

**UNIVERSIDADE DE CAXIAS DO SUL**  
**ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIA**

**MAURICIO DALZUCHIO**

**DESENVOLVIMENTO DE UM DISPOSITIVO DE AQUISIÇÃO DE DADOS REDE CAN  
PARA A LINHA AUTOMOTIVA**

**CAXIAS DO SUL**  
**2023**

**MAURICIO DALZUCHIO**

**DESENVOLVIMENTO DE UM DISPOSITIVO DE AQUISIÇÃO DE DADOS REDE CAN  
PARA A LINHA AUTOMOTIVA**

Trabalho de Conclusão de Curso II apresentado à Universidade de Caxias do Sul como requisito parcial à obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Professor Me. Ricardo Leal Costi

**CAXIAS DO SUL**

**2023**

Ficha catalográfica.

DALZOCHIO, Mauricio

Desenvolvimento de um dispositivo de aquisição de dados rede CAN para a linha automotiva / Mauricio Dalzochio; Orientador: Prof. Me. Ricardo Leal Costi.

64 folhas, 30 cm.

Trabalho de conclusão do curso de Engenharia Elétrica — Universidade de Caxias do Sul, Ciências exatas e Engenharias, 2023

Inclui referências bibliográficas.

1. OBD II 2. *Google Sheets*. I. Prof. Me. Ricardo Leal Costi II. Universidade de Caxias do Sul. III. Desenvolvimento de um dispositivo de aquisição de dados rede CAN para a linha automotiva.

**MAURICIO DALZUCHIO**

**DESENVOLVIMENTO DE UM DISPOSITIVO DE AQUISIÇÃO DE DADOS REDE CAN  
PARA A LINHA AUTOMOTIVA**

Trabalho de Conclusão de Curso II apresentado à Universidade de Caxias do Sul como requisito parcial à obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Professor Me. Ricardo Leal Costi

**Aprovado em 01/12/2023**

**Banca Examinadora**

---

Prof. Me. Ricardo Leal Costi (Orientador)  
Universidade de Caxias do Sul – UCS

---

Prof. Me. André Bernardes Michel  
Universidade de Caxias do Sul – UCS

---

Prof. Dr. Vagner do Nascimento  
Universidade de Caxias do Sul – UCS

*“Um homem inteligente tira mais proveito dos seus inimigos do  
que um tolo com seus amigos.”*

**Niki Lauda**

## RESUMO

O veículo automotor é o meio de transporte viário de pessoas e cargas, que está presente em escala global. Com o avanço da tecnologia, o monitoramento de dados tem sido uma ferramenta essencial no meio automotivo. Para entender o comportamento de um veículo em diferentes situações do mundo real, uma série de dados são analisados durante o processo de desenvolvimento através do conector OBD II, que está presente em todos os veículos com gerenciamento eletrônico do motor. Pensando nisso, foi desenvolvido um dispositivo de aquisição de dados automotivos, onde os dados poderiam ser coletados e armazenados em um servidor em nuvem. O principal objetivo é fazer leituras dos sensores de rotação, velocidade e posição do acelerador do veículo. Com base na bibliografia, foi desenvolvido um dispositivo de aquisição de dados que realiza a leitura dos sensores presentes no veículo. A rede CAN, é uma comunicação que envia pacotes contendo os códigos PID (*Parameter Identification*) de interesse, assim a ECU retorna os valores para o dispositivo de aquisição de dados. Através de uma conexão Wi-Fi, os dados são enviados ao servidor *online* onde o armazenamento e leitura ficam disponíveis para visualização. O dispositivo de aquisição de dados foi desenvolvido com o microcontrolador ESP32-S2-Wroom, onde é responsável pela conexão com a rede Wi-Fi e o processamento dos dados coletados do veículo. Os testes de validação do dispositivo de aquisição de dados foram realizados em dois veículos de passeio de porte pequeno, onde foram realizadas a coleta de dados em um percurso urbano de 650 metros. O dispositivo realizou a conexão com o veículo de forma estável, retornando os valores dos parâmetros requisitados. Os objetivos estipulados foram alcançados com êxito.

**Palavras chaves:** Veículo, dispositivo, aquisição, dados, OBD II, rede CAN.

## ABSTRACT

The motor vehicle is the standard way of transporting people and cargo by road, which is present on a global scale. With the advancement of technology, data monitoring has become an essential tool in the automotive world. To understand the behavior of a vehicle in different real-world situations, a series of data are analyzed during the development process through the OBD II connector, which is present in all vehicles with electronic engine management. With this in mind, an automotive data acquisition device was developed, where data could be collected and stored on a cloud server. The main objective is to take readings from the vehicle's rotation, speed, and throttle position sensors. Based on the research, a data acquisition device capable of reading vehicle sensors from the vehicle was developed. The CAN network is a communication protocol that sends packets containing the PID (Parameter Identification) codes of interest, so the ECU can return the values to the data acquisition device. Through a Wi-Fi connection, data is sent to the online server where it is stored and is available for reading on a computer. The data acquisition device was developed with the ESP32-S2-Wroom microcontroller, which is responsible for connecting to the Wi-Fi network and processing the vehicles collected data. Device's validation tests were conducted in two small passenger vehicles, where data were collected on a 650-meter urban route. The device's connection to the vehicle was stable, returning values for the requested parameters. The stipulated objectives were successfully achieved.

**Keywords:** Vehicle, datalogger, data, OBD II, CAN network.

## LISTA DE FIGURAS

Figura 1 – Diagrama do gerenciamento de um veículo.....	10
Figura 2 – Conector padrão OBD II.....	12
Figura 3 – Teste de condução genérico.....	14
Figura 4 – Comparativo sistema convencional e CAN.....	15
Figura 5 – Níveis lógicos do barramento CAN.....	16
Figura 6 – Comunicação CAN entre centrais.....	16
Figura 7 – Formato de mensagem CAN.....	17
Figura 8 – Conexão física CAN norma ISO-11898.....	18
Figura 9 – Aplicação típica do controlador junto ao transceptor.....	19
Figura 10 – Bandas ISM utilizadas nos espectros UHF e SHF.....	20
Figura 11 – Camadas do protocolo TCP/IP.....	21
Figura 12 – <i>Hardware</i> protótipo e aplicativo.....	23
Figura 13 – Plataforma de desenvolvimento ESP32.....	25
Figura 14 – Arquitetura do dispositivo de aquisição de dados.....	26
Figura 15 – Periféricos do dispositivo de aquisição de dados.....	30
Figura 16 – Circuito do microcontrolador.....	31
Figura 17 – Circuito de alimentação 3,3 V.....	32
Figura 18 – Circuito interface visual com o usuário.....	33
Figura 19 – Circuito de comunicação com o veículo.....	34
Figura 20 – Placa de circuito do dispositivo de aquisição de dados.....	36
Figura 21 – Esquemático do módulo transceptor CAN.....	36
Figura 22 – Esquemático do módulo ESP32 Saola.....	37
Figura 23 – Diagrama de montagem do chicote.....	38
Figura 24 – Dispositivo de aquisição de dados.....	38
Figura 25 – Diagrama de estados e transições TWAI.....	39
Figura 26 – Fluxograma do <i>firmware</i> da rede CAN.....	40
Figura 27 – Fluxograma do <i>firmware</i> Wi-Fi.....	41
Figura 28 – Testes de funcionamento do <i>hardware</i> .....	42
Figura 29 – Verificação do estado do led.....	43
Figura 30 – Verificação de conexão com a rede Wi-Fi.....	43
Figura 31 – Verificação de recebimento dos dados no servidor.....	44
Figura 32 – Testes de aplicação monitorado pela IDE.....	45



Figura 33 – Respostas da ECU no monitor da IDE e o sinal coletado no osciloscópio .....	45
Figura 34 – Respostas no formato CAN 11 <i>bits</i> no monitor da IDE e no osciloscópio.....	46
Figura 35 – Instrumentação do veículo <i>Onix</i> .....	46
Figura 36 – Percurso do teste de condução.....	47
Figura 37 – Registro parcial do teste de condução com o veículo <i>Onix</i> .....	47
Figura 38 – Aquisição dos dados do teste final no veículo <i>Up</i> . .....	48
Figura 39 – Registro parcial do teste de condução com o veículo <i>Up</i> .....	48

## LISTA DE TABELAS E QUADROS

Quadro 1 – Exemplos de SID .....	13
Tabela 1 – Exemplo de pergunta e resposta a ECU .....	28
Tabela 2 – Capacitores de filtragem da fonte de tensão .....	33
Tabela 3 – Capacidade de corrente por trilha .....	35

## LISTA DE EQUAÇÕES

Equação 1 – Rotação do Motor .....	28
Equação 2 – Velocidade Instantânea do Veículo .....	28
Equação 3 – Posição da Borboleta de Aceleração .....	29
Equação 4 – Cálculo do Capacitor de Filtragem .....	32
Equação 5 – Cálculo da resistência de referência .....	34
Equação 6 – Cálculo da corrente máxima da trilha do PCB.....	35

## LISTA DE SIGLAS

ACK	<i>Acknowledge</i>
BSS	<i>Base Transceiver System</i>
BCM	<i>Body Module Control</i>
CAN	<i>Controller Area Network</i>
CAN-L	Sinal baixo do barramento CAN
CAN-H	Sinal alto do barramento CAN
CCM	<i>Central Control Module</i>
CRC	<i>Cycle Redundancy Check</i>
CSV	<i>Comma-Separated Values</i>
DLC	<i>Data Length Code</i>
DTC	<i>Diagnostic Trouble Code</i>
ECU	<i>Eletronic Control Unit</i>
ETSI	<i>European Telecommunications Standards Institute</i>
EPROM	<i>Erasable Programmable Ready Only Memory</i>
FR	<i>Fuel Rate</i>
GEM	<i>General Electronic Module</i>
GM	<i>General Motors</i>
GPRS	<i>General Packet Radio Service</i>
ID	Identificação
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
Mbps	Megabits por segundo
MEMS	<i>Microelectromechanical systems</i>
OBD II	<i>On Board Diagnostic</i>
PCM	<i>Powertrain Control Module</i>
PID	<i>Parameter Identification</i>
RPM	Revoluções por minuto
RTR	<i>Remote Transmition Request</i>
SAE	<i>Society of Automotive Engineers</i>
SID	<i>Service Identification</i>
SPI	<i>Serial Peripheral Interface</i>
TPS	<i>Throttle Position Sensor</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
Wi-Fi	<i>Wireless Fidelity</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>6</b>
1.1	JUSTIFICATIVA.....	7
1.2	AMBIENTE DE TRABALHO .....	7
1.3	OBJETIVOS.....	8
<b>1.3.1</b>	<b>Objetivo Geral</b> .....	<b>8</b>
<b>1.3.2</b>	<b>Objetivos Específicos</b> .....	<b>8</b>
1.4	DESCRIÇÃO GERAL DO TRABALHO .....	8
<b>1.4.1</b>	<b>Limites do Trabalho</b> .....	<b>8</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>9</b>
2.1	GERENCIAMENTO ELETRÔNICO.....	9
<b>2.1.1</b>	<b>Unidade de controle eletrônica</b> .....	<b>9</b>
<b>2.1.2</b>	<b>Sensores de um veículo</b> .....	<b>10</b>
<b>2.1.3</b>	<b>Atuadores de um veículo</b> .....	<b>11</b>
2.2	PARÂMETROS DE LEITURA .....	12
<b>2.2.1</b>	<b>Conector de Diagnose OBD II</b> .....	<b>12</b>
<b>2.2.2</b>	<b>Coleta dos dados</b> .....	<b>13</b>
<b>2.2.3</b>	<b>Serviço 01: Exibição dos dados atuais</b> .....	<b>13</b>
<b>2.2.4</b>	<b>Serviço 09: Acompanhamento de desempenho em uso</b> .....	<b>14</b>
2.3	CONTROLLER AREA NETWORK .....	15
<b>2.3.1</b>	<b>Protocolo CAN</b> .....	<b>15</b>
<b>2.3.2</b>	<b>Formato de mensagem CAN</b> .....	<b>16</b>
<b>2.3.3</b>	<b>Conexão física com o barramento CAN</b> .....	<b>18</b>
<b>2.3.4</b>	<b>Transceptor CAN</b> .....	<b>18</b>
<b>2.3.5</b>	<b>Controlador CAN</b> .....	<b>19</b>
2.4	REDE <i>WIRELESS</i> .....	19
<b>2.4.1</b>	<b><i>Wireless fidelity</i></b> .....	<b>20</b>
<b>2.4.2</b>	<b>Protocolo TCP/IP</b> .....	<b>20</b>
<b>2.4.3</b>	<b>Serviço cliente servidor</b> .....	<b>21</b>
<b>2.4.4</b>	<b>Sistemas em <i>Cloud</i></b> .....	<b>21</b>
2.5	TRABALHOS CORRELATOS .....	22
<b>2.5.1</b>	<b>CAN automotivo sistema de monitoramento</b> .....	<b>22</b>
<b>2.5.2</b>	<b>Sistema de monitoramento automotivo via rede CAN</b> .....	<b>23</b>
<b>3</b>	<b>METODOLOGIA</b> .....	<b>24</b>
3.1	FERRAMENTAS DE DESENVOLVIMENTO.....	24

3.1.1	<b>Software EasyEDA</b> .....	24
3.1.2	<b>Software Visual Studio Code</b> .....	24
3.1.3	<b>Plataforma de desenvolvimento ESP32</b> .....	24
3.2	LÓGICA DE FUNCIONAMENTO .....	25
3.3	AQUISIÇÃO DE DADOS .....	26
3.3.1	<b>Protocolo de comunicação</b> .....	27
3.3.2	<b>Definição dos parâmetros</b> .....	27
3.3.3	<b>Aquisição dos parâmetros</b> .....	27
3.3.4	<b>Taxa de Amostragem</b> .....	29
3.3.5	<b>Processamento e envio dos dados</b> .....	29
3.4	TESTES FUNCIONAIS.....	30
4	<b>DESENVOLVIMENTO</b> .....	31
4.1	ARQUITETURA DO <i>HARDWARE</i> PROPOSTO .....	31
4.1.1	<b>Circuito de processamento e transmissão dos dados</b> .....	31
4.1.2	<b>Circuito de alimentação</b> .....	32
4.1.3	<b>Circuito de interface visual</b> .....	33
4.1.4	<b>Circuito de comunicação com o veículo</b> .....	33
4.1.5	<b>Placa de circuito impresso</b> .....	35
4.2	ARQUITETURA DO <i>HARDWARE</i> MODULAR.....	36
4.2.1	<b>Chicote de interface com o veículo</b> .....	37
4.2.2	<b>Gabinete de proteção do PCB</b> .....	38
4.3	DESENVOLVIMENTO DO <i>FIRMWARE</i> .....	39
4.3.1	<b>Rede CAN</b> .....	39
4.3.2	<b>Rede Wi-Fi</b> .....	41
5	<b>RESULTADOS E DISCUSSÕES</b> .....	42
5.1	TESTES DE <i>HARDWARE</i> .....	42
5.2	TESTES DE <i>FIRMWARE</i> .....	43
5.3	TESTES DE APLICAÇÃO .....	44
6	<b>CONCLUSÃO</b> .....	49
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	50
	<b>ANEXO A – FOLHA DE DADOS DA FAMÍLIA DO ESP32-S2-WROOM-I</b> .....	52
	<b>ANEXO B – LISTA DE PARÂMETROS PID DE UM VEÍCULO</b> .....	53
	<b>APÊNDICE A – GRÁFICO DO TESTE DE CONDUÇÃO COM O VEÍCULO ONIX</b> .....	54
	<b>APÊNDICE B – REGISTRO DO TESTE DE CONDUÇÃO COM O VEÍCULO ONIX</b> .....	55

## 1 INTRODUÇÃO

Com o surgimento do motor a combustão interna e do veículo automotor no final do século XIX, a humanidade passou a ter outro meio de locomoção, além das grandes máquinas movidas a vapor. O sucesso dos veículos automotores deve-se a sua produção em massa, que foi popularizada em escala global. Esse sucesso tornou o veículo em algo mais do que um meio de transporte, uma paixão, uma competição esportiva e um estilo de vida.

Com o passar dos anos, diversas tecnologias foram empregadas para melhorar o desempenho e acrescentar funcionalidades aos veículos, esse desenvolvimento deve-se a vários fatores como o avanço da tecnologia de eletrônica embarcada, linhas de produção em série, assim como a escassez de combustíveis e matérias primas. Muitas das tecnologias que empregam os veículos atualmente possuem origem militar, ou das competições automotivas após esse período. As competições funcionam como um laboratório de desenvolvimento para novas tecnologias, o maior exemplo disso é o sistema de aquisição de dados. Este sistema foi desenvolvido no final dos anos 70 para a Fórmula 1, com o intuito de medir, registrar e avaliar as mudanças realizadas nos monopostos. Essa solução foi adotada por todas as modalidades de competição automotiva e popularizada com o avanço da eletrônica embarcada.

