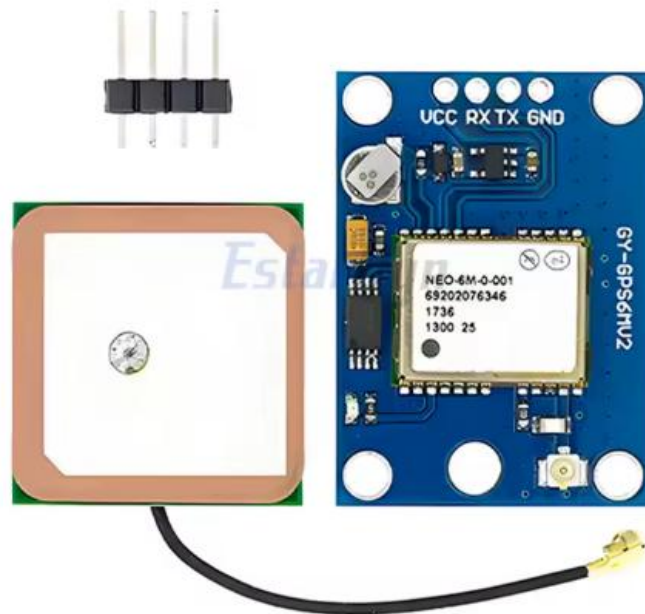
Tabela 3 - Detalhamento OBD2.

Pino	Função	Descrição	Utilizado?
1	Discricionário do Fabricante	Comunicação proprietária	Não
2	SAE J1850 Bus+	Protocolo de comunicação	Não
3	Discricionário do Fabricante	Comunicação proprietária	Não
4	Terra do Chassi	Terra do sistema automotivo	Sim
5	Terra do sinal	Terra do sistema automotivo	Sim
6	CAN High ISO 15765-4	Linha alta do barramento CAN	Sim
7	ISO 9141-2 / ISO 14230-4	Protocolo de comunicação	Não
8	Discricionário do Fabricante	Comunicação proprietária	Não
9	Discricionário do Fabricante	Comunicação proprietária	Não
10	SAE J1850 Bus	Complemento do barramento CAN	Não
11	Discricionário do Fabricante	Comunicação proprietária	Não
12	Discricionário do Fabricante	Comunicação proprietária	Não
13	Discricionário do Fabricante	Comunicação proprietária	Não
14	CAN Low ISO 15765-4	Linha baixa do barramento CAN	Sim
15	ISO 9141-2 / ISO 14230-4 Linha L (opcional)	Protocolo de comunicação	Não
16	Alimentação da Bateria do Veículo	Fonte de energia para o circuito do veículo	Sim

Fonte: Adaptada de (WEIS, 2024).

A Figura 4 apresenta o módulo GPS NEO-6M, utilizado para a coleta da posição geográfica em tempo real.

Figura 4 - Módulo GPS NEO-6M.



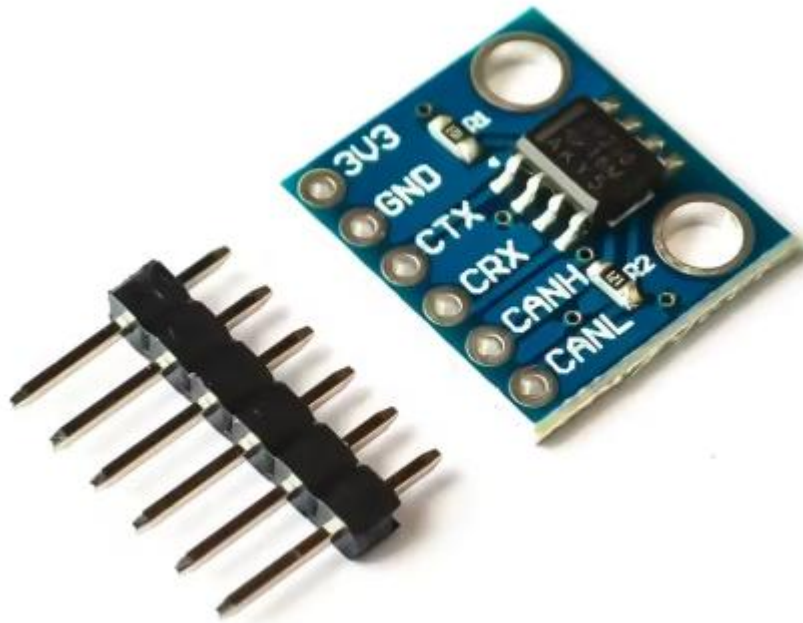
Fonte: Retirado de (ALIEXPRESS, 2025).

Os pinos do módulo GPS NEO-6M são:

- VCC (*Voltage at the Common Collector*) – Alimentação de 3,3V ou 5V, dependendo do modelo;
- GND (*Ground*) – Terra;
- TX (*Transmitter*) – Transmite os dados (saída do GPS);
- RX (*Receiver*) – Recebe dados (entrada, raramente usada).

A comunicação com o sistema do veículo é feita por meio do transceptor CAN SN65HVD230, como ilustrado na Figura 5 (cujos pinos estão descritos logo abaixo dela), que converte os sinais do barramento CAN para níveis compatíveis com o microcontrolador.

Figura 5 - Módulo Transceptor CAN SN65HVD230.



Fonte: Retirado de (ALIEXPRESS, 2025).

- VCC – Alimentação (tipicamente 3,3V);
- GND – Terra;
- CTX – Entrada de dados do microcontrolador (transmissão CAN);
- CRX – Saída de dados para o microcontrolador (recepção CAN);
- CANH – Linha CAN *High* (conectada ao pino 6 do OBD2);
- CANL – Linha CAN *Low* (conectada ao pino 14 do OBD2).

Já a Figura 6 detalha o mapeamento dos pinos da placa LoRa32 da TTGO, permitindo a correta integração com os módulos CAN, GPS e demais periféricos.

De forma mais precisa, são utilizados quatro pinos do conector OBD2: dois correspondem à alimentação (12 V e GND), e os outros dois ao barramento de dados (CAN_H e CAN_L). A alimentação de 12 V proveniente do veículo é reduzida para 5 V por meio de um conversor *buck*, sendo essa a tensão utilizada para alimentar os principais módulos do sistema. Os sinais de comunicação CAN_H e CAN_L são conectados ao transceptor CAN (SN65HVD230), que atua como interface entre o barramento do veículo e o microcontrolador ESP32. Importante destacar que esse transceptor é alimentado com 3,3 V fornecidos diretamente pela ESP32, garantindo compatibilidade lógica entre os dispositivos. O módulo GPS NEO-6M, por sua vez, é alimentado com 5 V provenientes do mesmo conversor abaixador, e a comunicação com a ESP32 é realizada via interface serial UART, sendo necessário o cruzamento dos pinos de transmissão e recepção (TX do GPS no RX da ESP32, e vice-versa) para assegurar a troca correta de dados.

Do outro lado, no receptor, não é necessária nenhuma conexão adicional além da alimentação elétrica, pois ele atua apenas como ponto final de recepção dos dados transmitidos pela rede LoRa.

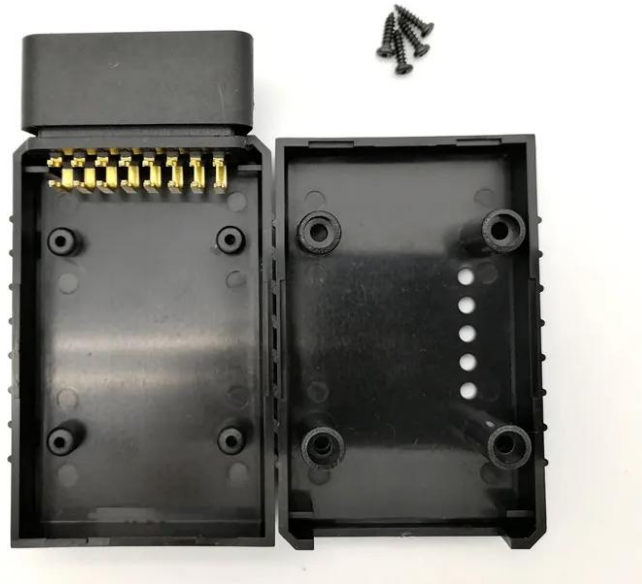
Por fim, para confecção do suporte que ficará todo circuito, foram feitas 2 versões:

1º versão: foi utilizado protoboard e jumpers para realizar as conexões entre os módulos e componentes. Essa abordagem facilitou a montagem inicial e os testes rápidos, porém apresentava limitações quanto à firmeza das conexões e à estabilidade do circuito durante o manuseio;

2º versão: foi adotada uma placa universal de circuito impresso (PCB), proporcionando maior fixação e estabilidade das ligações. A montagem foi realizada com solda COBIX 63/37, uma liga de estanho (63%) e chumbo (37%) com ponto de fusão de 183 °C, ideal para aplicações eletrônicas. Essa solda garante excelente aderência, transição direta entre sólido e líquido, e acabamento uniforme, resultando em conexões mais confiáveis e duráveis.

Vale salientar que, quanto ao invólucro físico do circuito, pensou-se na utilização de uma caixa plástica padrão, já pronta, capaz de acomodar toda a estrutura interna com segurança e proteção. A Figura 8 ilustra a ideia do compartimento onde o circuito deverá ser instalado futuramente.

Figura 8 – Invólucro físico do circuito.



Fonte: Retirado de (ALIEXPRESS, 2025).

A escolha do invólucro definitivo deve atender não apenas à proteção física do circuito, mas também às condições adversas do ambiente onde será instalado. Para garantir durabilidade e funcionalidade, três fatores críticos devem ser analisados: temperatura, vibração e segurança contra acesso não autorizado.

1. Temperatura

- Utilizar materiais termorresistentes (polycarbonato/ABS) em ambientes quentes;
- Adotar vedação hermética e sílica gel em locais úmidos.

2. Vibração

- Incluir amortecedores internos (espuma/borracha);
- Usar fixação reforçada (parafusos travantes, braçadeiras);
- Preferir estrutura rígida com reforços, se necessário.

3. Segurança

- Travas ou fechaduras para impedir acesso fácil;
- Instalação em locais elevados ou painéis trancados.

4 DESENVOLVIMENTO DO TRABALHO

Este capítulo apresenta a implementação prática do sistema de telemetria veicular proposto, com base na arquitetura e componentes descritos anteriormente. São detalhadas as configurações dos módulos utilizados, os testes de funcionamento em bancada, bem como os ajustes realizados para garantir a integração entre os dispositivos CAN, GPS e LoRa. Além disso, são apresentados os procedimentos adotados para preparar o sistema para o envio dos dados ao servidor MQTT, validando a comunicação ponta a ponta.

Diferentemente da metodologia, que tratou o sistema de forma geral e descritiva, o desenvolvimento será abordado em quatro partes, estruturadas de acordo com os quatro objetivos específicos definidos neste trabalho. Essa divisão tem como finalidade apresentar, de forma clara e sequencial, as etapas que compõem a construção e validação do sistema.

4.1 Comunicação CAN ao usar o microcontrolador

A princípio, foi realizado um estudo para identificar qual tipo de protocolo CAN estava presente em cada categoria de veículo. Constatou-se que em veículos leves — como carros de passeio, caminhonetes e similares — é geralmente utilizado o protocolo ISO 15765-4 (também conhecido como OBD-II sobre CAN). Já em veículos pesados — como caminhões, ônibus e tratores — é comum a presença do SAE J1939, um protocolo padronizado pela SAE, amplamente utilizado para a comunicação entre módulos eletrônicos em sistemas embarcados de grande porte.

Diante dessa diversidade, optou-se por focar o desenvolvimento em veículos leves, considerando a maior compatibilidade com os módulos disponíveis e a padronização do acesso via OBD-II. Como citado na fundamentação teórica, o OBD-II já define comandos padronizados, os chamados PIDs (*Parameter IDs*), que permitem a coleta de dados de sensores de forma uniforme e aberta. Em contrapartida, embora o SAE J1939 tenha alguns comandos padronizados, na maioria dos casos trata-se de um protocolo proprietário, ou seja, os códigos são definidos pelo

próprio fabricante do veículo ou do sistema, e somente quem tem acesso ao banco de dados desses fabricantes consegue interpretá-los corretamente.

No entanto, mesmo com o foco em veículos leves, não foi possível estabelecer comunicação de imediato, pois foi identificado que as ECUs utilizam comumente dois tipos distintos de identificadores CAN (IDs): um curto, como o 0x7E8, e um estendido, como o 0x18DB33F1, exigindo ajustes na configuração do sistema para reconhecer e tratar ambos os formatos de ID.

Além disso, para possibilitar a comunicação entre o microcontrolador e o transceptor CAN, existem diversas bibliotecas que implementam a interface necessária para interpretar os dados do barramento. Foi escolhida uma biblioteca que, ao enviar uma solicitação OBD-II, já retorna os dados em formato chave-valor — informando claramente qual PID foi respondido e qual valor decodificado foi recebido — simplificando o processo de leitura e análise dos dados e tornando o desenvolvimento mais eficiente.

Dito isso, com os ajustes realizados e utilizando a segunda versão do protótipo, que incorporava a placa universal de circuito impresso e a soldagem definitiva dos componentes, foi possível estabelecer uma conexão estável com a ECU do veículo.

4.2 Comunicação entre 2 microcontroladores via LoRa

Nesta etapa, foi implementada a comunicação entre dois microcontroladores utilizando o protocolo LoRa, com foco inicial na troca de dados ponto a ponto. Cada dispositivo foi equipado com um módulo de rádio LoRa configurado com os mesmos parâmetros de frequência, espalhamento e potência, garantindo compatibilidade e estabilidade na transmissão.

Foram enviados pacotes contendo mensagens de texto e dados simulados, permitindo validar a integridade da informação, o tempo de resposta, o alcance em ambiente controlado e a estabilidade da comunicação. Mecanismos de controle foram empregados para assegurar o recebimento dos pacotes, evitando perdas ou colisões, enquanto a interface serial dos microcontroladores auxiliava no monitoramento em tempo real e na identificação de eventuais falhas.

Após essa fase inicial bem-sucedida, o projeto avançou para o estudo de uma rede em malha (*mesh*) com o objetivo de ampliar o alcance e a robustez da comunicação. O protocolo LoRaLayer2 foi testado, porém apresentou problemas como colisões frequentes, falta de confirmação de recebimento (ACK) e tempos elevados de envio, impactando negativamente a eficiência do sistema.

Diante disso, adotou-se o protocolo Meshtastic, que foi implementado com sucesso, proporcionando melhorias significativas na confiabilidade da rede, redução de colisões, confirmação de entrega e otimização do tempo de resposta entre os nós. Essa evolução validou a viabilidade do uso de redes LoRa *mesh* para a aplicação proposta. Ao final dos testes, foi possível comprovar a eficácia da comunicação entre os dispositivos, com envio e recepção bem-sucedidos de dados de uma ESP32 para outra, consolidando a tecnologia LoRa como um meio confiável de transmissão sem fio dentro do sistema desenvolvido.

4.3 Envio de dados CAN do transmissor ao receptor via LoRa

Com a comunicação via *LoRa mesh* funcionando adequadamente e a coleta de dados via protocolo CAN já estabelecida, foi iniciada a integração entre esses dois sistemas. O objetivo desta etapa foi permitir que os dados adquiridos da ECU do veículo por meio do protocolo OBD-II fossem transmitidos, via LoRa, de um microcontrolador (transmissor) para outro (receptor), que por sua vez ficaria responsável por publicar as informações em um servidor MQTT.

Entretanto, durante a implementação, observou-se uma incompatibilidade inicial: o protocolo Meshtastic, utilizado para a comunicação em malha, já definia internamente os pinos e estruturas de controle para cada microcontrolador. Isso impôs restrições ao incorporar a funcionalidade de leitura de dados CAN simultaneamente à execução da pilha de comunicação *mesh*.