Com o avanço do uso de sistemas eletrônicos embarcados, pode-se garantir a segurança, o conforto e a confiabilidade de um veículo. Em 1978, a GM (*General Motors*) ofereceu como opcional em seu Cadillac, o computador de bordo. O computador consistia em um microprocessador Motorola 6802 modificado que exibia informações de velocidade, consumo de combustível, distância percorrida e rotação do motor. No entanto o microprocessador cumpria outra função, que era o controle e o monitoramento dos sistemas essenciais do veículo, tais como injeção de combustível, temporização eletrônica de ignição e controle de cruzeiro (CHARETTE, 2009).

Os veículos possuem múltiplas unidades de controle eletrônico instaladas em diversos pontos da carroceria, estando conectadas entre si através da rede de comunicação CAN (*Controller Area Network*). As informações são centralizadas na unidade de controle principal, onde as informações podem ser acessadas a partir do conector de diagnose, esses dados estão disponíveis para avaliar o funcionamento correto do veículo.

A proposta do trabalho é desenvolver um dispositivo de aquisição de dados, conectado ao sistema de diagnose, destinado a veículos da linha leve. O objetivo é acessar a rede CAN do automóvel e adquirir os dados disponibilizados pelos módulos de gerenciamento do veículo.

## 1.1 JUSTIFICATIVA

Os avanços tecnológicos permitiram que empresas pudessem reavaliar a metodologia de processos, de forma a monitorar e controlar seus processos produtivos. O principal objetivo disso é garantir um aumento de produtividade e redução de custos sem comprometer a estrutura e o orçamento do processo implementado. A aquisição de dados é fundamental para a avaliação desse controle, essa tecnologia encontra-se presente nas indústrias e nos veículos automotores.

A aquisição de dados no setor automotivo tem sido fundamental para o desenvolvimento de veículos, uma série de dados são analisados durante este processo para obter o comportamento de um veículo em diversas situações. Como citado anteriormente, o maior exemplo disso é o desenvolvimento realizado na Fórmula 1, que conta com sistemas de aquisição de dados que registram o comportamento dos monopostos nas pistas.

O desenvolvimento de um dispositivo de aquisição de dados permite coletar informações da ECU (*Electronic Control Unit*), que podem ser avaliadas posteriormente possibilitando a avaliação do desempenho do veículo em uma determinada condição de uso. Desta forma, esse trabalho foi desenvolvido com base na aquisição de dados, com o intuito de realizar as leituras dos sensores de um veículo da linha leve. Podendo esse dispositivo, se tornar uma ferramenta para avaliar o comportamento de veículos, oferecendo uma solução economicamente viável, confiável e adaptada para cada necessidade da aplicação.

## 1.2 AMBIENTE DE TRABALHO

A Universidade de Caxias do Sul (UCS) é uma instituição comunitária de educação superior, fundada em 10 de fevereiro de 1967. Possui seu Campus-Sede na cidade de Caxias do Sul, com atuação nos campos de ensino, pesquisa, inovação e extensão.

Para o desenvolvimento de projetos das disciplinas de eletrônica, assim como trabalho de conclusão de curso, a UCS disponibiliza aos estudantes locais apropriados para desenvolvimento do projeto, as salas 104, 109 e 111 do Bloco 71. Estes espaços em específico contam com bancadas didáticas, com computadores equipados com softwares para desenvolvimento e simulações de circuitos. A Universidade possui um laboratório que disponibiliza componentes eletrônicos, tais como microcontroladores, portas lógicas, entre outros circuitos integrados. Também disponibiliza ferramentas diversas como fontes de tensão, multímetros, osciloscópios entre outras. Além disto, conta com o apoio de um técnico de laboratório para suporte, esta estrutura está disponível aos alunos de engenharia.



## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

O objetivo do presente trabalho é desenvolver um dispositivo de aquisição de dados para realizar leituras dos sensores disponíveis de um veículo de porte leve, através da rede de comunicação pelo conector OBD II. Segundo o CONTRAN (Conselho Nacional de Trânsito, 2005), os veículos de porte leve são os que possuem peso bruto até 3500 kg e utilizam os combustíveis como o etanol e a gasolina.

### 1.3.2 Objetivos Específicos

De forma a atingir o objetivo geral, abaixo se destacam os objetivos específicos:

- Definir sensores de interesse;
- Desenvolver um *hardware* e *firmware*;
- Realizar a coleta dos dados e enviá-los para o servidor na nuvem;
- Realizar os testes de verificação e validação do dispositivo;

## 1.4 DESCRIÇÃO GERAL DO TRABALHO

O presente trabalho foi dividido em capítulos, organizando nas seguintes estruturas descritas abaixo:

- a) O capítulo I aborda a parte introdutória referente ao trabalho a ser desenvolvido.
- b) O capítulo II aborda os referenciais teóricos que compõem a estrutura.
- c) O capítulo III aborda a metodologia utilizada durante o desenvolvimento.
- d) O capítulo IV aborda o desenvolvimento do projeto.
- e) O capítulo V aborda os resultados obtidos, durante o desenvolvimento do projeto.
- f) O capítulo VI aborda as contribuições deixadas pelo projeto desenvolvido.

### 1.4.1 Limites do Trabalho

O projeto abrange o desenvolvimento e implementação de *hardware* e *firmware*. Este sistema apenas realiza a coleta dos dados não realizando a análise deles, seja por uma pessoa ou algoritmo. O servidor na nuvem foi implementado por uma equipe externa, não sendo o foco de desenvolvimento deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o conteúdo teórico e fundamental para a elaboração do projeto. Refere-se as conceituações e definições fundamentais de um sistema de gerenciamento de um veículo.

### 2.1 GERENCIAMENTO ELETRÔNICO

O conceito de gerenciamento eletrônico do motor de combustão interna não é novo. Em 1939, a Bosch realizava testes no primeiro sistema de injeção direta de gasolina em motores de aviões. Em 1954, era fabricado o primeiro veículo com injeção eletrônica, o Mercedes Benz 300 SL. Os marcos históricos decisivos para o surgimento do controle eletrônico de motores a combustão, foram a criação de novas regras de emissão de poluentes no mundo em 1967 e a crise global de petróleo em 1973 (CAPELLI, 2010).

#### 2.1.1 Unidade de controle eletrônica

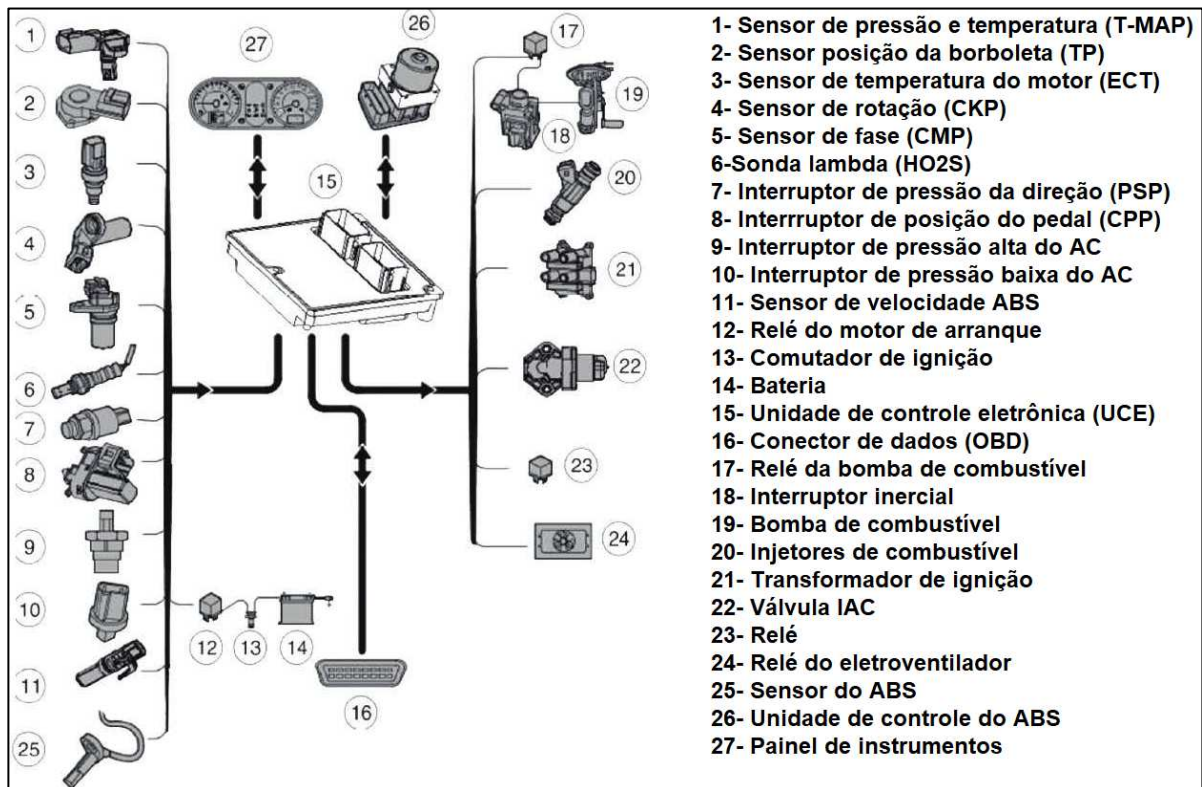
A ECU (*Electronic Control Unit*) é o dispositivo responsável por gerenciar as leituras dos sensores (entradas) e os atuadores (saídas), este gerenciamento garante o devido funcionamento do veículo. Esta funciona semelhante a um computador, possui um microcontrolador e um programa gravado na memória. Existem outras unidades de controle no veículo destinadas a seus respectivos sistemas, separadas pelas funções realizadas e pelas suas características técnicas (GUIMARÃES, 2007).

As unidades de controle estão subdivididas em módulos que se comunicam entre si, por meio de rede CAN Bus e são identificadas através de siglas. Abaixo seguem alguns exemplos de unidades de controle:

- ACU (*Airbag Control Unit*): Módulo de controle do Airbag;
- BCM (*Body Control Module*): Módulo de controle do chassi;
- CCM (*Central Control Module*): Módulo de controle central;
- GEM (*General Electronic Module*): Módulo eletrônico geral;
- EBD (*Electronic Brake Force Distribution*): Módulo de controle dos freios;
- ECM (*Engine Control Module*): Módulo de controle do motor;
- MT (*Multitimer*): Módulo de temporizações do veículo;
- SCM (*Suspension Control Module*): Módulo de controle da suspensão;
- TCM (*Transmission Control Module*): Módulo de controle da transmissão;

Atuadores, módulos e sensores são componentes instalados ao longo do veículo com a finalidade de receber ou enviar informações à ECU. Esses dados permitem o gerenciamento do veículo como um todo. Na Figura 1 pode ser visto o diagrama do sistema de gerenciamento do veículo (GUIMARÃES, 2007).

Figura 1 – Diagrama do gerenciamento de um veículo



Fonte: Adaptado de Comunicación (2023)

### 2.1.2 Sensores de um veículo

Os sensores são componentes que convertem grandezas físicas de diversas naturezas em variações de natureza elétrica. Por exemplo, o sensor de posicionamento da borboleta converte deslocamento angular mecânico em variação da sua resistência elétrica (CAPELLI, 2010). Abaixo segue uma lista de alguns sensores disponíveis em um veículo:

- Sensor de rotação e PMS (Ponto Morto Superior): É o sensor que informa a rotação do motor e a referência do PMS dos êmbolos para a ECU;
- Sensor de fase: Esse sensor informa a referência da fase do primeiro cilindro, para determinar a injeção de combustível e a ignição;
- Medidor de massa do ar: Esse sensor mede a quantidade de massa de ar, que é admitida pelo motor do veículo;

- Sensor de temperatura do ar: Esse sensor informa a temperatura do ar, que é admitida pelo motor do veículo;
- Sensor de posição da borboleta: É o sensor responsável por informar a ECU o ângulo de abertura da borboleta de aceleração;
- Sensor de oxigênio: Esse sensor mede a concentração de oxigênio nos gases da exaustão do veículo, é responsável por garantir a estequiometria da mistura ar/combustível;
- Sensor de velocidade: Também conhecido como sensor taqueométrico, informa a velocidade do veículo para a ECU.
- Sensor de pressão absoluta: Conhecido como sensor MAP (*Manifold Absolute Pressure*), tem como função informar as variações de pressão existentes no coletor de admissão e a pressão atmosférica local, para cálculo da massa de ar admitida.
- Sensor inercial: Também conhecido como interruptor inercial, é responsável por desativar a bomba de combustível em caso de um acidente com o veículo;
- Sensor de detonação: Possui a função de transformar as vibrações mecânicas do motor em relações elétricas capazes de serem interpretadas pela ECU.

### 2.1.3 Atuadores de um veículo

Os atuadores são componentes que executam o gerenciamento das grandezas físicas no motor. Basicamente respondem mecanicamente aos estímulos elétricos enviados pela ECU (CAPELLI, 2010). Abaixo segue uma lista de alguns atuadores disponíveis em um veículo:

- Atuador da marcha lenta: Possui um motor de passo, que limita a quantidade de ar para manter a rotação do motor constante. O corpo da borboleta possui um orifício por onde parte do ar admitido passa, que é regulado dinamicamente pelo eixo do atuador;
- Transformador de ignição: Cumpre a função de transformar um nível de tensão baixo em outro mais alto, para alimentar as velas de ignição;
- Válvulas injetoras: Esses componentes são válvulas solenoides, que por meio de um campo magnético, atraem uma agulha, liberando a passagem de combustível por um orifício localizado na sua base. Na ausência de sinal, uma mola de retorna a agulha para a sede e veda a passagem de combustível;
- Relés: É um interruptor no qual os contatos são acionados por meio do campo magnético. Dentre as funções que o relé desempenha no sistema de injeção/ignição está a alimentação da eletrobomba de combustível.

## 2.2 PARÂMETROS DE LEITURA

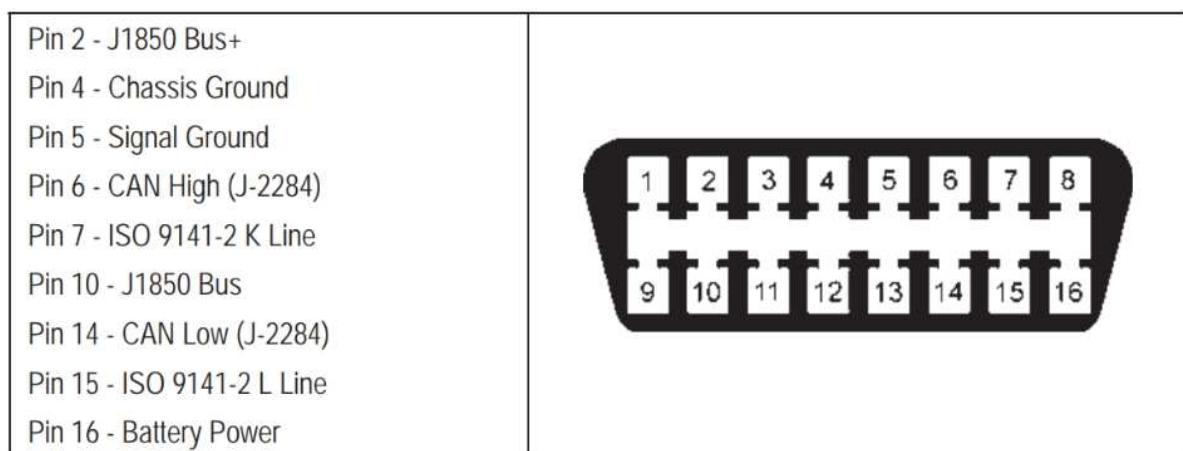
A ECU disponibiliza os parâmetros do veículo através do conector de diagnose, onde os dados podem ser coletados, através de um *scanner* ou outro dispositivo. Os primeiros sistemas de aquisição de dados foram introduzidos no mundo automotivo na Fórmula 1 e são a base de parâmetros que os engenheiros usam para determinar a configuração correta do veículo na pista (Universia, 2021). Da mesma forma, esses dados podem ser utilizados para monitorar os veículos da linha leve.

### 2.2.1 Conector de Diagnose OBD II

Conforme Guimarães (2007), o termo diagnóstico veicular representa as funções ou ferramentas que permitem a programação ou verificação do funcionamento de cada módulo eletrônico existente em um veículo. Dentro desse conceito, a indústria automotiva e as organizações de padronização industriais, trabalham na padronização de protocolos de diagnóstico veicular.

De forma a realizar uma padronização para a leitura, a norma ISO (*International Organization for Standardization*) 15031, especifica os aspectos de comunicação via OBD II (*On-Board Diagnostic II*) entre o veículo e o equipamento de diagnóstico externo, permitindo a realização da leitura dos parâmetros e dos sensores do veículo (GUIMARÃES, 2007). Este conector tornou-se padrão nos veículos que contam com o gerenciamento eletrônico, sendo um conector de duas fileiras com oito pinos cada, totalizando em dezesseis pinos, como é possível ver na Figura 2.

Figura 2 – Conector padrão OBD II



Fonte: Adaptado de Guimarães (2007)

### 2.2.2 Coleta dos dados

Os dados de leitura coletados via conector OBD II são os valores de sensores ou parâmetros disponibilizados pela ECU que permitem entender o funcionamento e o uso do veículo (GUIMARÃES, 2007). Os dados são coletados a partir leitura dos registradores da ECU e não diretamente dos sensores e essa leitura ocorre no formato pergunta e resposta por meio da rede CAN (CUNHA, 2011).

A norma ISO 15031, especifica os aspectos de comunicação via OBD II entre o veículo e o equipamento de diagnóstico externo em termos de interpretação de dados e tempos de resposta. Também define dois parâmetros importantes para serem definidos no formato de pergunta e resposta (ISO-15031, 2016):

- **SID (*Service Identification Description*):** Identificação de serviço solicitado. A identificação de serviço consiste num número hexadecimal entre 01 e 0A. No Quadro 1 pode ser visto os identificadores SID de uma ECU;
- **PID (*Parameter Identification Description*):** Identificação dos parâmetros que se pode solicitar à ECU. Esses parâmetros são identificados por um número Hexadecimal entre 00 e FF.

Quadro 1 – Exemplos de SID

Serviço	Descrição	Serviço	Descrição
\$01	Tela de status de teste da prontidão do sistema.	\$05	Resultados do teste de monitoramento do sensor de oxigênio <sup>1</sup>
	Status e controle MIL (lâmpada do indicador do mau funcionamento)	\$06	Resultados do teste específico do sistema de diagnóstico monitorado
	Solicitação dos dados de diagnóstico atuais do powertrain	\$07	Códigos de problemas de diagnóstico relacionados à emissão durante o último ciclo de condução concluído ou o atual (=códigos pendentes)
\$02	Solicitação dos dados de captura de tela do powertrain	\$08	Testes de ativação do sistema de bordo <sup>2</sup>
\$03	Solicitação de DTCs (códigos de problema de diagnóstico) relacionados à emissão	\$09	Solicitar informações do veículo
\$04	Limpar/reiniciar dados de diagnóstico relacionados à emissão	\$09	Acompanhamento de desempenho em uso
		\$0A	DTC relacionado à emissão com status permanente

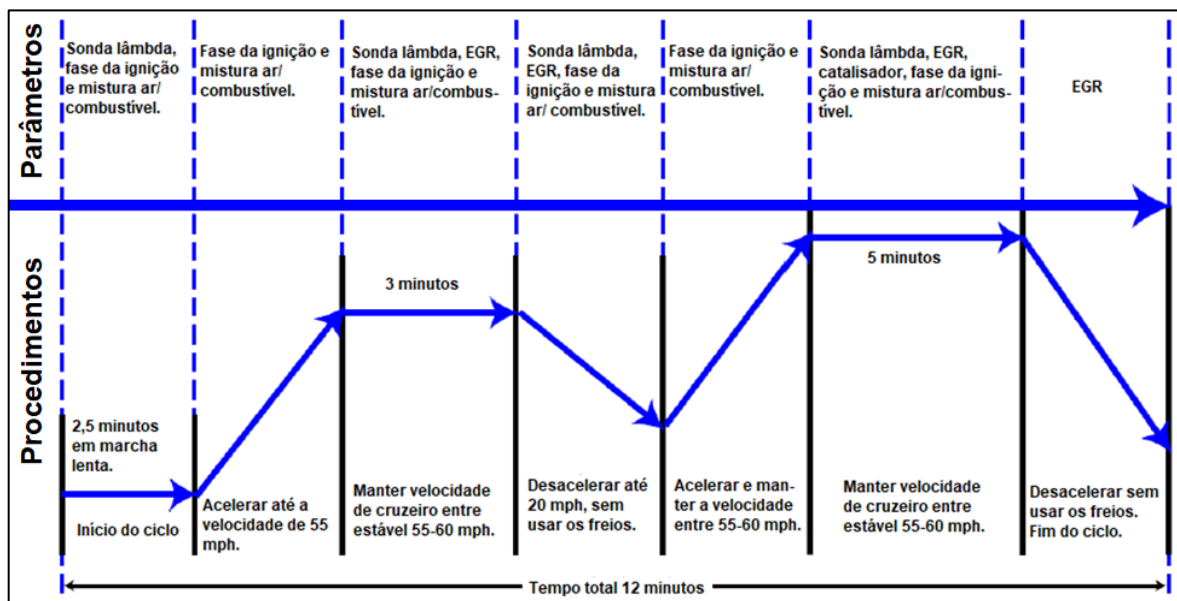
Fonte: Adaptado de SNAP-ON (2013)

### 2.2.3 Serviço 01: Exibição dos dados atuais

A ECU é a responsável por transmitir os dados do veículo que são conhecidos como PID e exibem as leituras atuais dos sensores. A taxa de amostragem pode variar dependendo da quantidade de parâmetros selecionados. É possível monitorar as funções dos subsistemas, sejam contínuos como o monitoramento do sistema de combustível, assim como o não contínuo tal qual o monitoramento da válvula de recirculação de gases (SNAP-ON, 2013).

Cada fabricante possui critérios específicos podendo variar de modelo. Com um scanner ou um dispositivo de aquisição de dados pode ser realizado um teste, para a verificação dos parâmetros do veículo, um exemplo de teste que pode ser realizado é o teste de ciclo de condução. Nesse teste o veículo inicia o aquecimento do motor em marcha lenta, logo após é acelerado até determinada velocidade, percorre em velocidade de cruzeiro e desacelera sem frear o veículo (SNAP-ON, 2013). Na Figura 3 pode ser visto um exemplo genérico do teste.

Figura 3 – Teste de condução genérico



Fonte: Adaptado de SNAP-ON (2013)

#### 2.2.4 Serviço 09: Acompanhamento de desempenho em uso

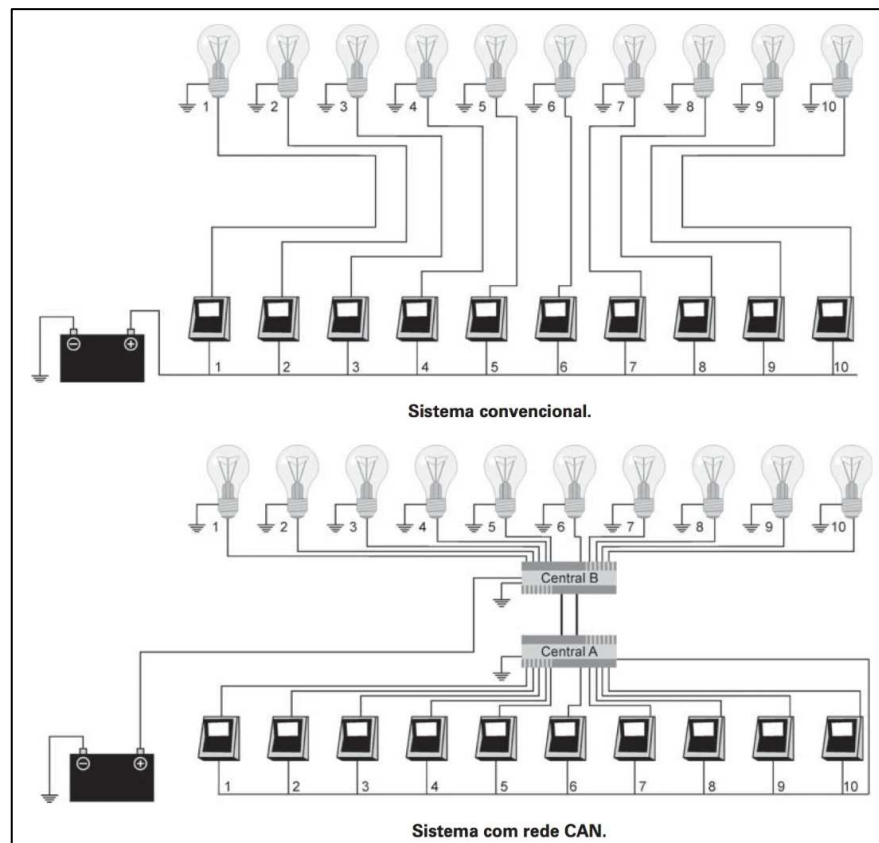
Os fabricantes de veículos devem implementar uma forma de verificar o desempenho dos sistemas de gerenciamento, esta deve ser realizada no algoritmo da ECU. As informações podem ser exigidas por órgãos regulamentários e devem possuir um formato padrão. As informações específicas de um veículo, são disponibilizadas através do identificador de serviço 09 (SNAP-ON, 2013). Abaixo seguem algumas das informações disponibilizadas:

- Número de identificação do veículo;
- Identificadores de calibração;
- Números de verificação de calibração.

## 2.3 CONTROLLER AREA NETWORK

A rede CAN (*Controller Area Network*) foi desenvolvida em 1986 por Robert Bosch, com o intuito de reduzir a quantidade de condutores elétricos em um chicote. Devido ao aumento de componentes eletrônicos, mais condutores eram necessários em um chicote para a conexão dos mesmos, o que resultava em um maior volume para acomodá-lo. Com a implementação da rede CAN foi possível reduzir o número de condutores utilizados no chicote de um veículo (CAPELLI, 2010). Na Figura 4 pode ser visto um exemplo comparativo entre as duas situações.

Figura 4 – Comparativo sistema convencional e CAN



Fonte: Adaptado de Capelli (2010)

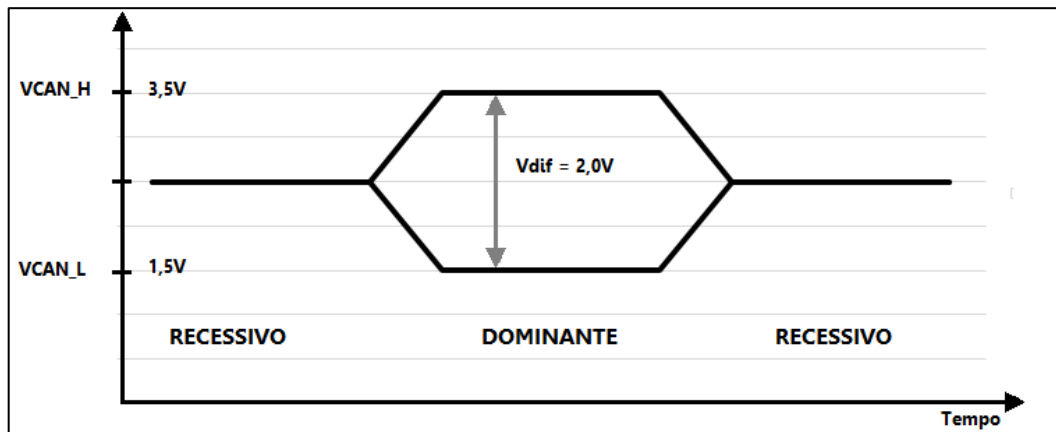
### 2.3.1 Protocolo CAN

De acordo com Guimarães (2007), CAN é um protocolo de comunicação serial síncrono. Este sincronismo entre os módulos é realizado no início de cada mensagem enviada ao barramento, este evento ocorre em intervalos de tempo conhecidos e regulares. A velocidade de transmissão varia inversamente proporcional ao comprimento do barramento, a taxa de transmissão máxima é de 1 Mbps para um barramento de até 40 metros.



O barramento CAN pode ser formado por dois ou quatro condutores elétricos, os condutores são trançados e blindados para evitar interferências eletromagnéticas. Cada *bit* é transmitido de forma constante por um valor de tensão específico, eles podem ser definidos como dominantes (1) ou recessivos (0). Fisicamente, o CAN\_L (L de *Low* = baixo) é aproximadamente 1,5 V, e o CAN\_H (H de *High* = alto), aproximadamente 3,5 V (Figura 5) (GUIMARÃES, 2007).

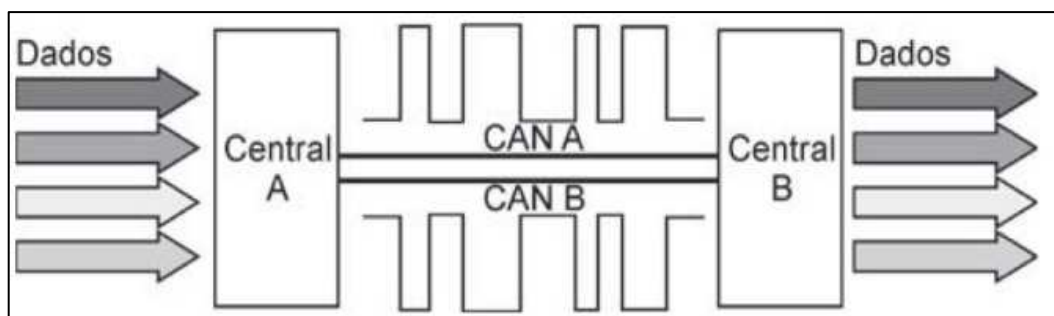
Figura 5 – Níveis lógicos do barramento CAN



Fonte: Adaptado de Guimarães (2007)

O barramento é composto por dois condutores de comunicação, CAN "A" e CAN "B" por uma questão de segurança. O mesmo sinal enviado pelo CAN A é também enviado pelo CAN B, mas de forma invertida (Figura 6). Esse método permite que o sistema continue funcionando mesmo com um condutor interrompido (CAPELLI, 2010).

Figura 6 – Comunicação CAN entre centrais



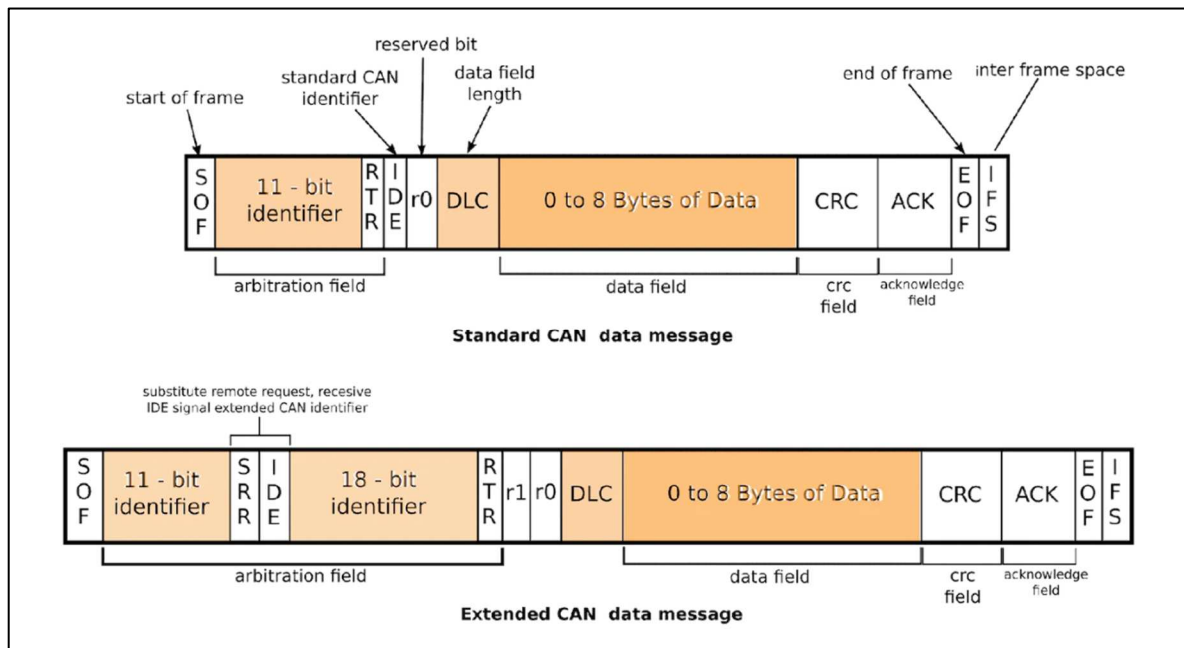
Fonte: Adaptado de Capelli (2010)

### 2.3.2 Formato de mensagem CAN

Há dois formatos de protocolos CAN denominados CAN 2.0 A e CAN 2.0 B. O primeiro possui um identificador de onze *bits* enquanto o outro possui um identificador de vinte

e nove *bits* (GUIMARÃES, 2007). Na Figura 7, pode ser visto o formato de mensagem de ambos os protocolos. O número de *bits* do identificador limita a quantidade de mensagens suportada. O tempo de transmissão está associado ao tamanho da mensagem, quanto maior for a mensagem maior será o tempo de transmissão (DEWESOFT, 2019).