Ao investigar a fundo o funcionamento das bibliotecas envolvidas, identificou-se que o problema estava relacionado à falta de sincronização adequada entre as tarefas (threads) do sistema, resultando em falhas intermitentes na coleta ou no envio dos dados. Enquanto o Meshtastic operava de forma robusta com sua própria lógica de agendamento, a biblioteca utilizada para leitura CAN não implementava corretamente

o controle de bloqueio e retomada de tarefas, o que gerava conflitos de acesso a recursos compartilhados.

Para resolver esse impasse, foi realizado um estudo aprofundado da biblioteca de acesso ao barramento CAN, e como solução foi incorporada a função **ulTaskNotifyTake()** da FreeRTOS, sistema operacional em tempo real utilizado nos microcontroladores ESP32. Essa função permite que uma tarefa entre em estado de bloqueio até que receba uma notificação de outra tarefa, proporcionando uma forma eficiente de sincronização entre processos assíncronos sem consumir ciclos de CPU desnecessariamente. Com isso, a tarefa responsável pela leitura dos dados CAN, mostrada na Figura 9, passou a aguardar corretamente até que o processo de envio via LoRa estivesse finalizado, evitando sobreposição ou perda de dados.

Figura 9 - Código para sincronização da tarefa CAN.

```
void ESP32CAN::task_CAN( void *pvParameters )
{
    ESP32CAN* espCan = (ESP32CAN*)pvParameters;
    CAN_FRAME rxFrame;

    while (1) {
        // Espera por uma notificação ou por 100ms (timeout)
        uint32_t notificationValue = ulTaskNotifyTake(pdTRUE, pdMS_TO_TICKS(100));

        if (notificationValue > 0) {
            // Processa todos os frames pendentes
            while (uxQueueMessagesWaiting(espCan->callbackQueue)) {
                if (xQueueReceive(espCan->callbackQueue, &rxFrame, portMAX_DELAY) == pdTRUE) {
                    espCan->sendCallback(&rxFrame);
                }
            }
        }

        #if defined(CONFIG_FREERTOS_UNICORE)
            delay(10);
        #endif
    }

    vTaskDelete(NULL);
}
```

Fonte: Autor, 2025.

Após essa correção, a integração foi bem-sucedida: os dados lidos da ECU foram transmitidos com êxito do transmissor para o receptor via LoRa *mesh* e, em

seguida, publicados no broker MQTT, completando o fluxo ponta a ponta da coleta e encaminhamento das informações veiculares.

4.4 Confeção da placa de circuito impresso

Para a montagem do circuito físico, foram adotadas diferentes abordagens ao longo das versões do protótipo. Na versão inicial, utilizou-se protoboard e jumpers para as interligações dos componentes, sendo esses materiais de uso pessoal, o que possibilitou testes rápidos e reconfigurações durante o processo de desenvolvimento.

Na segunda versão, visando maior estabilidade e organização do circuito, foi utilizada uma placa universal de circuito impresso (PCB) fornecida pela empresa, assim como os componentes eletrônicos, enquanto o restante da estrutura ainda foi montado com materiais disponíveis pessoalmente. A soldagem foi realizada com ferro de solda de 60 W, utilizando solda COBIX 63/37 e, para um melhor acabamento estético, foram aproveitados fios de cabos de rede antigos, que estavam à disposição, conferindo mais rigidez e limpeza visual ao circuito.

Com o amadurecimento do projeto, partiu-se para uma terceira (e possivelmente última) versão, desenvolvida com maior precisão e foco em aplicação prática. Nessa etapa, a placa foi desenhada digitalmente no *software Fusion 360*, uma plataforma amplamente utilizada para o projeto de circuitos eletrônicos e impressão 3D. A utilização desse recurso permitiu criar um *layout* de placa otimizado, organizado e pronto para fabricação, elevando a qualidade final do sistema. Nesta versão, buscou-se manter os mesmos itens utilizados anteriormente, porém com o uso de componentes menores e mais compactos, a fim de reduzir o tamanho físico do circuito e facilitar sua integração em aplicações reais.

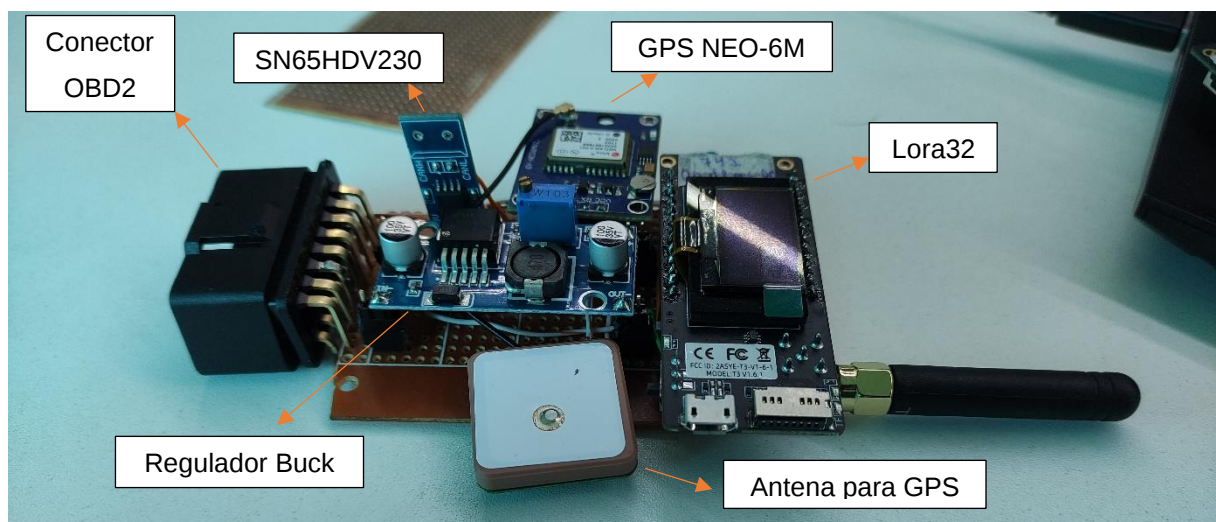
5 RESULTADOS

Esta seção apresenta os resultados obtidos a partir da implementação prática do sistema proposto. Os testes foram conduzidos com base nas etapas descritas anteriormente, considerando tanto a comunicação entre os módulos quanto o desempenho na coleta e transmissão de dados. Para melhor organização e clareza, os resultados serão apresentados de forma dividida conforme os objetivos específicos estabelecidos no início deste trabalho, possibilitando uma análise direta do cumprimento de cada meta traçada.

5.1 Confeção da placa de circuito impresso

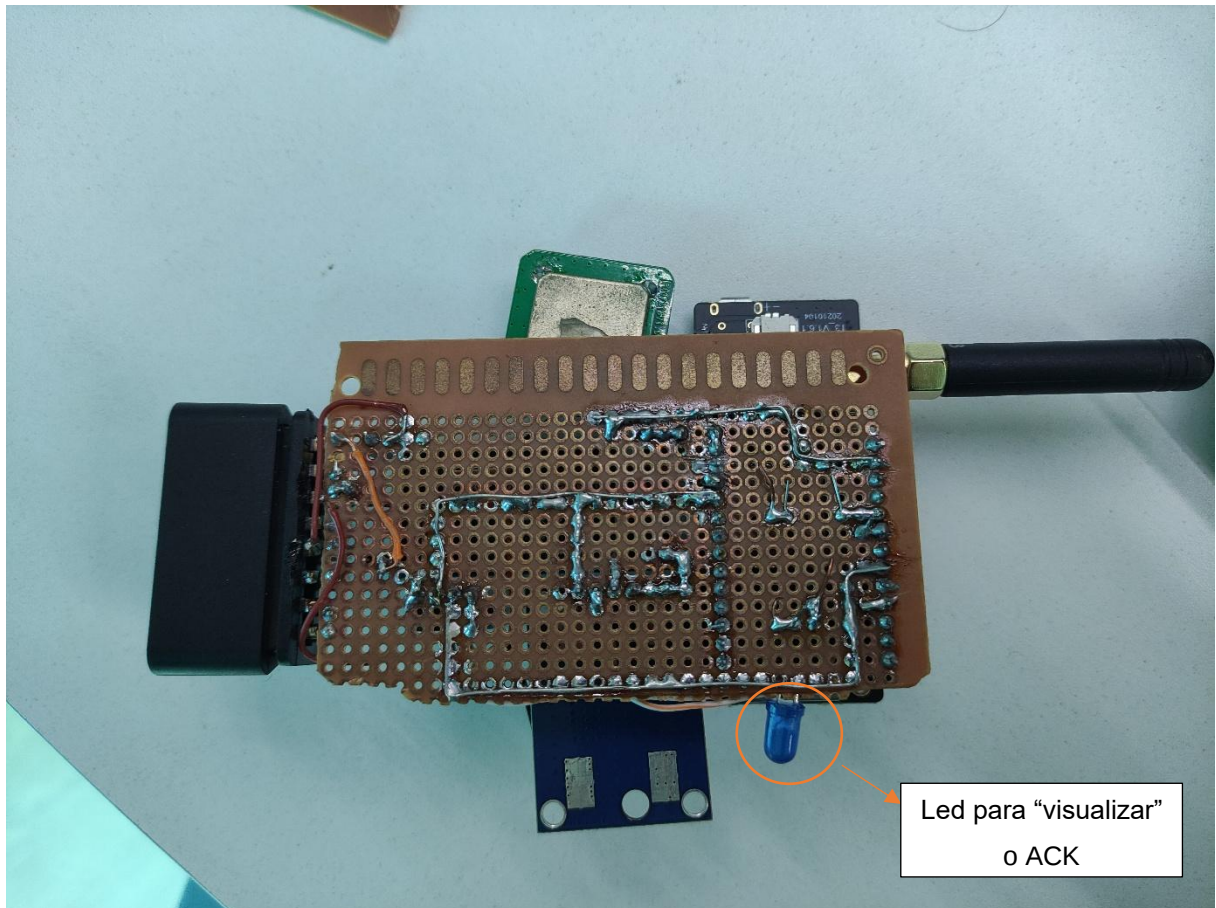
A Figura 10 e a Figura 11 mostram a PCB projetada e fixada no carro para a coleta e envio de dados.

Figura 10 - Parte superior PCB



Fonte: Autor, 2025.

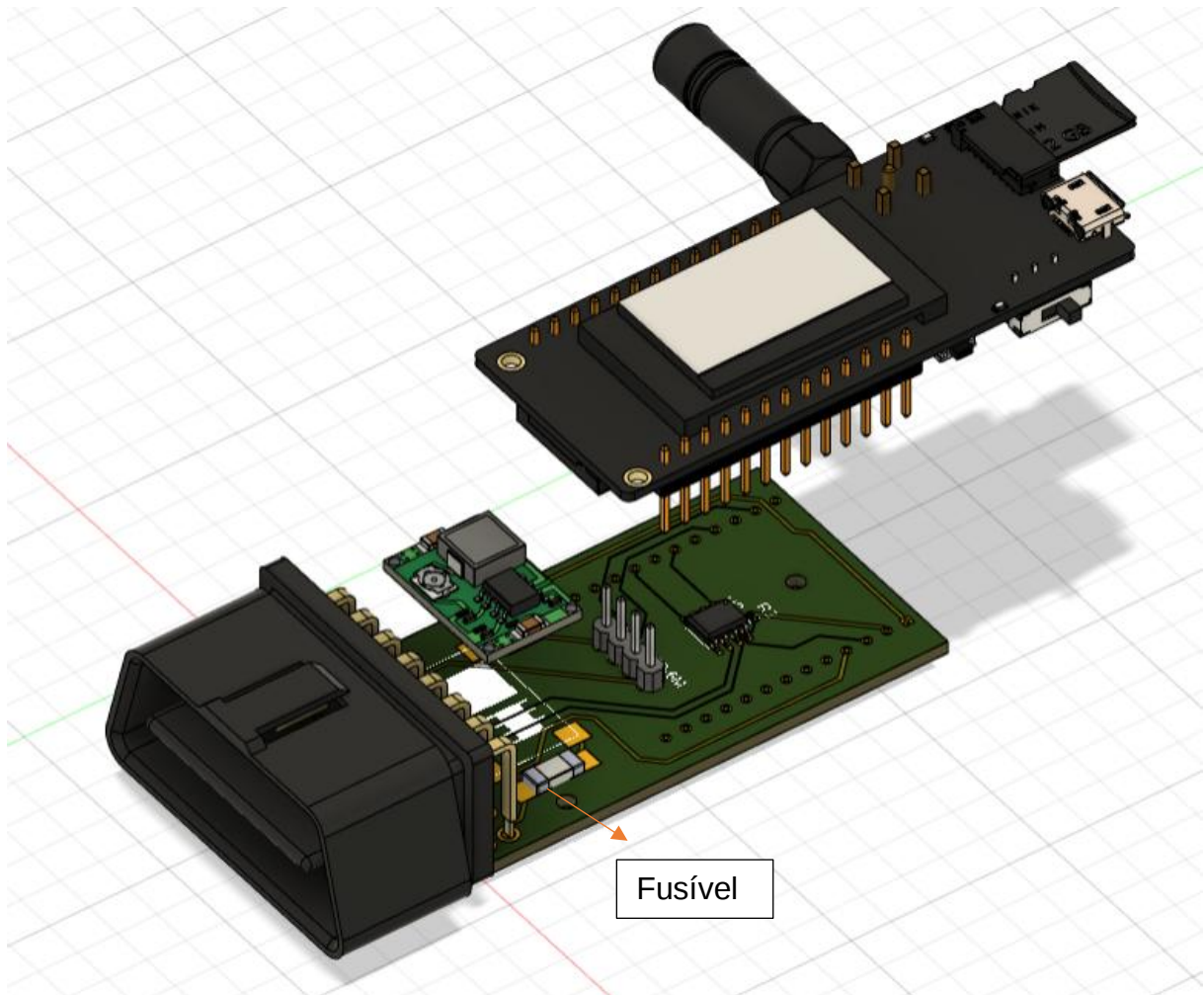
Figura 11 - Parte Inferior PCB.



Fonte: Autor, 2025.

Após a conclusão do protótipo, desenvolveu-se uma versão idealizada da placa, ilustrada na Figura 12. Essa versão foi projetada com foco na menor dimensão física possível, aproveitando ambas as faces (superior e inferior) para a fixação dos componentes. Seu objetivo é apenas ilustrar a forma como se planejava finalizar o projeto.

Figura 12 - PCB idealizada.



Fonte: Autor, 2025.

5.2 Comunicação entre 2 microcontroladores via LoRa

A Figura 13 mostra o envio das mensagens coletadas via LoRa. À esquerda, o terminal do transmissor exibe os dados enviados; à direita, o terminal do receptor mostra a mensagem recebida, confirmando a transmissão correta entre os dispositivos. A Figura 14 apresenta a recepção do bit de confirmação (ACK) por parte do transmissor, indicando que o receptor recebeu a mensagem com sucesso.