Figura 7 – Formato de mensagem CAN



Fonte: Adaptado de Dewesoft (2019)

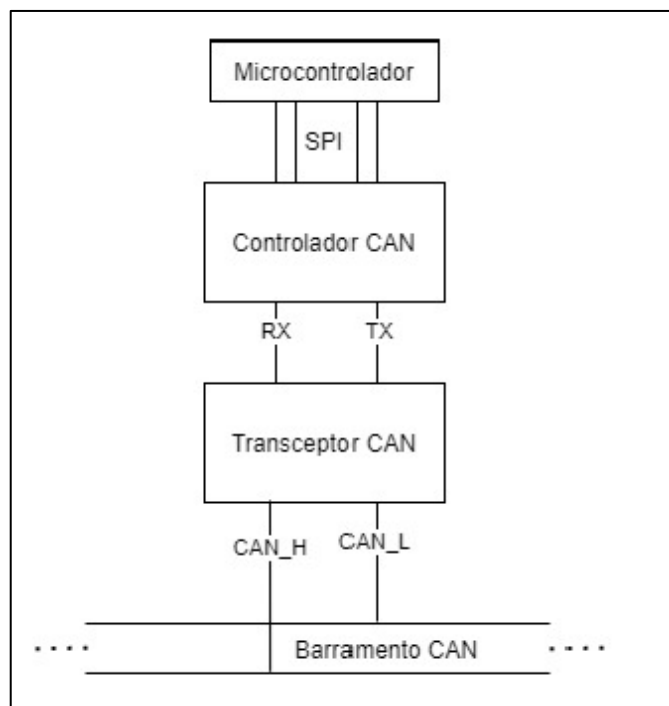
- SOF (*Start of Frame*): É o *bit* que identifica o início de uma mensagem e sincroniza os elementos conectados ao barramento;
- Identificador: *Bits* que definem a prioridade da mensagem;
- RTR (*Remote Transmission Request*): É o *bit* que faz a requisição de transmissão remota;
- IDE (*Identifier Extension*): Se esse *bit* for dominante o identificador da mensagem é de onze *bits*, configurando o formato padrão CAN 2.0 A;
- R0: *Bit* reservado para futuras variações da rede CAN;
- DLC (*Data Length Code*): São quatro *bits* que indicam o número de *bytes* do campo de dados;
- Campo de dados (*Data Field*): Carregam toda a informação transmitida até oito *bytes* (64 bits);
- CRC (*Cyclic Redundancy Check*): Detecta os erros e é composto por dezesseis *bits*, sendo que quinze *bits* são destinados ao código de detecção de falhas e um *bit* recessivo é destinado ao delimitador do campo;

- ACK (*Acknowledge*): São dois *bits* de dados, sendo um *bit* indica se a mensagem foi recebida corretamente no nó de destino e outro *bit* recessivo delimitador;
- EOF (*End of Frame*): São sete *bits* que indicam o fim da mensagem no barramento CAN;
- IFS (*Inter-Frame Space*): É um campo de sete *bits* que marcam um espaço entre duas mensagens.

### 2.3.3 Conexão física com o barramento CAN

De forma que um microcontrolador possa interpretar um pacote de dados do barramento CAN, necessita-se uma conexão física que interligue o barramento e o microcontrolador. Essa ligação está prevista na norma ISO-11898, como pode ser visto na Figura 8 (ISO-11898, 2016).

Figura 8 – Conexão física CAN norma ISO-11898



Fonte: Adaptado ISO-11898 (2016)

### 2.3.4 Transceptor CAN

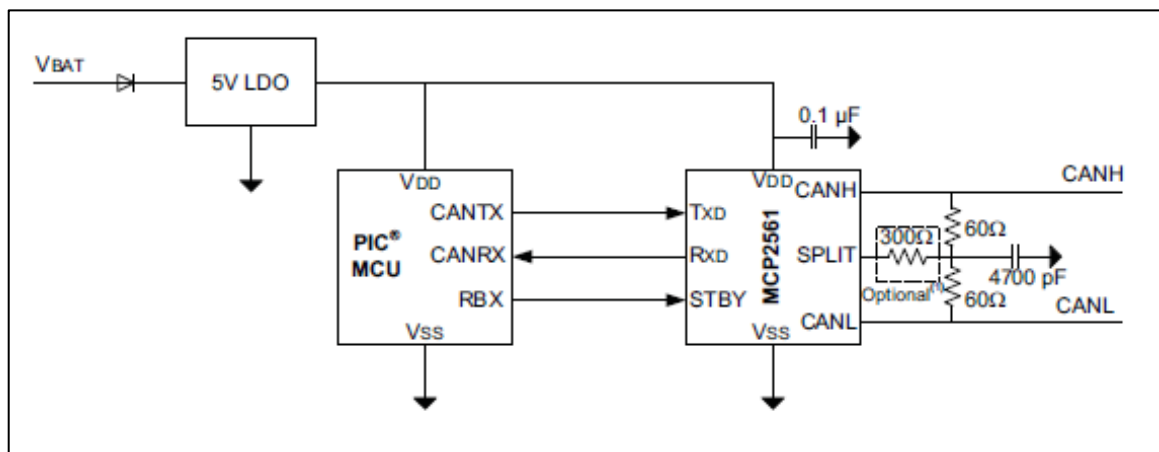
O transceptor é o componente responsável pela interação física entre o barramento e o controlador CAN. A função desse circuito é transformar a diferença das tensões de entrada, CAN-H e CAN-L, em sinais digitais de nível lógico 0 e 1. No processo de transmissão

de uma mensagem, o controlador emite dados seriais para a entrada lógica do transceptor. Já no processo de recebimento, o transceptor recebe estados dominantes ou recessivos nos pinos CAN-H e CAN-L. As informações são enviadas *bit a bit* para o controlador interpretar a sequência de acordo com o protocolo CAN (MICROCHIP, 2014).

### 2.3.5 Controlador CAN

O controlador é o componente responsável por decodificar os sinais digitais enviados pelo transceptor. O controlador envia e recebe mensagens nas especificações do protocolo CAN, interpretando o formato de mensagem (CAN 2.0 A ou CAN 2.0 B) e transmitindo para o microcontrolador (MICROCHIP, 2019). Na Figura 9 pode ser visto um exemplo de aplicação típica do controlador junto ao transceptor.

Figura 9 – Aplicação típica do controlador junto ao transceptor



Fonte: Adaptado de Microchip (2019)

## 2.4 REDE WIRELESS

As redes sem fio surgiram considerando um sistema simples de funcionamento de pequenos sistemas de comunicação de dados que permitiam a comunicação entre os dispositivos, posteriormente dando origem a WPAN (*Wireless Personal Area Networks*), ou redes pessoais sem fio. O conceito de rede WPAN não define um ambiente fixo, especifica qualquer tipo de rede sem fio operando em área restrita ou privada, com cobertura limitada, interligando equipamentos eletrônicos na rede. As redes sem fio apresentam baixo custo de operação, menor consumo, viabilizando a interconexão dos dispositivos e operando em bandas específicas de frequência (SILVA, 2021).

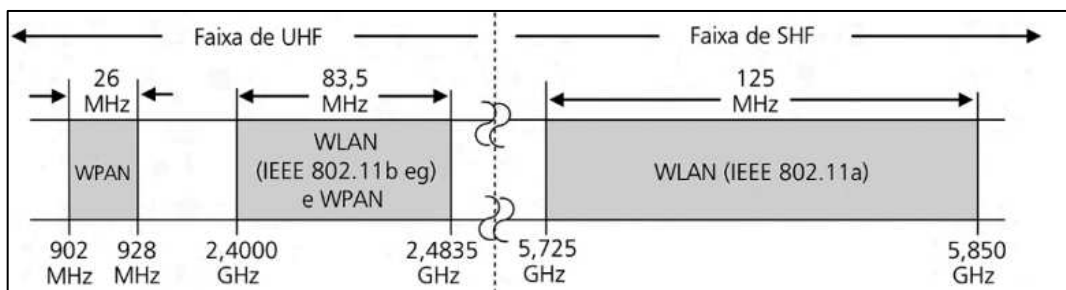
### 2.4.1 Wireless fidelity

De acordo com Moraes (2020), a rede Wi-Fi (*Wireless Fidelity*) é um consórcio criado pela WECA (*Wireless Ethernet Compatibility Alliance*), que realiza testes de confiabilidade e interoperabilidade com dispositivos aderentes a esse padrão. Porém, no Brasil, uso do termo Wi-Fi está associado como sinônimo de LAN sem fio. A rede sem fio é um sistema de comunicação que combina conectividade de dados através da tecnologia de radiofrequência, o sinal é transmitido através do espaço que o cerca. As redes são classificadas de acordo com a faixa de largura de banda listadas abaixo:

- 900 MHz, com uma largura de banda de 26 MHz;
- 2,4 GHz, com uma largura de banda de 83,5 MHz;
- 5 GHz, com uma largura de banda de 125 MHz.

As definições de faixa variam em um intervalo de banda ISM (*Industrial Scientific Medical*) específico, uma das quais ficam na faixa de UHF (*Ultra-High Frequency*) e a outra na faixa de SHF (*Super-High Frequency*), como pode ser visto na Figura 10. O alcance de uma rede wireless em ambiente *indoor* varia no raio de 35 a 100 metros (MORAES, 2020).

Figura 10 – Bandas ISM utilizadas nos espectros UHF e SHF

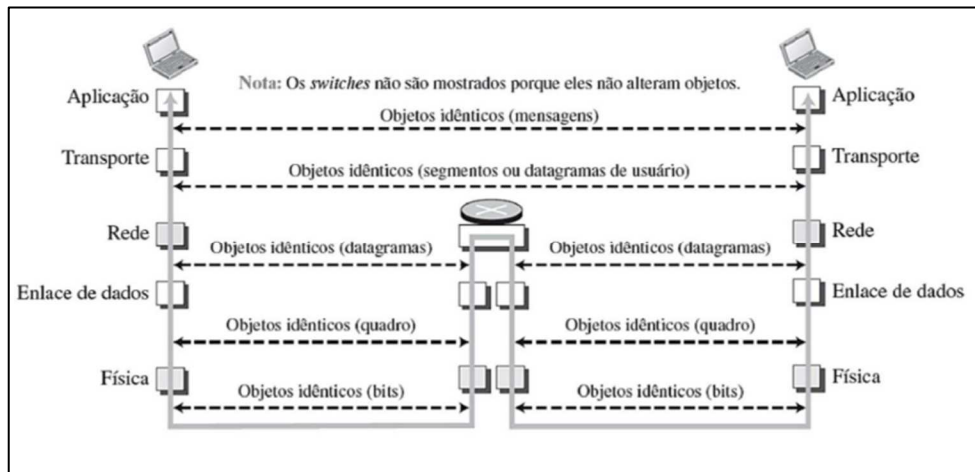


Fonte: Adaptado de Silva (2021).

### 2.4.2 Protocolo TCP/IP

O TCP/IP (*Transmission Control Protocol/Internet Protocol*), consiste em um conjunto de protocolos organizados em diferentes camadas usados na internet. É um protocolo hierárquico composto de módulos interativos com uma funcionalidade específica. Isso significa que cada protocolo do nível superior é apoiado pelos serviços fornecidos por um ou mais protocolos dos níveis inferiores. Originalmente a pilha de protocolo era formada por um modelo de quatro camadas. Atualmente, é formada por um modelo de cinco camadas (FOROUZAN, 2013). Na Figura 11 pode ser visto o modelo de cinco camadas.

Figura 11 – Camadas do protocolo TCP/IP



Fonte: Adaptado de Forouzan (2013)

### 2.4.3 Serviço cliente servidor

O cliente-servidor é um serviço que pode ser acessado através de um navegador e é distribuído entre muitos locais, conhecidos como *sites*. Cada *site* possui um ou mais documentos, chamados de páginas *Web*, onde algumas informações, ligações ou links para outras páginas do mesmo ou de outros *sites*. Uma página *Web* simples é aquela que não possui links de acesso a outras páginas, já a composta é aquela que tem um ou mais links de acesso (FOROUZAN, 2013).

De forma a distinguir uma página *Web* das outras, são necessários quatro identificadores, protocolo, *host*, porta e caminho. O protocolo HTTP define o formato das mensagens de pedido e resposta (FOROUZAN, 2013).

### 2.4.4 Sistemas em *Cloud*

Sistema em *cloud* (computação em nuvem) corresponde a uma arquitetura de computação que utiliza a internet para disponibilizar sistemas e aplicações de forma integrada. Um sistema na nuvem não inclui apenas a infraestrutura, mas também serviços incorporados, o que facilita o trabalho das equipes de empresas. Pode oferecer o modelo de infraestrutura privada e dedicada a um único cliente. Uma das principais vantagens desse modelo é que os serviços e os dados podem ser acessados de qualquer lugar (MORAES, 2020).

O objetivo é possuir uma infraestrutura de máquinas virtuais na nuvem, de forma que seja parametrizada igualmente a uma local, onde possa ser selecionando o tipo de CPU, a capacidade de memória, a interface de rede e a configuração de sistema (MORAES, 2020).

## 2.5 TRABALHOS CORRELATOS

De forma a analisar e compreender a proposta do projeto, foi realizada uma pesquisa a partir de artigos, trabalhos de conclusão de curso na área das engenharias. Desta forma, é possível avaliar as soluções propostas sobre a área de coleta de dados utilizando a rede CAN.

### 2.5.1 CAN automotivo sistema de monitoramento

O desenvolvimento do projeto de mestrado de Marques (2004), consiste em um sistema para comunicação de um computador com uma ECU automotiva através de rede CAN. Para o desenvolvimento do projeto utilizou-se o kit de MCP2510 da Microchip, este possui uma estrutura de *hardware* que possibilita o uso de um barramento CAN simulado ou real. O kit da disponibilizado pela Microchip na época, realizava a interface entre o computador e o microcontrolador, ou microcontrolador e o barramento CAN. Este kit de desenvolvimento, possuía filtros para os registradores que realizavam a recepção dos dados, que eram capazes de realizar a seleção das mensagens desejadas e evitava o superdimensionamento do microcontrolador.

Marques (2004), desenvolveu uma interface gráfica em linguagem de programação VB (*Visual Basic*), para agilizar o processo da comunicação entre o computador e o microcontrolador. A interface gráfica possibilita a visualização dos dados coletados pelo sistema, também é possível realizar operações de escrita, de leitura de diversas informações do MCU-CAN, de leitura do registro de estado de forma a realizar a verificação do estado lógico de *bits*, após a ocorrência de um evento relacionado ao MCU-CAN.

Ainda nesse estudo, Marques (2004) montou dois protótipos com frequências de *clocks* diferentes (16 MHz e 20 MHz), para os testes preliminares e ajustes da taxa de amostragem. Este por sua vez substitui o kit de desenvolvimento da Microchip na parte de conexão entre o computador e a ECU do veículo. O sistema testado foi do veículo *FORD Fiesta*, que possui um conector OBD II e uma rede CAN interna.

Os resultados obtidos por Marques (2004) comprovaram que as informações monitoradas pelo sistema desenvolvido mostraram o mesmo comportamento de uma ferramenta comercial, como o scanner da Alfatest. A partir dos dados coletados em teste, foi possível obter os resultados de funcionamento de um atuador ou sensor de escolha. Uma base de dados foi criada, que pode ser utilizada para auxiliar a manutenção veicular. Dessa forma, é possível otimizar o tempo dispendido e os recursos financeiros durante uma manutenção em oficinas mecânicas.

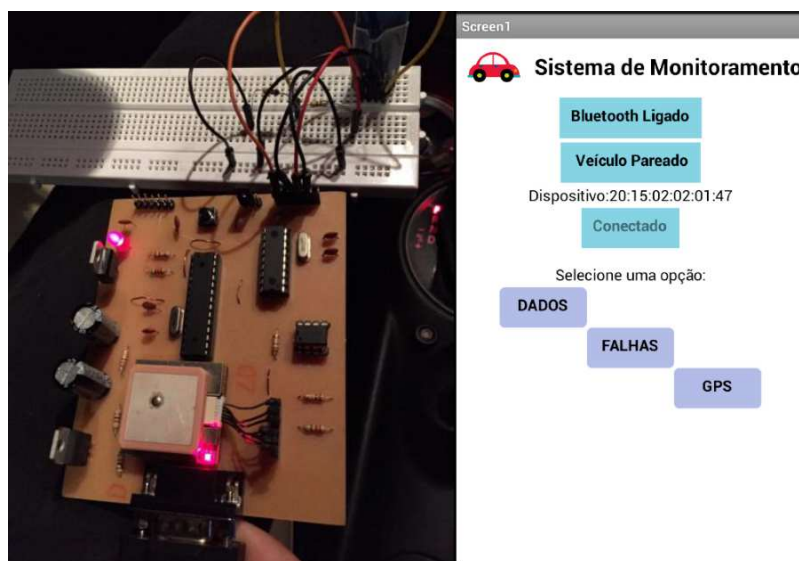
## 2.5.2 Sistema de monitoramento automotivo via rede CAN

O trabalho de conclusão de curso em engenharia mecatrônica, realizado por Cavalcante (2018), visa a implementação de ferramentas que aumentem a segurança dos motoristas a partir de um sistema de monitoramento automotivo. A obtenção das informações é feita através do conector OBD II via rede CAN, que monitora o veículo em tempo real e conta com localização geográfica.

Além do *hardware* de fabricação própria, Cavalcante (2018) também desenvolveu o aplicativo que registra os valores obtidos do veículo. Os valores adquiridos são transmitidos via *Bluetooth* para o telefone móvel do usuário. O protótipo utiliza o microcontrolador ATMEGA328p que possibilita o gerenciamento dos processos do sistema de monitoramento, comandando a transmissão das informações da rede CAN, além de controlar o funcionamento do GPS e do módulo *Bluetooth*. A autora sugere o uso a mudança de módulo de comunicação remota e um sistema que permita o armazenamento de dados na memória de um *hardware*, para realizar a análise dos dados posteriormente.

Os resultados do sistema de monitoramento desenvolvido por Cavalcante (2018), mostraram eficiência e precisão dos dados coletados em ambos os veículos que a autora testou. A placa protótipo desenvolvida apresentou com precisão os valores em tempo real, da velocidade e rotação do motor, mantendo a eficiência mesmo após longos períodos de teste. O aplicativo contribuiu para a utilização do sistema, pelo fato de ser uma ferramenta de acesso intuitivo e rápido, dispensando que o usuário possua conhecimento prévio sobre a mecânica do carro. Na Figura 12 podem ser vistos o *hardware* e o aplicativo desenvolvidos.

Figura 12 – *Hardware* protótipo e aplicativo



Fonte: Adaptado de Cavalcante (2018)



### 3 METODOLOGIA

Este capítulo apresenta a metodologia aplicada para o desenvolvimento inicial do dispositivo de aquisição de dados. As etapas deste capítulo foram separadas de acordo com a lógica de funcionamento do *hardware* e do *firmware*.

#### 3.1 FERRAMENTAS DE DESENVOLVIMENTO

Para desenvolvimento do *firmware*, foi optado pelo uso de um kit de desenvolvimento ESP32 fornecido pela Espressif e do *software Visual Studio Code* fornecido pela *Microsoft*. O uso da placa em conjunto com o *software* permite que o usuário possa agilizar o processo de desenvolvimento, testando a funcionalidade do conjunto, sem o *hardware* ser uma variável. Também permite que o projetista possa prever possíveis melhorias ou implementações que não foram previstas, antes de fabricar o *hardware* definitivo.

##### 3.1.1 *Software EasyEDA*

Para o desenvolvimento do hardware definitivo foi optado pelo uso do *software EasyEDA* que permite a realização do esquemático e do *layout* do PCB (*Printed Circuit Board*). Este *software* foi escolhido por ser uma ferramenta de interface intuitiva, possuir licença livre e ainda permite que o usuário possa fabricar seu projeto por um fornecedor parceiro do desenvolvedor do *software*.

##### 3.1.2 *Software Visual Studio Code*

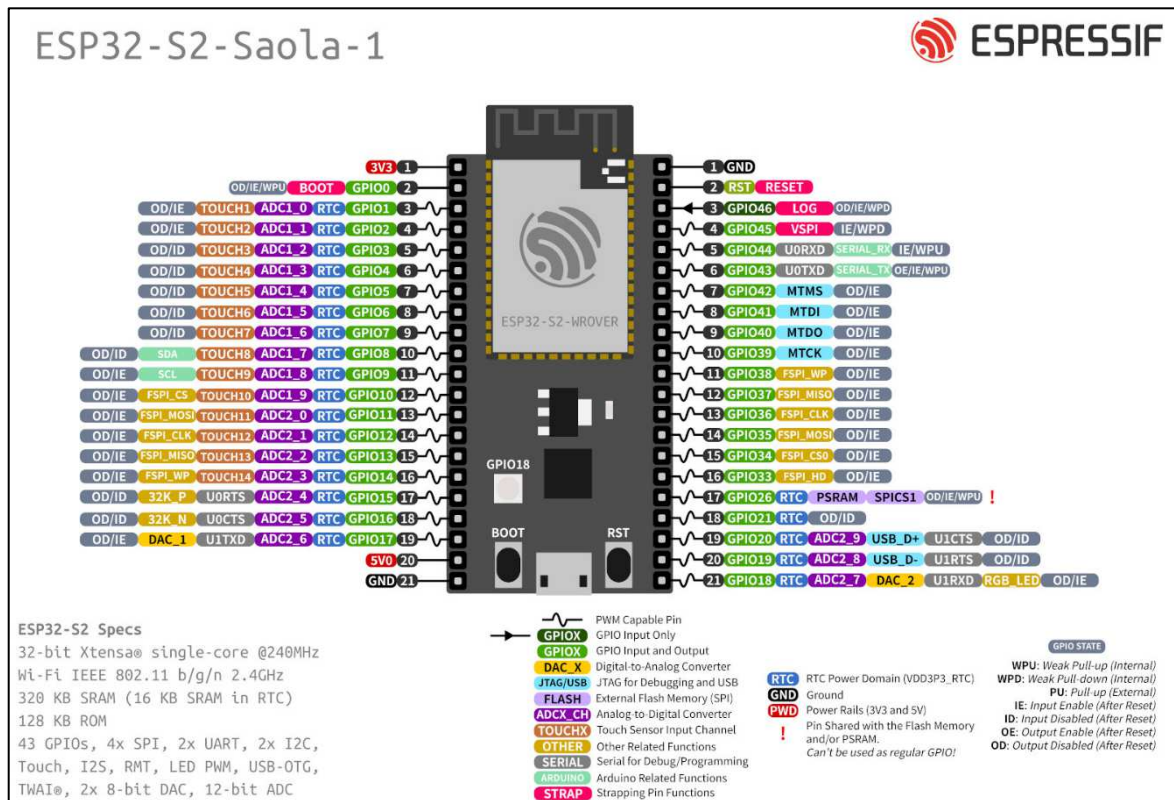
O *software Visual Studio Code* foi escolhido para o desenvolvimento do *firmware*, por possuir compatibilidade com a plataforma de desenvolvimento ESP32 e ser uma ferramenta disponibilizada de forma gratuita. Também outro ponto que foi importante para a escolha, foi o número de ferramentas e ajuda de suporte que o *software* disponibiliza de forma nativa.

##### 3.1.3 Plataforma de desenvolvimento ESP32

A plataforma de desenvolvimento ESP32 Saola, foi escolhida por possuir nativamente Wi-Fi 2,4 GHz protocolo TCP/IP e também por possuir uma ampla documentação, bibliotecas para realizar o desenvolvimento do *firmware*. As especificações

do MCU que equipa a plataforma de desenvolvimento podem ser vistas no Anexo A. Na Figura 13 pode ser visto o *layout* da plataforma ESP32.

Figura 13 – Plataforma de desenvolvimento ESP32



Fonte: Adaptado de Espressif (2023)

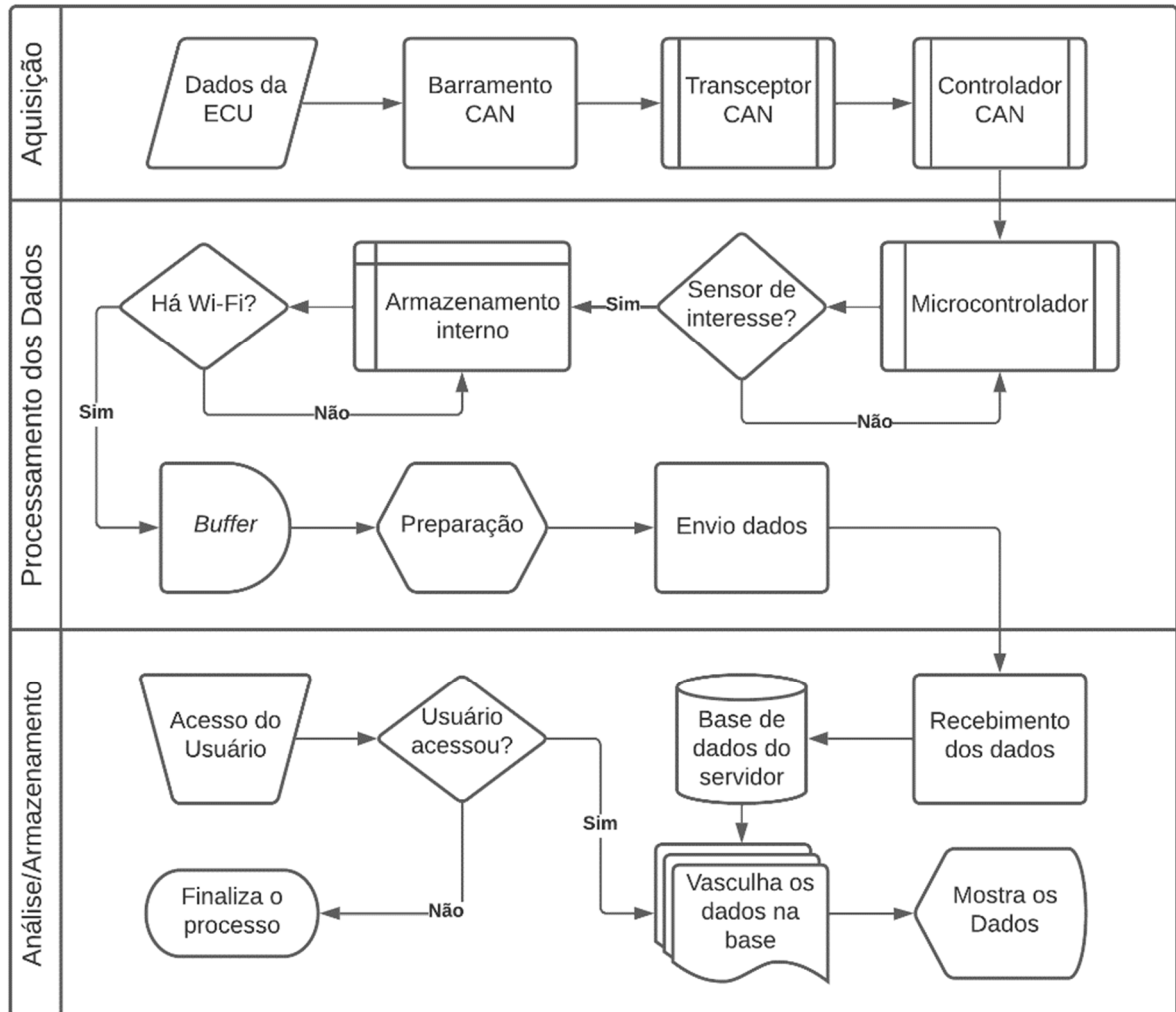
### 3.2 LÓGICA DE FUNCIONAMENTO

Para o desenvolvimento do projeto optou-se por separá-los em etapas, a fim de estabelecer uma conexão lógica e funcional entre as partes. Através de um fluxograma foi possível representar a arquitetura preliminar do dispositivo de aquisição de dados. Este fluxograma que pode ser visto na Figura 14.

A arquitetura do dispositivo de aquisição de dados foi separada em três etapas:

- **Aquisição de Dados:** O transceptor e o controlador CAN são os componentes responsáveis pela interação do dispositivo de aquisição de dados com a ECU do veículo.
- **Processamento:** Realizado por um microcontrolador responsável pelo tratamento, armazenamento e envio dos dados coletados.
- **Análise/Armazenamento na nuvem:** Os dados ficam disponíveis para uma análise em um servidor na nuvem.

Figura 14 – Arquitetura do dispositivo de aquisição de dados



Fonte: O autor (2023)

### 3.3 AQUISIÇÃO DE DADOS

A aquisição de dados é o processo macro onde é possível realizar a leitura dos parâmetros da ECU do veículo. De forma a facilitar o entendimento deste processo, foram separados em subprocessos listados abaixo:

- Protocolo de comunicação;
- Definição dos parâmetros;
- Aquisição dos parâmetros;
- Taxa de amostragem;
- Processamento e envio dos dados.

### 3.3.1 Protocolo de comunicação

Atualmente os veículos, fabricados no Brasil possuem o protocolo de comunicação SAE J2284 no formato de mensagem CAN 2.0 B, esses por sua vez são compatíveis com o microcontrolador utilizado. Um veículo com rede CAN e em conformidade com o conector OBD II permite o uso dos seguintes protocolos de comunicação:

- ISO 9141-2 (K-LINE);
- SAE J1850 PWM (Modulação da largura de pulso);
- ISO 14230-4 (Protocolo de palavra-chave 2000);
- SAE J2284;

### 3.3.2 Definição dos parâmetros

A ECU disponibiliza mais de 50 parâmetros de leitura em um veículo, para validar a integração do dispositivo de aquisição de dados foram definidos apenas três. A velocidade instantânea, a rotação do motor e a posição da borboleta de aceleração, foram os parâmetros definidos pelo autor. Estes são suficientes para validar a proposta, já que o método de requisição é o mesmo, variando somente o endereço específico do parâmetro desejado para a aquisição dos dados.

### 3.3.3 Aquisição dos parâmetros

Através do Serviço 01 (Exibição dos dados atuais) é possível solicitar os dados de diagnósticos do *powertrain*. Nesta seção é possível realizar a leitura dos valores de cada sensor disponível da ECU através dos PID, que são os identificadores dos parâmetros. Os sensores escolhidos a para realizar a leitura foram:

- Rotação do motor do veículo (PID-0C);
- Velocidade do veículo (PID-0D);
- Posição da borboleta de aceleração (PID-11).

A lista completa de endereços PID de uma ECU pode ser vista no Anexo B. De forma a obter as informações dos parâmetros do veículo, necessita-se enviar uma mensagem utilizando protocolo padrão OBD II. Na Tabela 1 pode ser visto um exemplo do formato de mensagem para a ECU.

Tabela 1 – Exemplo de pergunta e resposta a ECU

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
Definição	CAN ID	Data length	SID	PID	Data A	Data B	Data C	Data D	-
Pergunta	7DF	02	01	0C	FF	FF	FF	FF	FF
Resposta	7E8	02	01	0C	0F	51	FF	FF	FF

Fonte: O autor (2023)

No exemplo da Tabela 1, foi requisitado a ECU sobre a rotação do motor do veículo, com os valores obtidos da resposta é possível calcular a rotação em RPM. Os valores de rotação são convertidos para RPM utilizando a Equação 1.

$$\omega = \frac{(A * 256) + B}{4} \quad (01)$$

$$\omega = \frac{(15 * 256) + 51}{4} = 973 \text{ RPM}$$

Onde:

$\omega$  = Rotação do motor (RPM);

$A$  = Dados do *Byte* 3;

$B$  = Dados do *Byte* 4;

A velocidade instantânea do veículo não necessitou de conversão para obtenção do valor. Este foi obtido de forma direta a partir da Equação 2.

$$V = A \quad (02)$$

Onde:

$V$  = Velocidade (km/h);

$A$  = Dados do *Byte* 3;

A unidade de medida da velocidade instantânea do veículo pode estar disponível no sistema métrico ou imperial. Esta informação depende do modelo, fabricante ou região onde veículo é comercializado, podendo possuir ambos os sistemas de medidas disponíveis.

A posição da borboleta de aceleração é a medida de abertura dela em relação ao corpo onde está instalada o TBI (*Throttle Body Fuel Injection*). Essa abertura é diretamente proporcional a carga que é solicitada ao motor do veículo. Por exemplo ao dirigir em uma pista plana a posição da borboleta de aceleração fica entorno de 30 % de sua abertura total, porém

quando o veículo está subindo uma pista com elevação, esse valor pode ultrapassar os 50 % dependendo da inclinação da pista. Este valor foi convertido na porcentagem da abertura utilizando a Equação 3.

$$TPS = \frac{A * 100}{255} \quad (03)$$

Onde:

$TPS = Throttle Position Sensor$  (%);

$A =$  Dados do *Byte 3*;

### 3.3.4 Taxa de Amostragem

A frequência de amostragem dos dados é um fator importante para a aquisição de dados, onde é possível definir a fidelidade do sinal coletado. Na rede CAN a taxa de transmissão depende do comprimento de barramento, de acordo com a norma ISO 11898 (2003), a taxa máxima corresponde a 1 Mbit/s para um tamanho de 40 metros (*High-speed CAN*), devido a essa definição foi adotada mesma taxa para o projeto.