Figura 13 - Envio e recepção via LoRa.

```

--50 hop
DEBUG | 13:02:27 178 [Router] Update DB node 0x6ce13e0, rx_time=0
INFO | 13:02:27 178 [Router] Heard new node on ch. 20, send NodeInfo and ask for response
DEBUG | 13:02:27 178 [Router] cancelSending id=0x65cc0202, removed=0
DEBUG | 13:02:27 178 [Router] Skip send NodeInfo since we sent it <5min ago
DEBUG | 13:02:27 178 [Router] Forwarding to phone (id=0xf8e5c100 fr=0x06ce13e0 to=0x2f934fec, WantAck=0, HopLim=2 Ch=0x14 Portnum=5 requestId=4953aa01 Payload: # rxSNR=12.5 rxRSSI=-50 hops
WARN | 13:02:27 178 [Router] ToPhone queue is full, drop packet
DEBUG | 13:02:27 178 [Router] Module 'routing' considered
DEBUG | 13:02:27 186 Partially randomized packet id 1707868674
INFO | 13:02:27 186 Send message id=1707868674, dest=6ce13e0, msg=Taisa Repeater
DEBUG | 13:02:27 187 Update DB node 0x2f934fec, rx_time=0
DEBUG | 13:02:27 187 Set next retransmission in 8088 msecs:
DEBUG | 13:02:27 187 (id=0x65cc0202 fr=0x2f934fec to=0x06ce13e0, WantAck=1, HopLim=3 Ch=0x14 Portnum=1 Payload: Taisa Repeater)
DEBUG | 13:02:27 187 Add packet record (id=0x65cc0202 fr=0x2f934fec to=0x06ce13e0, WantAck=1, HopLim=3 Ch=0x14 Portnum=1 Payload: Taisa Repeater)

R=8.75
DEBUG | 13:02:27 66 [Router] Module 'text' wantsPacket=1
INFO | 13:02:27 66 [Router] Received text msg from=0x2f934fec, id=0x65cc0202, RSSI=-51, SNR=8.750000, msg=Taisa Repeater
INFO | 13:02:27 66 [Router] Primeiro PUB
INFO | 13:02:27 66 [Router] Publicando...
INFO | 13:02:27 66 [Router] Publicando mensagem com ###tricas RF...
DEBUG | 13:02:27 66 [Router] Module 'text' considered
DEBUG | 13:02:27 66 [Router] Module 'StoreForward' wantsPacket=1
DEBUG | 13:02:27 66 [Router] Module 'StoreForward' considered
DEBUG | 13:02:27 66 [Router] Module 'ExternalNotificationModule' wantsPacket=1
INFO | 13:02:27 66 [Router] External Notification Module Disabled or muted
DEBUG | 13:02:27 66 [Router] Module 'ExternalNotificationModule' considered
DEBUG | 13:02:27 66 [Router] Module 'routing' wantsPacket=1
INFO | 13:02:27 66 [Router] Received routing from=0x2f934fec, id=0x65cc0202, portnum=1, payloadlen=14
DEBUG | 13:02:27 66 [Router] Routing sniffing (id=0x65cc0202 fr=0x2f934fec to=0x06ce13e0,

```

Fonte: Autor, 2025.

Figura 14 - Confirmação de recepção (ACK).

```

DEBUG | 13:02:27 188 [Router] Show 0 module frames
DEBUG | 13:02:27 188 [Router] Total frame count: 103
DEBUG | 13:02:27 188 [Router] Added modules. numframes: 0
DEBUG | 13:02:27 188 [Router] Finished build frames. numframes: 3
DEBUG | 13:02:27 188 [Router] Module 'canned' considered
DEBUG | 13:02:27 188 [Router] Module 'routing' wantsPacket=1
INFO | 13:02:27 188 [Router] Received routing from=0x6ce13e0, id=0xe56d3101, portnum=5, payloadlen=2
DEBUG | 13:02:27 188 [Router] Routing sniffing (id=0xe56d3101 fr=0x06ce13e0 to=0x2f934fec, WantAck=0, HopLim=2 Ch=0x14 Portnum=5 requestId=65cc0202 Payload: # rxSNR=12.25 rxRSSI=-50 hopSta
DEBUG | 13:02:27 188 [Router] Received a ACK for 0x65cc0202, stopping retransmissions
ACK recebido para o pacote com ID: 0x65CC0202
DEBUG | 13:02:27 188 [Router] Delivering rx packet (id=0xe56d3101 fr=0x06ce13e0 to=0x2f934fec, WantAck=0, HopLim=2 Ch=0x14 Portnum=5 requestId=65cc0202 Payload: # rxSNR=12.25 rxRSSI=-50 ho
DEBUG | 13:02:27 189 [Router] Update DB node 0x6ce13e0, rx_time=0

```

Fonte: Autor, 2025.

A Figura 15 apresenta o ambiente no qual foi realizada a validação da rede em malha, utilizando três nós: um receptor (R) e dois transmissores (T1 e T2). Para fins de teste, cada dispositivo foi programado para enviar periodicamente uma mensagem de “vizinhança” a cada 30 segundos, com o objetivo de construir a tabela de vizinhos apresentada na Figura 16. Essa figura exibe o endereço MAC (*Media Access Control*) — identificador único de cada dispositivo — e o valor do SNR — relação entre sinal e ruído, utilizada para avaliar a qualidade da comunicação — da última mensagem de “vizinhança”. Ressalta-se que o node 0 representa o receptor, enquanto o node 1 corresponde ao transmissor 1, alimentado por bateria.

Figura 15 - Ambiente para verificação dos vizinhos.

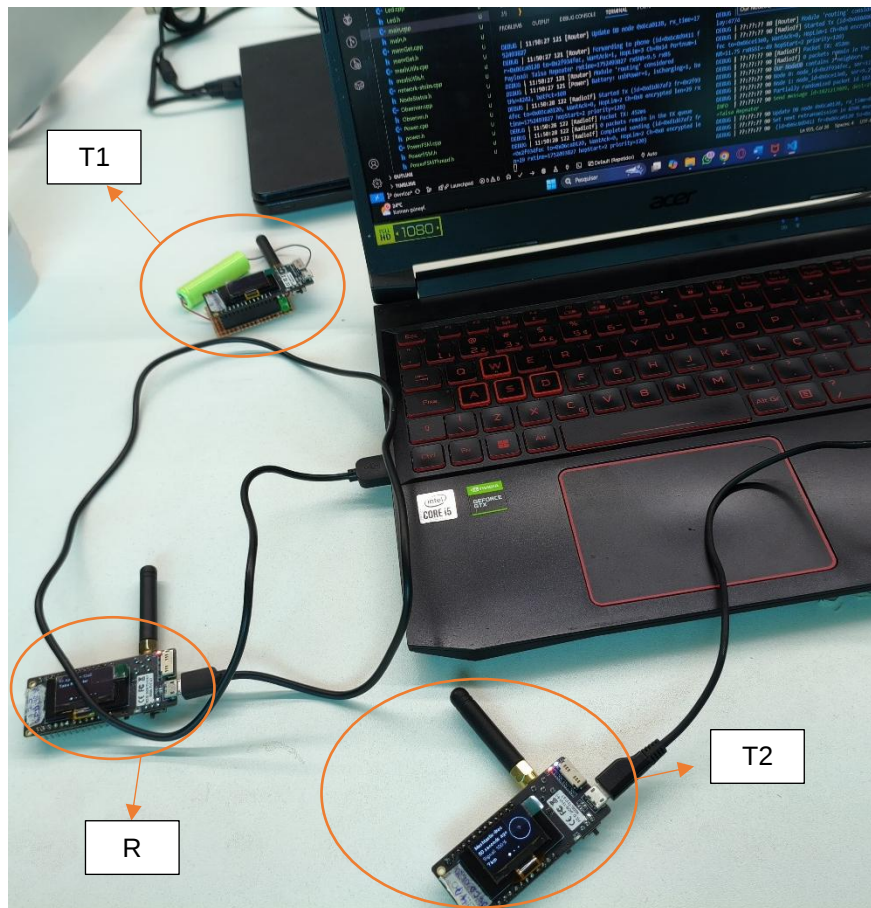


Figura 16 - Verificação dos vizinhos.

```
DEBUG | ????:?? 90 Our NodeDB contains 2 neighbors
DEBUG | ????:?? 90 Node 0: node_id=0x2f934fec, snr=11.75
DEBUG | ????:?? 90 Node 1: node_id=0x6ce13e0, snr=9.25
```

Fonte: Autor, 2025.