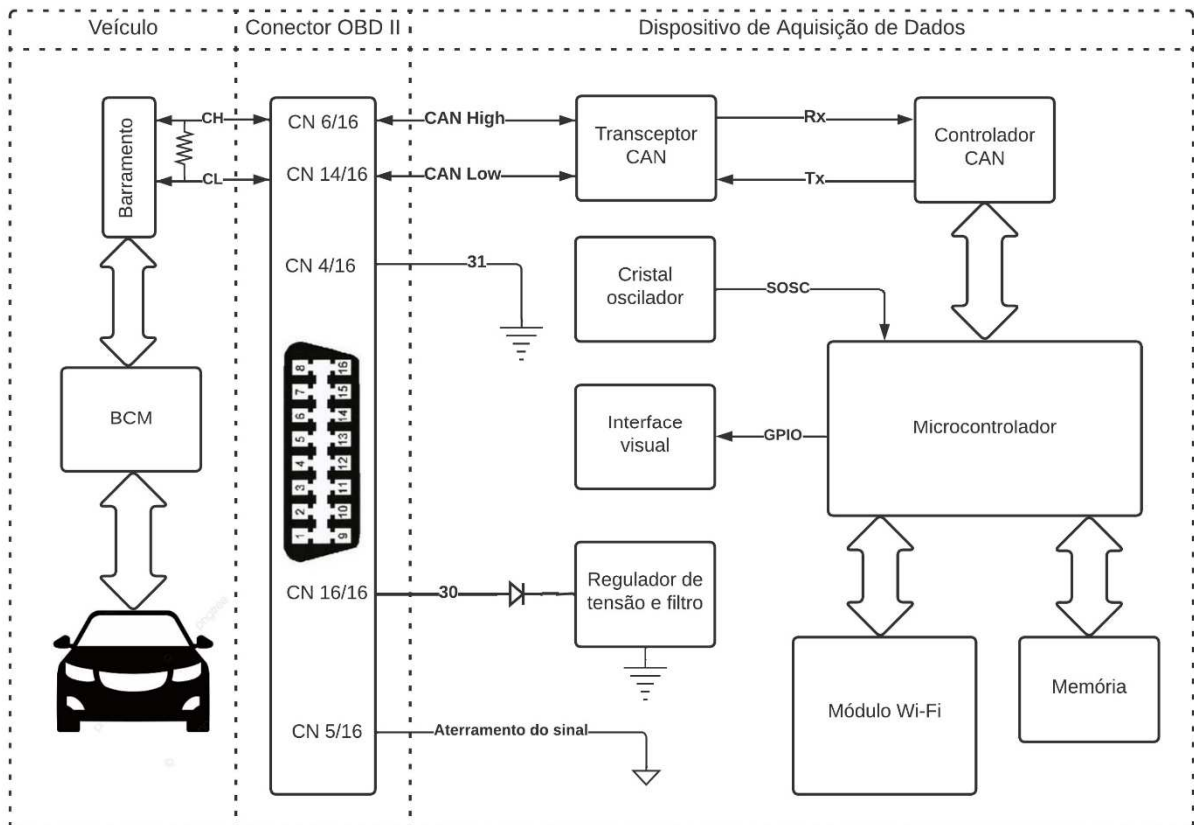
### 3.3.5 Processamento e envio dos dados

Para acessar os dados armazenados na ECU foi utilizado um transceptor, que é responsável pela ligação física entre o barramento e o controlador CAN. O transceptor e o controlador trabalham simultaneamente, enquanto o transceptor transforma as tensões da rede CAN em sinais lógicos, o controlador emite os dados para entrada lógica do transceptor decodificando os sinais recebidos e os enviando para o microcontrolador.

O microcontrolador escolhido para realizar as funções de processamento foi o ESP32-S2-Wroom-I, que executa as diferentes operações com os dados recebidos da ECU do veículo. Com estes dados é possível realizar as operações para conversão dos sinais a partir do algoritmo (*firmware*), da mesma forma que foi realizada na Equação 1 citada anteriormente e armazena os dados adquiridos em memória.

O módulo realiza a busca de sinal de rede disponível, quando uma rede Wi-Fi é encontrada os dados armazenados em memória, são enviados para o servidor na nuvem. Os dados precisam ser formatados (pacote de dados, informações de segurança dispositivo) para serem enviados por Wi-Fi, preservando a integridade deles. Na Figura 15 podem ser vistos os periféricos que compõe o dispositivo de aquisição de dados.

Figura 15 – Periféricos do dispositivo de aquisição de dados



Fonte: O autor (2023)

### 3.4 TESTES FUNCIONAIS

O teste funcional é a etapa de verificação dos objetivos estabelecidos, onde as necessidades do projeto são comparadas com os resultados obtidos. Esses testes podem validar ou refutar, os conceitos e métodos adotados. A validação do estudo consiste em um teste onde as informações coletadas são fidedignas ao produto.

As etapas para a execução dos testes estão separadas em:

- Teste de bancada: Durante este processo são realizados os testes de *firmware* e *hardware* com a placa de desenvolvimento que consiste em verificar o funcionamento e depurar o código.
- Teste de validação: Após o processo anterior, os testes de validação foram realizados com o *hardware* final desenvolvido. Esta etapa consiste em verificar a integração do dispositivo de aquisição de dados com o veículo e com o servidor.

## 4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento do dispositivo de aquisição de dados. As etapas deste capítulo foram separadas de acordo com o *hardware* proposto com o memorial de cálculos, o *hardware* utilizado (plataforma de desenvolvimento) e o *firmware* desenvolvido.

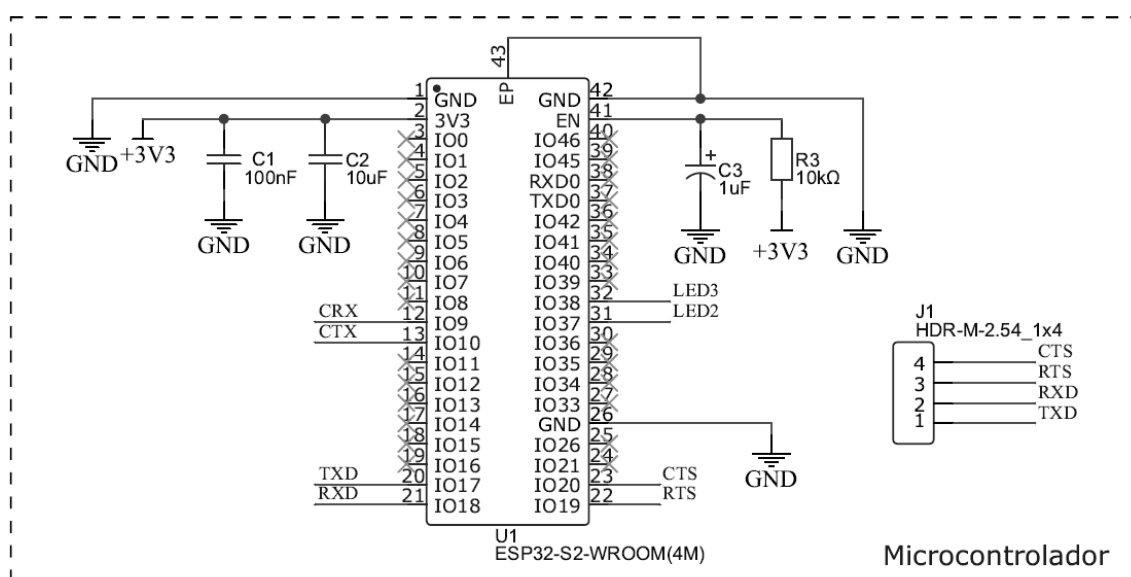
### 4.1 ARQUITETURA DO *HARDWARE* PROPOSTO

O desenvolvimento do *hardware* proposto foi separado de acordo com as funções dos circuitos e periféricos que o compõe. O intuito dessa parte é descrever como o dispositivo de aquisição de dados realiza a operação entre o veículo e o servidor, sem a utilização de plataformas pré-prontas de desenvolvimento.

#### 4.1.1 Circuito de processamento e transmissão dos dados

O circuito do microcontrolador foi montado com as recomendações descritas no *datasheet* da Espressif (2023), apenas foram realizadas as trilhas de conexão para os periféricos e para a comunicação TWAI (*Two-Wire Automotive Interface*), que é o nome específico que o fabricante denomina a rede CAN. A parte de transmissão dos dados é realizada pelo circuito TWAI conectado ao transceptor CAN e a transmissão dos dados Wi-Fi é realizada pela antena disponível junto ao microcontrolador. Na Figura 16 pode ser visto o circuito simplificado do microcontrolador.

Figura 16 – Circuito do microcontrolador



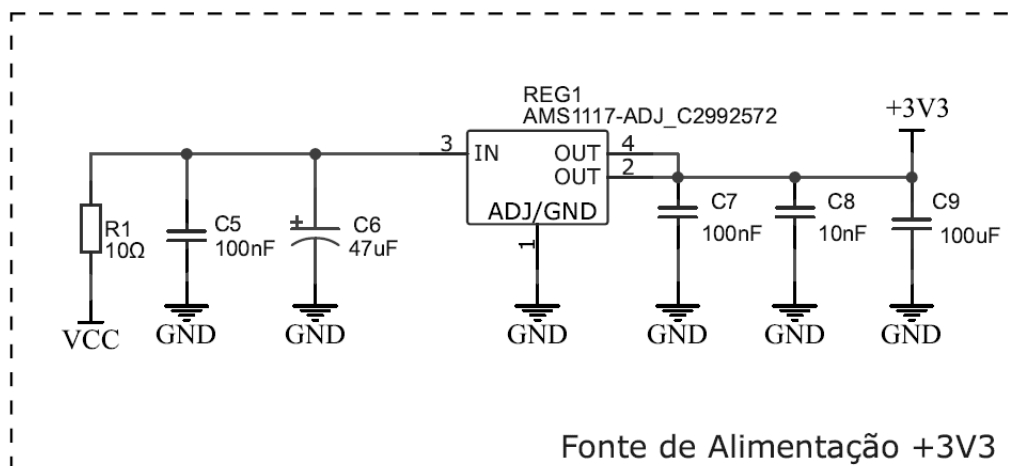
Fonte: O autor (2023)



### 4.1.2 Circuito de alimentação

Nesta etapa de desenvolvimento, realizou-se o projeto de uma fonte de tensão de 3,3 V para alimentar o microcontrolador e o transceptor CAN. Foi optado pelo uso de um regulador de tensão do tipo fixo com saída pré-definida, já que a potência dissipada pelo circuito é inferior a 2 W. Na Figura 17 pode ser visto o circuito de alimentação 3,3 V.

Figura 17 – Circuito de alimentação 3,3 V



Fonte: O autor (2023)

A tensão de operação do sistema automotivo pode variar de 11,6 V quando o motor está desligado, à 14,5 V quando o motor está em funcionamento. Essa diferença de 20 % de tensão foi adotada para realizar o dimensionamento dos capacitores de filtragem. A corrente máxima adotada foi e 0,8 A (limite do diodo escolhido), sendo que a corrente nominal é de 0,1 A. Os valores dos capacitores foram obtidos utilizando a Equação 4.

$$C = \frac{I_{out}}{2 * f * V_r} \quad (04)$$

Onde:

$C$  = Capacitância (F);

$I_{out}$  = Corrente de saída (A);

$f$  = Frequência (Hz);

$V_r$  = Tensão de *ripple* (V);

A partir da equação acima, pode-se calcular os capacitores para filtrar as faixas de frequência desejadas. Na Tabela 2 pode ser visto os capacitores que compõe a fonte de tensão.

Tabela 2 – Capacitores de filtragem da fonte de tensão

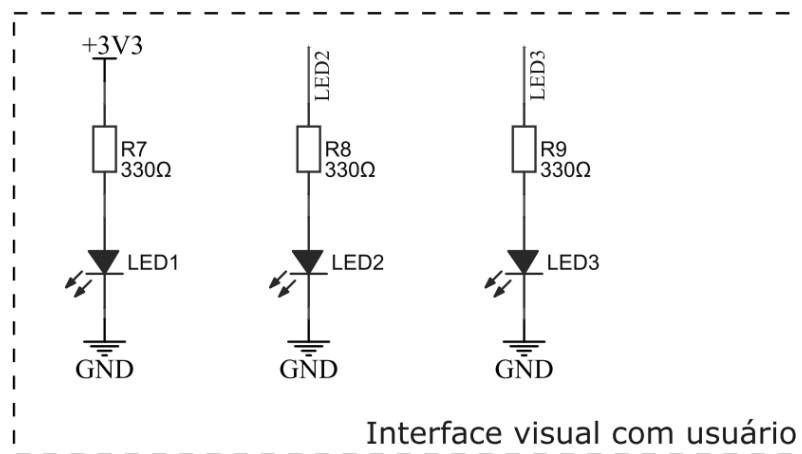
$I_{out}$ (A)	freq (Hz)	$V_r$ (V)	C (uF)
0,8	1379,3	2,9	100
0,8	2934,7	2,9	47
0,8	1,38E+06	2,9	0,10
0,8	1,38E+07	2,9	0,01

Fonte: O autor (2023)

#### 4.1.3 Circuito de interface visual

A parte de *feedback* ao usuário é realizada por 3 leds (*Light Emitting Diode*) com o intuito de mostrar os estados de funcionamento. O primeiro led indica se há alimentação no microcontrolador, o segundo se há comunicação com o veículo e o último se há uma rede Wi-Fi conectada. Na Figura 18 pode ser visto o circuito de interface visual com o usuário.

Figura 18 – Circuito interface visual com o usuário



Fonte: O autor (2023)

#### 4.1.4 Circuito de comunicação com o veículo

De forma a estabelecer uma conexão entre o veículo e o dispositivo de aquisição de dados, foi necessário utilizar um transceptor CAN, já que o microcontrolador possui compatibilidade com rede CAN. O transceptor SN65HVD230 realiza a conexão com o barramento CAN do veículo e envia as informações coletadas para o microcontrolador, no qual realiza o processamento.

As proteções na linha de conexão com a porta OBD II foram realizadas através de um resistor de 10 Ω e um termistor PTC (*Positive Temperature Coefficient*) de 0,1 A, para atuarem como um protetor térmico em caso de curto-circuito nas linhas CAN *High* e CAN *Low*.

Consultando os valores do componente no *datasheet* e aplicando-os na Equação 5, obteve-se o valor do resistor de referência.

$$R = \frac{D * (T_{cp} - T_a)}{I_a^2} \quad (05)$$

$$R = \frac{0,183 * 10^{-3} * (570 - 25)}{0,1^2} = 10 \Omega$$

Onde:

$D$  = Constante de dissipação (mW/°C);

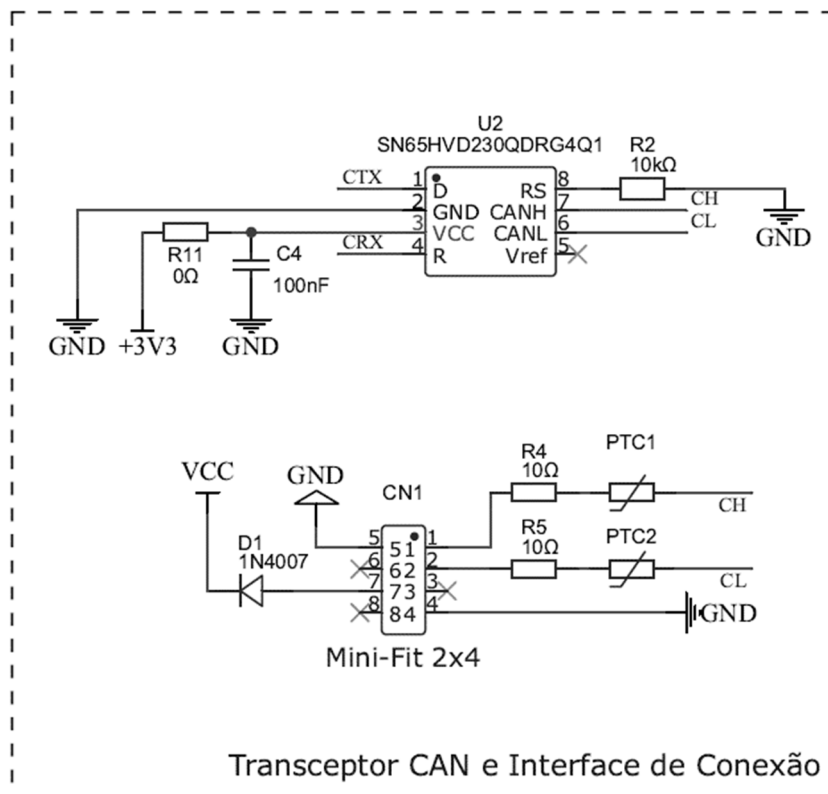
$I_a$  = Corrente (A);

$T_{cp}$  = Temperatura de *Curie* (°C);

$T_a$  = Temperatura ambiente (°C).

Para proteger a entrada das fontes de tensão foi optado pelo uso de um diodo 1N4007 (D1) para evitar a inversão de polaridade. Esse modelo de diodo suporta uma tensão reversa de até 1000 V. Na Figura 19 pode ser visto o circuito de comunicação com o veículo.

Figura 19 – Circuito de comunicação com o veículo



Fonte: O autor (2023)

#### 4.1.5 Placa de circuito impresso

A placa de circuito impresso foi desenvolvida a partir dos esquemáticos desenhados no *software EasyEDA*. O *software* gera um *layout* de interligação entre os componentes distribuídos ao longo da placa, aproveitando o espaço disponível e facilitando as conexões.

Os parâmetros adotados para o projeto do PCB, foram a largura de trilha mínima de 30 mils (milésimos de polegada) para a de alimentação, 25 mils para as trilhas de sinal e a espessura de trilha adotada foi de 1 mil. Para os cálculos foi utilizada a norma IPC-2221 (*Institute for Printed Circuits*, 2003) como referência, a corrente máxima por trilha foi obtida a partir da Equação 6.

$$I = k * \Delta T^b * A^c \quad (06)$$

Onde:

$I$  = Corrente (A);

$\Delta T$  = Elevação de Temperatura (°C);

$A$  = Área de secção da trilha (mils<sup>2</sup>);

$b, c, k$  = Constantes para trilhas externas.

A partir da equação acima, pode-se calcular os valores de capacidade de corrente por trilha em cada segmento do circuito. Na Tabela 3 podem ser vistos os valores de capacidade de corrente por trilha.

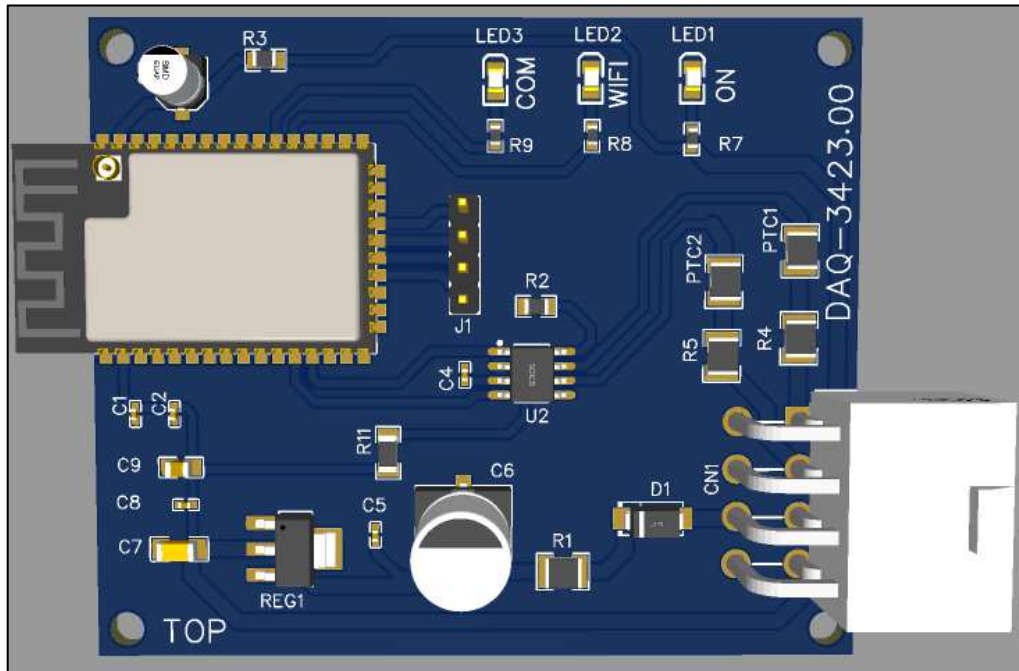
Tabela 3 – Capacidade de corrente por trilha

Trilha	Larg. (mils)	Esp. (mils)	Temp. (°C)	I (A)
Vcc	50	1	40	4,15
Gnd	50	1	40	4,15
Com	25	1	40	2,51
IO	25	1	40	2,51

Fonte: O autor (2023)

Os valores de corrente obtidas nos cálculos, validam a escolha da largura das trilhas de alimentação e dos outros periféricos. Essa configuração resultou em uma placa de circuito impresso de dimensões 56 x 67 mm com trilhas apenas em uma das faces. Também se utilizou plano de terra ao longo da placa, para proteção de ruídos eletromagnéticos. Na Figura 20 pode ser visto o *layout* da placa de circuito impresso.

Figura 20 – Placa de circuito do dispositivo de aquisição de dados



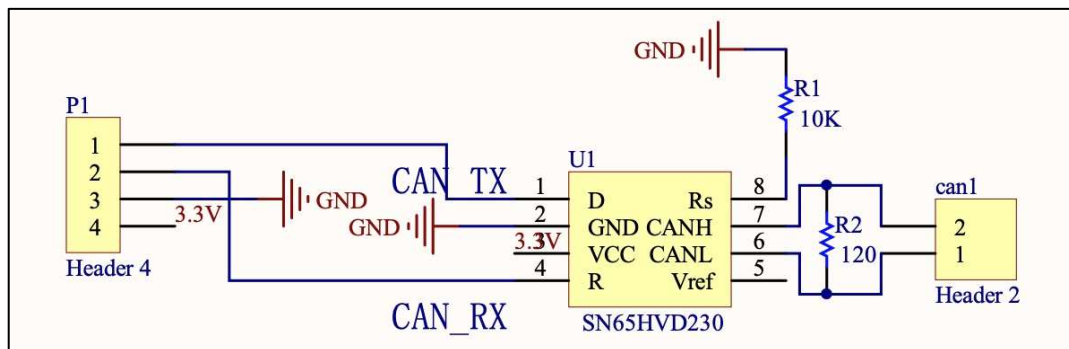
Fonte: O autor (2023)

#### 4.2 ARQUITETURA DO *HARDWARE* MODULAR

O desenvolvimento do *hardware* foi realizado a partir de módulos interligados disponíveis no mercado. O módulo CAN BUS utilizado, possui um transceptor SN65HVD230 que é o responsável por realizar a comunicação do veículo com o microcontrolador. Este módulo suporta o protocolo CAN 2.0 B e possui uma velocidade de comunicação de 1 Mbit/s.

A interface de ligação entre o módulo transceptor CAN e o microcontrolador foi realizada através da comunicação TWAI. Na Figura 21 pode ser visto o esquemático do módulo transceptor CAN.

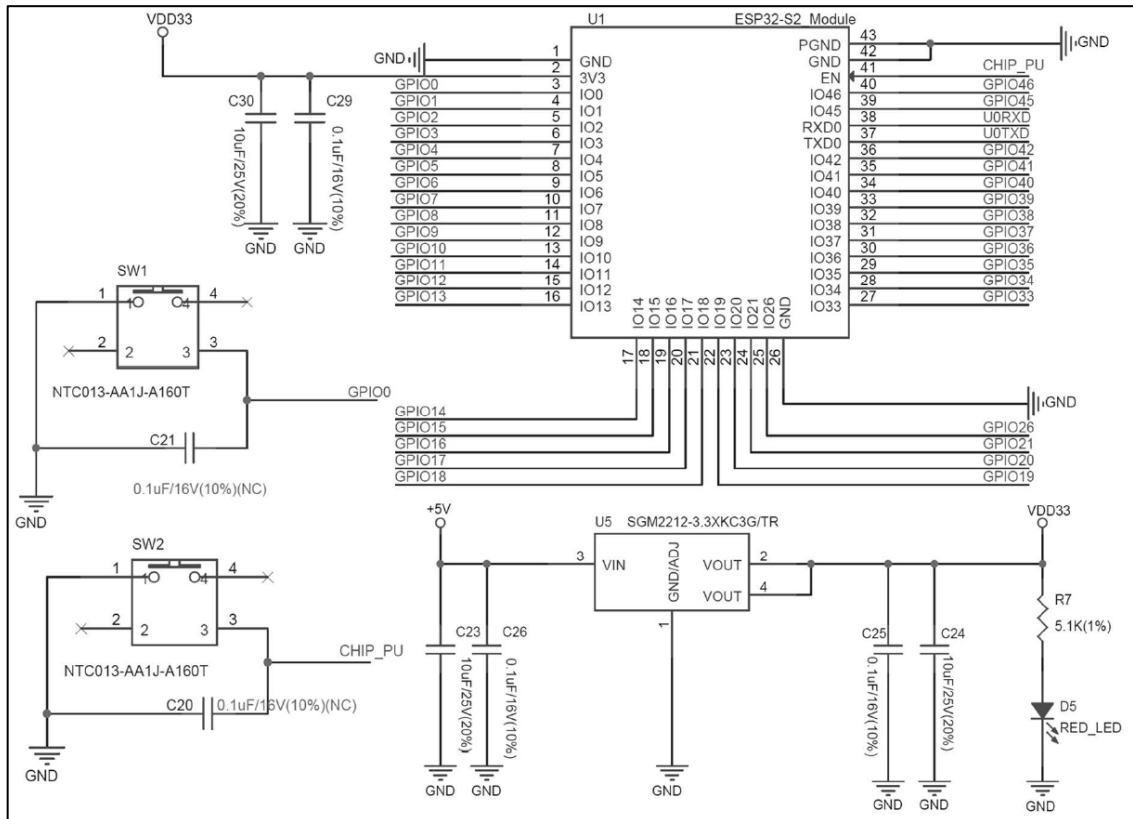
Figura 21 – Esquemático do módulo transceptor CAN



Fonte: Adaptado de Coperhilltech (2023)

O módulo responsável pelo processamento dos dados é a plataforma de desenvolvimento ESP32 Saola, que possui um microcontrolador, uma fonte de 3,3 V e uma antena impressa na placa de circuito. A plataforma de desenvolvimento deve ser alimentada com 5 V e essa fonte de tensão foi fabricada com seus devidos componentes em uma placa padrão. Na Figura 22 pode ser visto o esquemático do módulo ESP32 Saola.

Figura 22 – Esquemático do módulo ESP32 Saola

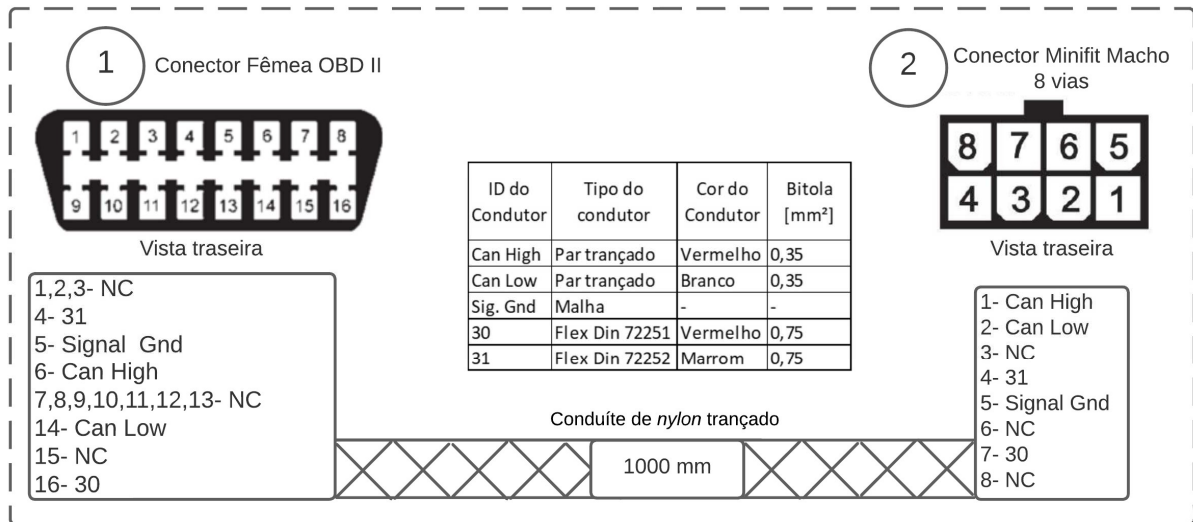


Fonte: Adaptado de Espressif (2023)

#### 4.2.1 Chicote de interface com o veículo

O chicote de interface com o veículo foi desenvolvido com condutores de cobre norma DIN 72251-6 (*Deutsches Institut für Normung, 2001*), que especifica a tensão e isolamento de condutores elétricos destinados a veículos automotores. Os condutores elétricos para alimentação são de seção 0,75 mm<sup>2</sup> do tipo flexível e os de comunicação são 0,35 mm<sup>2</sup> do tipo par trançado envolto por malha de metal. Também foram escolhidos conectores que atendam a classificação UL94-V2 (*Underwriters Laboratories, 2023*), esses foram o *minifit* de 8 vias e conector padrão OBD II. Para isolamento suplementar foi utilizado um conduíte de *nylon* trançado que também atende a classificação UL94-V2. Na Figura 23 pode ser visto o diagrama de construção do chicote de interface com veículo.

Figura 23 – Diagrama de montagem do chicote



Fonte: O autor (2023)

#### 4.2.2 Gabinete de proteção do PCB

A placa de circuito impresso do dispositivo de aquisição de dados foi montada em um gabinete plástico padrão, o CP-018 fornecido pela empresa Patola. Este gabinete possui uma abertura frontal, onde é possível remover a tampa e inserir a placa nos trilhos laterais do gabinete, sem a necessidade do uso de grampos ou parafusos para a fixação. Essa escolha de montar a placa modular em um gabinete plástico deve-se ao fato de ter uma imagem de um produto final. Na Figura 24 pode ser visto o dispositivo de aquisição de dados.

Figura 24 – Dispositivo de aquisição de dados



Fonte: O autor (2023)

### 4.3 DESENVOLVIMENTO DO *FIRMWARE*

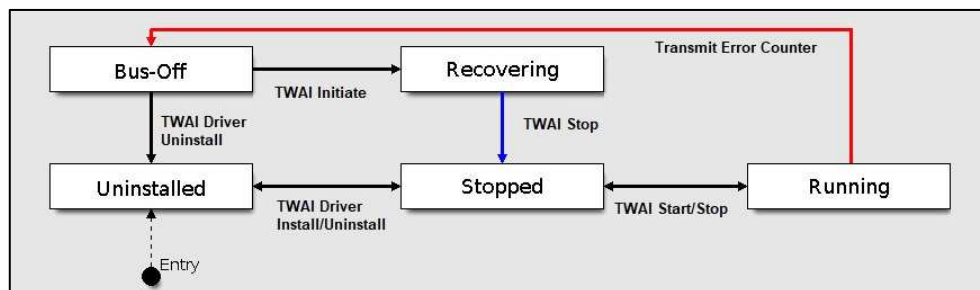
O desenvolvimento do *firmware*, foi realizado no *hardware* modular que consiste na junção dos módulos EPS32 Saola e do transceptor CAN. Dessa forma, o *firmware* desenvolvido e pode ser utilizado no *hardware* definitivo sem necessitar alterações. O desenvolvimento do *firmware*, foi dividido em etapas para melhor entendimento:

- Comunicação da rede CAN, com a ECU do veículo;
- Comunicação da rede Wi-Fi com servidor.

#### 4.3.1 Rede CAN

O desenvolvimento do *firmware* da rede CAN com o veículo, foi realizado a partir da inclusão das bibliotecas necessárias para as funções TWAI e GPIO (*General Purpose Input/Output*). A TWAI possui uma operação com estados distintos e regras rígidas em relação às funções ou condições, que desencadeiam uma transição de cada estado. Na Figura 25 pode ser visto o diagrama de estados e transições TWAI.

Figura 25 – Diagrama de estados e transições TWAI



Fonte: Adaptado de Espressif (2023)

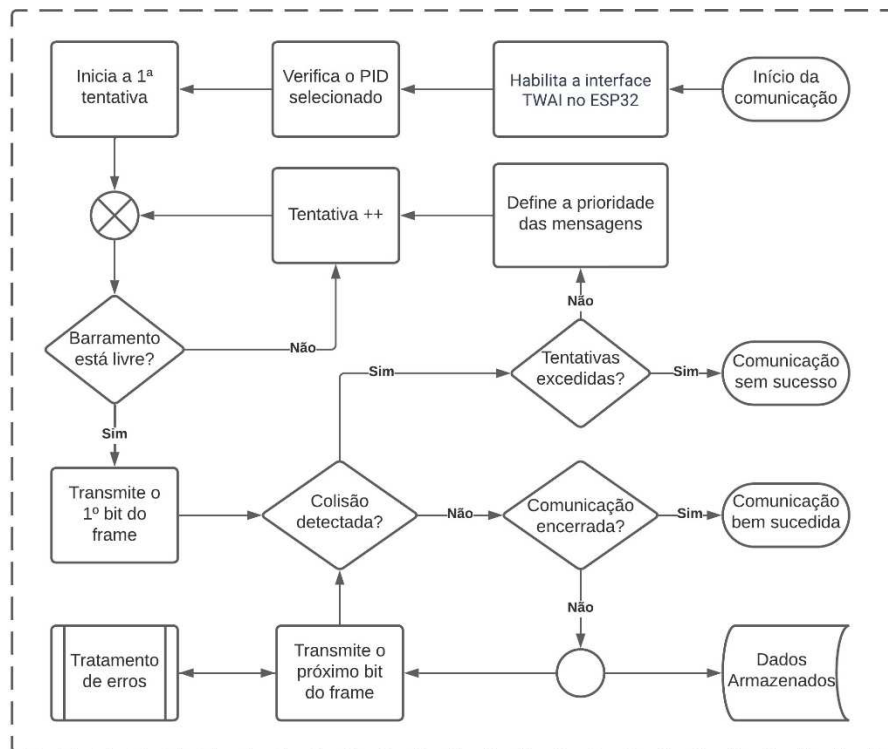
A configuração inicial do *firmware* foi realizada incluindo cada função de estado descritas anteriormente em suas devidas posições. A função de instalação dos *drivers* (*twai\_driver\_install*) é responsável por realizar a configuração das estruturas e da macro iniciais para a comunicação CAN. A função para iniciar os *drivers* de operação (*twai\_start*) é responsável por permitir que as filas de recepção e transmissão sejam limpas, eliminando todas as mensagens não lidas ou com transmissão pendente. As funções responsáveis pela transmissão (*twai\_transmit*) e recepção (*twai\_receive*) das mensagens, são configuradas a partir de uma função que formata o tipo de mensagem a ser transmitida (*twai\_message\_t*), isso inclui o comprimento de dados, os identificadores de mensagem e as estruturas necessárias para o funcionamento.