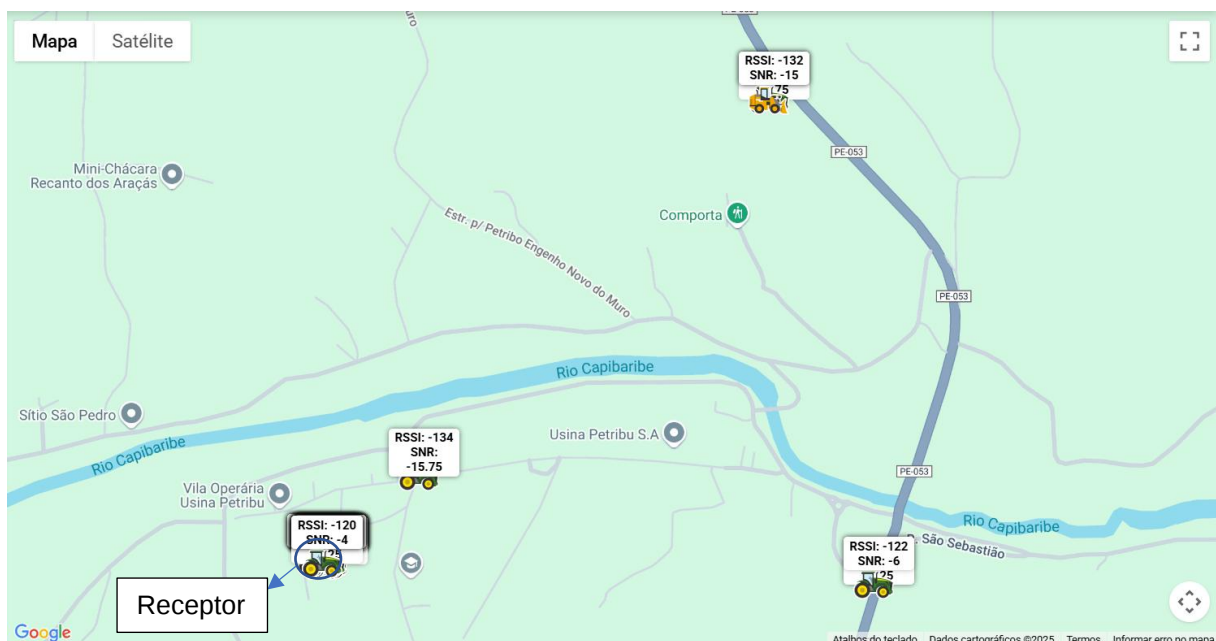
5.3 Envio de dados CAN do transmissor ao receptor via LoRa

A Figura 17 apresenta o resultado dos testes realizados com o veículo em movimento, enquanto uma antena receptora permaneceu fixa recebendo os dados via LoRa. Para fins de visualização, no software utilizado com a API do *Google Maps* e implementação em *Python* e tecnologias *web*, foram adotados ícones aleatórios durante esta fase de testes para representar as posições registradas do veículo Fiat

Fiorino 2022. Ressalta-se que a localização da antena receptora está próxima e claramente indicada no canto inferior da figura correspondente.

Para armazenamento dos dados, foi utilizado o broker *Mosquitto* para o protocolo MQTT, com os dados sendo armazenados em um banco de dados Oracle, garantindo confiabilidade e segurança no gerenciamento das informações coletadas.

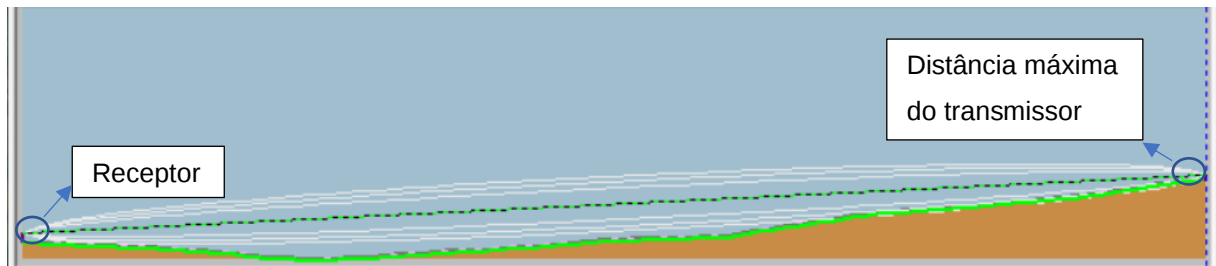
Figura 17 - Transmissão ponto a ponto.



Fonte: Autor, 2025.

A Figura 18 apresenta o ganho de relevo no ponto mais distante alcançado durante os testes (à direita), com elevação de aproximadamente 125 metros em relação ao nível do mar, bem como a posição do receptor (à esquerda), localizado a cerca de 95 metros de altitude. A imagem permite visualizar a variação topográfica entre os dois pontos.

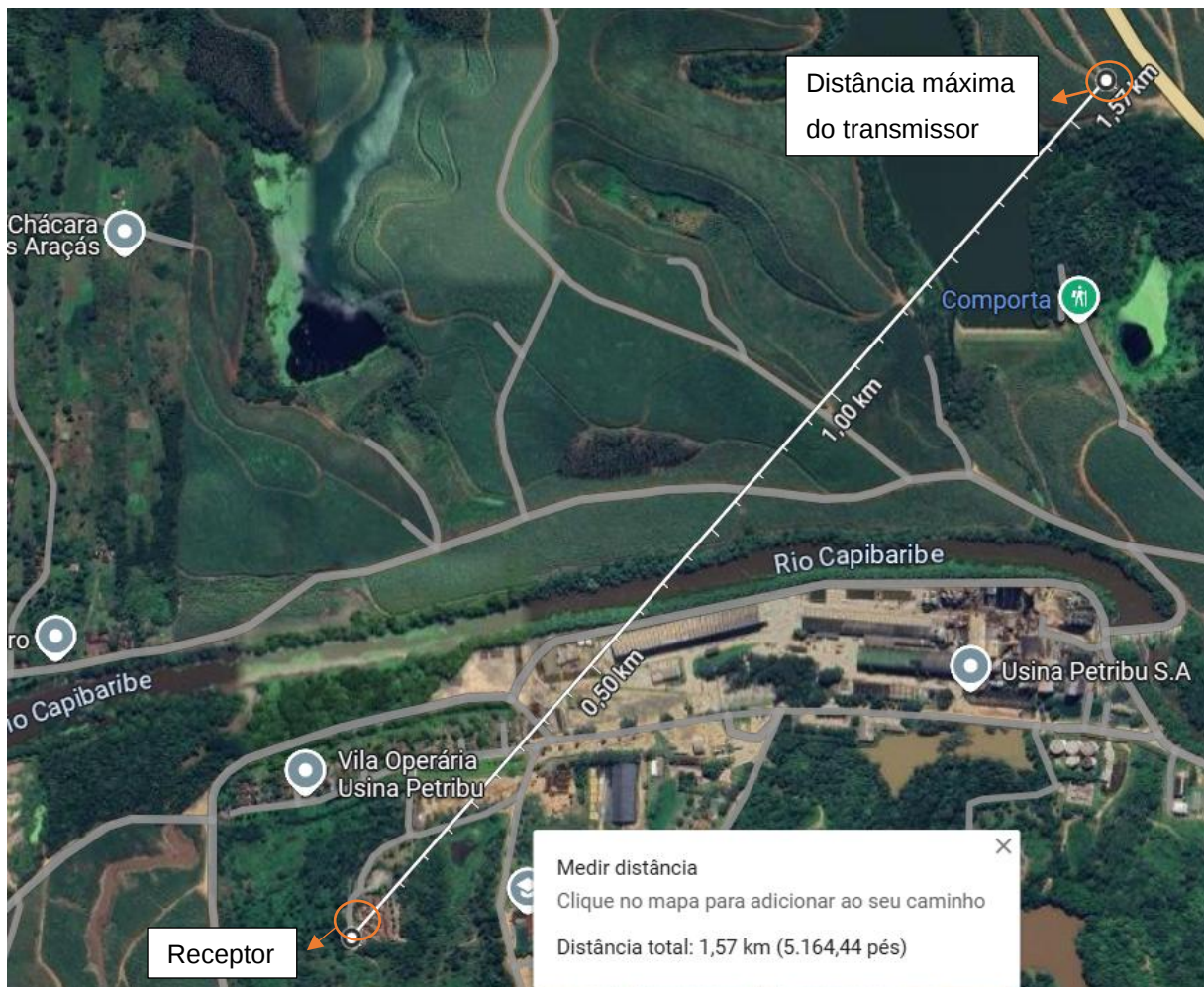
Figura 18 - Ganho de relevo da maior distância.



Fonte: Autor, 2025.

A Figura 19 exibe a distância estimada entre o transmissor e o receptor, calculada com base nas coordenadas geográficas por meio do Google Maps.

Figura 19 - Maior distância alcançada entre veículo e central de monitoramento.



Fonte: Autor, 2025.

5.4 Comunicação CAN ao usar o microcontrolador

Nesta subseção, são apresentadas imagens que ilustram o ambiente planejado para a coleta de dados CAN em um veículo Fiat Uno Fiorino 2022. O objetivo dessas imagens é evidenciar a estrutura prática desenvolvida para captação dos dados, conforme o proposto nas etapas anteriores do projeto.

A Figura 20 e Figura 21 mostram o início da montagem do ambiente de coleta, com os dispositivos sendo preparados para integração com o sistema do veículo e evidenciando a conexão física entre dispositivo e automóvel.

Figura 20 - Coleta de dados CAN.



Fonte: Autor, 2025.

Figura 21 - Coleta de dados CAN, 2.



Fonte: Autor, 2025.

A Figura 22 exibe o zoom correspondente às 2 figuras anteriores.

Figura 22 - Zoom da conexão.



Fonte: Autor, 2025.

A Figura 23 mostra um exemplo de mensagem coletada diretamente da rede CAN, demonstrando a estrutura do dado recebido. Essa mensagem foi construída no formato "identificador/chave valor/chave valor/chave valor/...", pois, devido à limitação de caracteres disponíveis, utilizar JSON (*JavaScript Object Notation*) acabaria ocupando muito espaço por conta das aspas, espaços e outros símbolos que, nesse contexto, seriam considerados “inúteis” como informação. Dessa forma, optou-se por projetar um modelo de mensagem personalizado, mais enxuto e eficiente para o propósito da aplicação.

Figura 23 - Mensagem coletada 1.

```
DEBUG | 13:11:28 54 Use AES128 key!
DEBUG | 13:11:28 54 decoded message (id=0x34a628c0 fr=0x06d8a6f0 to=0x06ce13e0, WantAck=1, HopLim=3 Ch=0x14 Portnum=1 Payload: 742/9
7 0/98 0/99 0/1 352512 /3 0 /4 0.00/5 24.00/6
Mensagem final enviada: 742/97 0/98 0/99 0/1 352512 /3 0 /4 0.00/5 24.00/6 0.00/7 0.78/11 100.00/12 0.00/13 0.00/14 0.00/15 26.00/17
19.22/19 3 /20 0.00/21 0.00/28 35 /33 0.00/47 66.67/51 100.00/66 12.00/73 14.90/74 14.12
Mensagem, loop principal, enviada para destino 0x6CE13E0 no canal 20
DEBUG | 13:11:28 54 [RadioIf] completed sending (id=0xd2b410b9 fr=0x06d8a6f0 to=0x06ce13e0, WantAck=1, HopLim=3 Ch=0x8 encrypted len
=220 rxtime=1751980257 hopStart=3 priority=100)
```

Fonte: Autor, 2025.

A Figura 24, por sua vez, apresenta uma mensagem cujo conteúdo foi segmentado em duas partes (2 transmissões feitas), já planejado para caso o futuro envio LoRa exceda 230 caracteres por pacote.

Figura 24 - Mensagem coletada 2.

```
DEBUG | 13:13:55 75 decoded message (id=0x60df657b fr=0x06d8a6f0 to=0x06ce13e0, WantAck=1, HopLim=3 Ch=0x14 Portnum=1 Payload: 742/1
3 0.00/14 0.00/15 26.00/17 19.22/19 3 /20 0.00
Mensagem dividida e enviada em partes:
742/97 -7.890142/98 -35.251106/99 1751980429/1 352512 /3 0 /4 0.00/5 24.00/6 0.00/7 0.78/11 100.00/12 0.00
742/13 0.00/14 0.00/15 26.00/17 19.22/19 3 /20 0.00/21 0.00/28 35 /33 0.00/47 66.67/51 100.00/66 12.00/73 14.51/74 14.12
Mensagem, loop principal, enviada para destino 0x6CE13E0 no canal 20
```

Fonte: Autor, 2025.

A formatação usada é do tipo chave e valor sequencial, em que o primeiro valor indica o identificador da frota e os demais pares seguem como “/chave valor/” separados por espaço e separando cada chave (sensor) por barra, como em 742/97 0/98 0/99. Esse formato economiza muito mais espaço que JSON, essencial no LoRa, permitindo transmitir dados de forma mais rápida e eficiente.

6 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Após a elaboração dos planos, o desenvolvimento do sistema e a análise dos resultados, verificou-se que a proposta de telemetria veicular baseada na tecnologia LoRa não apenas se mostrou viável, mas também apresentou diferenciais significativos em relação às soluções tradicionais. A arquitetura concebida demonstrou elevada eficiência na transmissão de dados a longa distância, mantendo baixo consumo energético e simplicidade de implementação — fatores determinantes para aplicações embarcadas e em contextos com restrições de infraestrutura.

Os experimentos de campo confirmaram a estabilidade da comunicação mesmo diante de obstáculos físicos moderados, além de evidenciarem a reduzida demanda energética dos dispositivos, aspecto que viabiliza sua utilização em operações autônomas de longo prazo. A precisão dos dados coletados reforça a aplicabilidade prática da solução, validando seu uso em cenários reais de monitoramento veicular e demonstrando que é possível conciliar eficiência técnica e viabilidade econômica em uma mesma proposta.

Por outro lado, foram observadas limitações relacionadas a efeitos de somreamento, que ocasionalmente comprometeram a qualidade do sinal entre determinados nós. Essas limitações, embora relevantes, não inviabilizam a aplicação prática da solução, mas reforçam a necessidade de estratégias complementares de mitigação, como o posicionamento adequado dos dispositivos, a calibração dos parâmetros de comunicação e o uso de topologias mais sofisticadas. Ainda assim, o trabalho se destaca por propor uma abordagem inicial de supervisão e por indicar a necessidade de um sistema definitivo de visualização e análise em tempo real, requisito essencial para consolidar a automação da plataforma em escala operacional e atender a demandas de alta confiabilidade.

Entre as contribuições originais, ressalta-se a exploração do potencial da topologia em malha aplicada à telemetria veicular, estratégia ainda incipiente, mas promissora para ampliar a cobertura, reduzir custos operacionais e aumentar a resiliência da rede. A capacidade de formar rotas alternativas de maneira autônoma entre os nós confere robustez inédita ao sistema, mitigando falhas comuns a redes móveis baseadas em operadoras. Esse diferencial abre caminho para a criação de redes de monitoramento mais independentes, escaláveis e menos dependentes de

infraestrutura externa, tornando a solução especialmente atrativa para áreas remotas ou regiões de difícil cobertura.

Além disso, este trabalho contribui para o avanço do conhecimento na área de redes LPWAN aplicadas ao setor de transportes, fornecendo evidências experimentais sobre o desempenho da tecnologia LoRa em cenários veiculares. Ao demonstrar na prática que a comunicação pode ser mantida com baixo consumo energético, mesmo em condições adversas, o estudo reforça a importância de soluções abertas, adaptáveis e de baixo custo para aplicações em larga escala.

Dessa forma, este estudo não apenas confirma a viabilidade técnica da telemetria veicular via LoRa, mas também abre caminho para avanços futuros, dentre os quais destacam-se:

- Inserção de repetidores e nós para expandir e estabilizar a cobertura da rede LoRa, especialmente em ambientes urbanos ou com grande presença de obstáculos físicos;
- Planejamento detalhado do enlace de comunicação, considerando a topologia da rede, a distância entre os nós e possíveis interferências locais, a fim de otimizar o desempenho geral do sistema;
- Otimização da placa de circuito impresso (PCB) e do invólucro de proteção para instalação embarcada nos veículos, com foco na organização dos componentes, no melhor aproveitamento de espaço e na resistência a condições ambientais adversas;
- Definição e estruturação de um banco de dados seguro e escalável, capaz de armazenar de forma eficiente os dados veiculares coletados pela rede;
- Desenvolvimento de um sistema supervisor com interface gráfica, que permita a visualização em tempo real das informações coletadas, bem como funcionalidades de análise e gestão dos dados de telemetria.

7 REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Medição de resistência de aterramento e de potenciais na superfície do solo em sistemas de aterramento. **ABNT NBR 15749**, p. 49, set. 2009.
- AHMED, S. E. A. Vegetation Effects on LoRa-Based Wireless Sensor Communication for Remote Monitoring of Automatic Orchard Irrigation Status. **IOT**, 6, 26 dezembro 2024. 2.
- AKYILDIZ, I. F.; WANG, X.; WANG, W. Wireless mesh networks: a survey. **Computer Networks**, 2005. 445–487.
- AL-FUQAHA, A. E. A. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. **IEEE Communications Surveys & Tutorials**, v. 17, p. 2347–2376, 2015.
- ALIEXPRESS. **GY-NEO6MV2 Novo módulo GPS**. [S.l.]. 2025.
- ALIEXPRESS. **Módulo CAN BUS SN65HVD230**. [S.l.]. 2025.
- ALIEXPRESS. **obd2 16 pinos fêmea conector**. [S.l.]. 2025.
- ALIEXPRESS. **obd2 conector capa com gabinete**. [S.l.]. 2025.
- AUTOMATION, H. **WiFi LoRa 32(V3), ESP32S3 + SX1262 LoRa Node, compatível com Meshtastic e LoRaWAN**. [S.l.]. 2023.
- B, A. P. **Connected Vehicles using LoRa**. International Conference on Smart Structures and Systems ICSSS. Thiruvananthapuram: IEEE. 2020. p. 5.
- BABIUCH, M.; FOLTÝNEK, P.; SMUTNÝ, P. **Using the ESP32 Microcontroller for Data**. 20TH International Carpathian Control Conference (ICCC). Wieliczka: [s.n.]. 2019. p. 1-2.
- BANZI, M. E. A. **Arduino Microcontroller Processing for Everyone!** 3. ed. [S.l.]: Springer, 2022.
- BHOOKYA, J. E. A. Implementation of PID controller for liquid level system using mGWO and integration of IoT application. **Journal of Industrial Information Integration**, Julho 2022. Acesso em: 13 Julho 2025.
- CHEN, F. E. A. ConservationBots: Autonomous aerial robot for fast robust wildlife tracking in complex terrains. **Journal of Field Robotics**, p. 443-469, Novembro 2023. Acesso em: 19 Junho 2025.
- CHILAMKURTHY, N. S. E. A. Low-Power Wide-Area Networks: A Broad Overview of Its Different Aspects. **IEEE Access**, 10, 2022. 81926–81959.
- DEBELL, T. E. A. OPEnS Hub: Real-Time Data Logging, Connecting Field Sensors to Google Sheets. **Frontiers in Earth Science**, v. 7, 31 Maio 2017. Acesso em: 14 Julho 2025.
- DEHROUYEH, F. E. A. On TinyML and Cybersecurity: Electric Vehicle Charging Infrastructure Use Case. **IEEE**, v. 12, p. 108703–108730, Agosto 2024.
- DIMCEV, V.; HANDJISKI, B.; SEKERINSKA, R. Alternative fall-of-potential method for grounding grids impedance measurements and inductive coupling between leads. **Electromagnetic Compatibility, 2003. EMC '03. 2003 IEEE International Symposium on**, v. 1, p. 74-77, May 2003.
- ESPRESSIF. **ESP32-WROOM-32**. [S.l.]. 2019.
- EUGSTER, P. T. E. A. The Many Faces of Publish/Subscribe. **ACM Computing Surveys**, v. 35, p. 114–131, 2003.

- FARAHANI, S. **ZigBee Wireless Networks and Transceivers**. Amsterdam: Elsevier: [s.n.], 2008.
- GUIMARÃES, A. D. A.; SARAIVA, A. M. **O Protocolo CAN: Entendendo e Implementando uma Rede de Comunicação Serial de Dados baseada no Barramento “Controller Area Network”**., 2002. 10.
- GUPTA, P.; KUMAR, P. R. The Capacity of Wireless Networks. **IEEE Transactions on Information Theory**, v. 46, Março 2020.
- HOANG, T. N.; VAN, S.-T.; NGUYEN, B. D. **ESP-NOW Based Decentralized Low Cost Voice Communication Systems For Buildings**. 2019 International Symposium on Electrical and Electronics Engineering (ISEE). Ho Chi Minh City, Vietnam: IEEE. 2019.
- IEEE - INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS. IEEE Guide for Measuring Earth Resistivity, Ground Impedance, and Earth Surface Potentials of a Grounding System. **IEEE Std 81-2012 (Revision of IEEE Std 81-1983)**, p. 1-86, Dec. 2012.
- INSTRUMENTS, T. **3.3-V CAN TRANSCEIVERS**. [S.I.]. 2002.
- INSTRUMENTS, T. **LM2596 SIMPLE SWITCHER Power Converter 150 kHz**. [S.I.]. 2013.
- JOSÉ SANTA ET AL. LPWAN-Based Vehicular Monitoring Platform with a Generic IP Network Interface, 2019. Acesso em: 04 Junho 2025.
- JOVALEKIC, N. E. A. LoRa transceiver with improved characteristics. **IEEE Wireless Communications Letters**, 06 Dezembro 2018. 1058–1061.
- KOLOBE, L.; LEBEKWE, C. K.; SIGWENI, B. LoRa Network Planning and Deployment: A Terrestrial Navigation Application. **Institute of Electrical and Electronics Engineers(IEEE)**, v. 9, p. 14, Setembro 2021.
- KOYANAGI, F., 13 Abril 2018. Disponível em: <<https://www.youtube.com/watch?v=7mLMwOJPh2M>>. Acesso em: 04 Junho 2025.
- LILYGO. **LoRa32 V2.1_1.6**. [S.I.]. 2025.
- MESHTASTIC. Range Tests. **Meshtastic**, 2025. Disponível em: <<https://meshtastic.org/docs/overview/range-tests/>>. Acesso em: 04 Junho 2025.
- MESSINA, F.; SANTORO, C.; SANTORO, F. F. Enhancing Security and Trust in Internet of Things through Meshtastic Protocol Utilising Low-Range Technology. **Electronics**, Catania, 13, 12 Março 2024. 1055.
- NIKOLOV, N.; GOTSEVA, D. **Make a prototype of IoT connected diagnostic tool using ESP32 and MQTT for reading data from car CAN bus OBD2**. 59th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST 2024. [S.I.]: IEEE. 2004.
- PAKDEL, M. **Advanced Programming with**. [S.I.]: Elektor, 2020.
- PAUL, C. R. **Eletromagnetismo para Engenheiros: Com Aplicações**. Rio de Janeiro: LTC, 2006.
- PLAUSKA, I.; LIUTKEVIČIUS, A.; JANAVIČIŪTĖ, A. Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller. **Electronics**, v. 12, p. 143, 28 Dezembro 2023.
- PURNAMA, A. A. F.; NASHIRUDDIN, M. I. **SigFox-based Internet of Things Network Planning for Advanced Metering Infrastructure Services in Urban Scenario**. In: IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). Edinburgh: IEEE. 2022. p. 17-19.

RADIOENGINE. **Módulo LoRaMESH: Manual de Utilização**. [S.l.]. 2021.

RODRÍGUEZ, A.; RAÚL, J.; INOUE, R. Implementation of an OBD-II Diagnostics Tool over CAN-BUS with Arduino, Pinar del Río, 02 Fevereiro 2018.

ROJNARONG, P.; PORA, W. **Signal Strength and Energy Consumption on**. 21st International Joint Conference on Computer Science and Software Engineering (JCSSE). Phuket, THAILAND: [s.n.]. 2024. p. 5.

SEMTECH. **SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver**. [S.l.]. 2015.

SINGH, R. K. E. A. Leveraging LoRaWAN Technology for Precision Agriculture in Greenhouses. **Sensors**, Abril 2020.

SINHA, R. S.; WEI, Y. Q.; HWANG, S. H. A survey on LPWA technology: LoRa and NB-IoT. **ICT Express**, 3, 2017. 14-21.

SUNDE, E. D. **Earth conduction effects in transmission systems**. New York: Dover Publications, INC., 1968.

U-BLOX. **NEO-6M**. [S.l.]. 2011.

WEIS, O. Conector OBD2 Explicado. **FLEXIHUB**, 4 Julho 2024. Disponível em: <<https://www.flexihub.com/pt/oobd2-pinout/>>.

ZEPHYR. **TTGO LoRa32**. [S.l.]. 2025.