A configuração das funções para encerrar as operações da TWAI (*twai\_stop*), é uma função na qual impede que uma mensagem seja transmitida ou recebida até que (*twai\_start*) seja chamado. Essa função só pode ser chamada quando o driver TWAI está no estado de execução. A função de desinstalação dos *drivers* (*twai\_driver\_uninstall*), é responsável por liberar a memória utilizada pelo *driver*, essa função só pode ser chamada nos estados *stop* ou *bus-off*.

A aquisição de dados ocorre quando o código chama repetidamente a função de processamento de dados (*ProcessData*), que solicita e lê informações do veículo. Os valores dos parâmetros de rotação, velocidade e posição da borboleta de aceleração são lidos e armazenados em um vetor. Na próxima etapa, a função (*RequestData*), é a responsável por realizar a requisição dos identificadores PID (pergunta para a ECU), além de chamar a função de leitura das mensagens (resposta da ECU). A função de leitura das mensagens (*ReadObdMessage*) é responsável por realizar a parte aritmética dos dados obtidos da ECU, onde são convertidos para as suas respectivas unidades de medida e armazenados nas variáveis. Caso não retorne o PID requisitado, a função de requisição é chamada novamente até receber o retorno da ECU. Ao receber retorna o valor pra função de processamento de dados que funciona como uma ligação entre a CAN e a função de envio de dados para o servidor na nuvem. Na Figura 26 pode ser visto um fluxograma de interação com rede CAN.

Figura 26 – Fluxograma do *firmware* da rede CAN

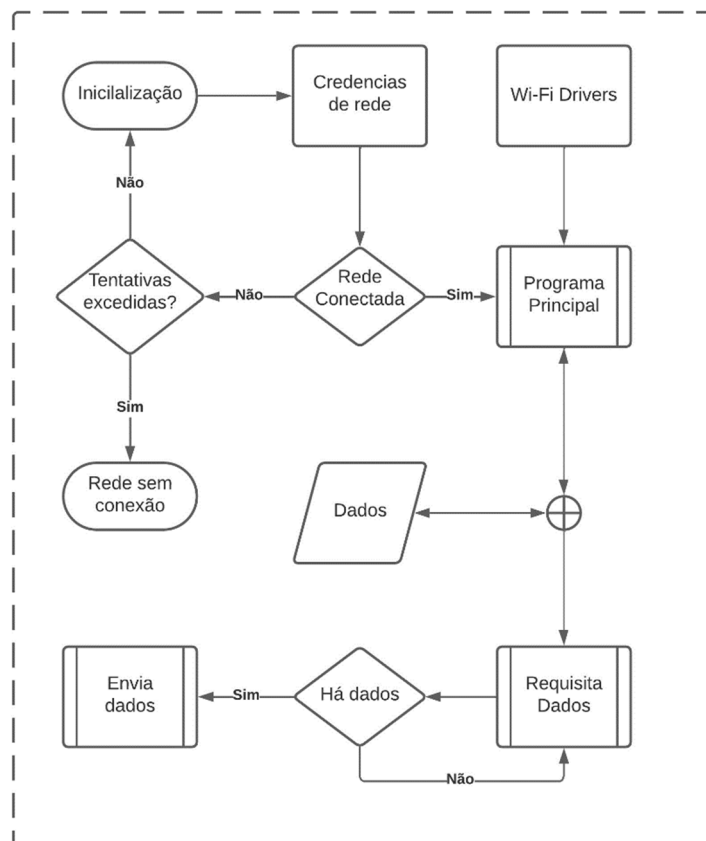


Fonte: O autor (2023)

### 4.3.2 Rede Wi-Fi

A função que realiza o envio dos dados (*send\_to\_cloud*) é responsável por receber os valores dos parâmetros coletados e estabelecer uma ligação do *hardware* com o servidor. Essa função é composta por um vetor de 300 posições que é utilizada para construir a URL (*Uniform Resource Locator*) que será utilizada no HTTP. A variável *scriptUrl* contém a URL do servidor remoto para onde os dados serão enviados e inclui os valores dos 3 dos parâmetros coletados. A estrutura de configuração é chamada e criada para configurar a solicitação HTTP. Isso inclui a URL (definida na etapa anterior), o método HTTP (*GET*) e o tipo de transporte (TCP). A solicitação HTTP é enviada para o servidor remoto, o resultado é armazenado em uma variável, se essa variável for igual a *ESP\_OK*, a solicitação foi bem-sucedida. Por fim, o cliente HTTP é encerrado e liberado. Na Figura 27 pode ser visto um fluxograma simplificado do *firmware* da rede Wi-Fi.

Figura 27 – Fluxograma do *firmware* Wi-Fi



Fonte: O autor (2023)

## 5 RESULTADOS E DISCUSSÕES

Este capítulo possui o objetivo de fornecer uma visão geral dos resultados e discussões relacionados ao dispositivo de aquisição de dados realizado neste estudo. Os resultados são apresentados conforme a realização dos processos durante o desenvolvimento, sendo separados por testes de *firmware*, de *hardware* e de aplicação.

### 5.1 TESTES DE *HARDWARE*

O primeiro teste consistiu em uma inspeção visual nos pontos de solda, verificando por pontos de falhas no processo de soldagem. Em seguida foi realizada uma inspeção com o multímetro com o intuito de verificar se havia alguma trilha ou componente em curto circuito.

O segundo teste foi o da fonte de tensão de 5 V, que alimenta a plataforma de desenvolvimento. Nesta etapa, verificou-se a saída de tensão com a fonte operando a vazio, após a confirmação do fornecimento dos 5 V, foram conectados os periféricos e verificados os valores de corrente, tensão e a temperatura no encapsulamento do regulador de tensão LM7805. O dispositivo permaneceu por 1 hora em teste onde nesse período a tensão de entrada foi variada de 11,5 V até 14,5 V, essa faixa representa as operações de partida e carga do veículo. A temperatura ambiente do teste foi de 22 °C, sendo que a temperatura no encapsulamento do regulador de tensão estabilizou em 37 °C resultando em uma elevação de 15 K, que é uma faixa de trabalho permitida do componente de acordo com o *datasheet* do fabricante. Na Figura 28 pode ser visto o teste realizado na bancada.

Figura 28 – Testes de funcionamento do *hardware*

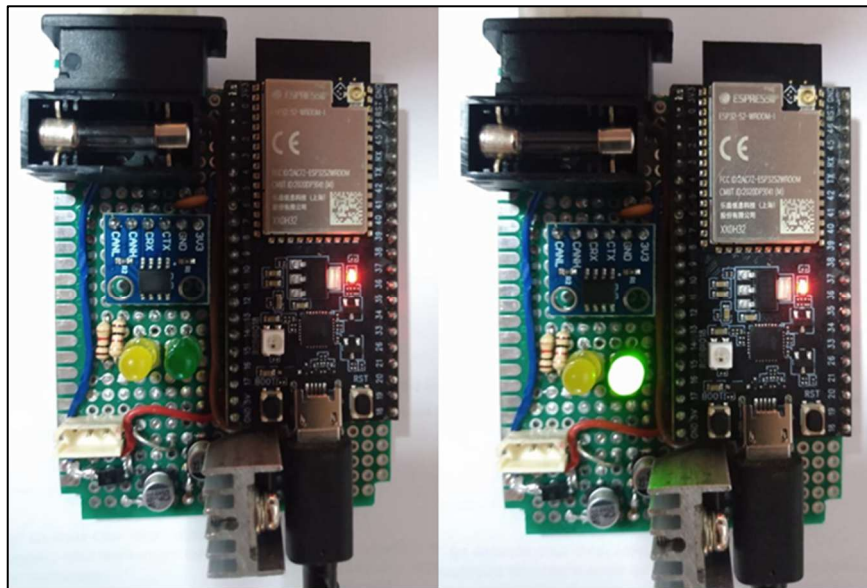


Fonte: O autor (2023)

## 5.2 TESTES DE *FIRMWARE*

Na primeira etapa de teste, foi verificada a função de inversão do estado de um pino (IO) com o intuito de ligar e desligar um led, este posteriormente indicará a que a rede Wi-Fi estará conectada. O estado do led foi verificado visualmente no *hardware* como pode ser visto na Figura 29.

Figura 29 – Verificação do estado do led



Fonte: O autor (2023)

O próximo teste ocorreu com a função que controla a rede Wi-Fi, onde foi verificada a conexão do dispositivo de aquisição de dados com a rede através do monitor de conexão remota disponível no computador. No terminal da IDE foi verificado o estado da conexão com a rede e também foi acompanhado as interações no qual o *firmware* se encontrava. Na Figura 30 pode ser visto a verificação de conexão com a rede Wi-Fi.

Figura 30 – Verificação de conexão com a rede Wi-Fi

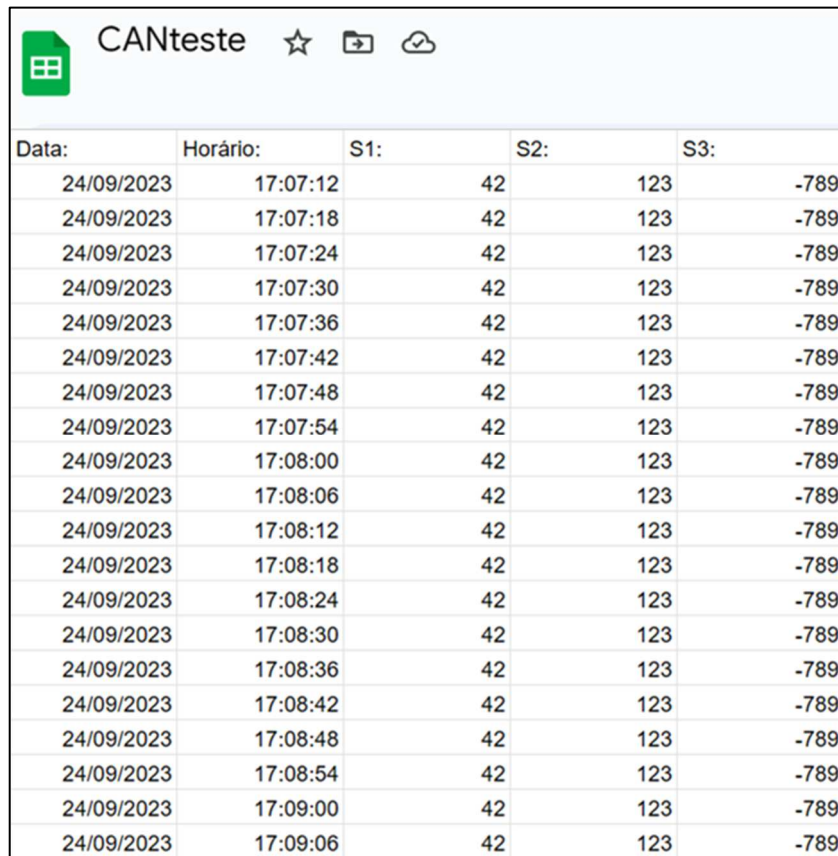
```
wifi CONNECTED
I (702) wifi:AP's beacon interval = 102400 us, DTIM period = 1
I (1192) wifi:<ba-addr>idx:0 (ifx:0, 2e:c9:d3:51:39:2a), tid:0, ssn:0, winSize:64
Its alive
I (2182) esp_netif_handlers: sta ip: 192.168.137.172, mask: 255.255.255.0, gw: 192.168.137.1
wifi got IP...
https://script.google.com/macros/s/AKfycbw7uNjjL8othU0M7HLK1amDEzKx0oLFd0iWYeQoMnh-WCcjl0BkrkmVqa5eI8ND-A3D/exec?value1=42&value2=123&value3=-789HTTP POST Status = 200, content_length = -1
```

Hotspot móvel		Dispositivos conectados: 1 de 8	
Compartilhar minha conexão com a Internet com outros dispositivos			
<input type="checkbox"/> Ativado			
Nome do dispositivo	Endereço IP	Endereço físico (MAC)	
espressif	192.168.137.172	7c:dfa1:3cd6:50	

Fonte: O autor (2023)

O teste seguinte foi realizado com a função que controla o envio de dados para o servidor na nuvem, onde foi realizada uma verificação no monitor da IDE para ver a situação de rede e dos dados enviados. Foram enviados os valores de 3 variáveis de valor fixo para o servidor na nuvem (*Google sheets*), este por sua vez possui o intuito de simular a aquisição dos parâmetros da ECU. Para confirmar o recebimento dos valores das variáveis, foi realizada a verificação na base de dados do servidor. O servidor por sua vez armazena a data, a hora de envio e o valor de cada variável, esse recebimento pode ocorrer em um intervalo de 500 milissegundos ou tempos maiores, o autor definiu um tempo de 6 segundos. Na Figura 31 pode ser vista a verificação do recebimento dos dados no servidor.

Figura 31 – Verificação de recebimento dos dados no servidor



Data:	Horário:	S1:	S2:	S3:
24/09/2023	17:07:12	42	123	-789
24/09/2023	17:07:18	42	123	-789
24/09/2023	17:07:24	42	123	-789
24/09/2023	17:07:30	42	123	-789
24/09/2023	17:07:36	42	123	-789
24/09/2023	17:07:42	42	123	-789
24/09/2023	17:07:48	42	123	-789
24/09/2023	17:07:54	42	123	-789
24/09/2023	17:08:00	42	123	-789
24/09/2023	17:08:06	42	123	-789
24/09/2023	17:08:12	42	123	-789
24/09/2023	17:08:18	42	123	-789
24/09/2023	17:08:24	42	123	-789
24/09/2023	17:08:30	42	123	-789
24/09/2023	17:08:36	42	123	-789
24/09/2023	17:08:42	42	123	-789
24/09/2023	17:08:48	42	123	-789
24/09/2023	17:08:54	42	123	-789
24/09/2023	17:09:00	42	123	-789
24/09/2023	17:09:06	42	123	-789

Fonte: O autor (2023)

### 5.3 TESTES DE APLICAÇÃO

O primeiro teste de aplicação ocorreu no veículo GM *Onix* 2019, onde o dispositivo de aquisição de dados foi conectado ao veículo e acompanhado via o monitor da IDE. Neste primeiro teste foi verificado a requisição dos parâmetros do veículo. Na Figura 32 pode ser visto o dispositivo de aquisição de dados e o computador conectados a ECU do veículo.

Figura 32 – Testes de aplicação monitorado pela IDE



Fonte: O autor (2023)

Durante esse teste foram requisitados os parâmetros de rotação do motor, velocidade e da posição da borboleta de aceleração do veículo. A sonda do osciloscópio foi conectada ao Crx do transceptor para realizar a leitura de sinal recebido. Na Figura 33 podem ser vistas as respostas da ECU e o sinal coletado no osciloscópio.

Figura 33 – Respostas da ECU no monitor da IDE e o sinal coletado no osciloscópio

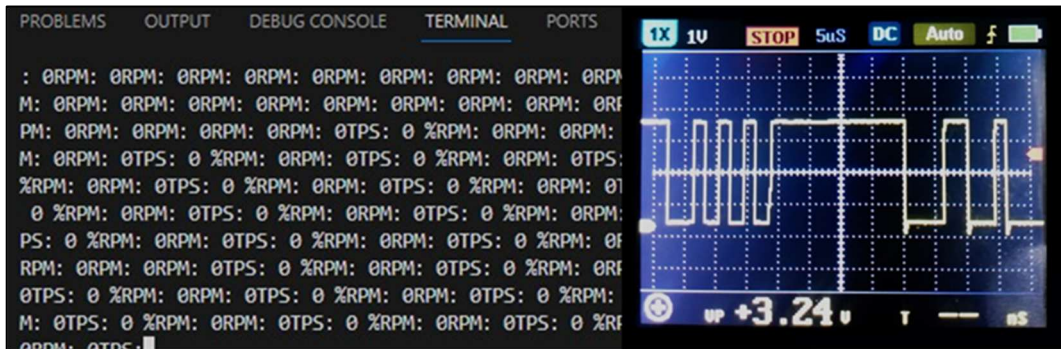


Fonte: O autor (2023)

No primeiro momento de teste houve uma falha durante a comunicação entre o dispositivo de aquisição de dados e a ECU, a requisição dos parâmetros foi realizada com sucesso, porém a ECU não retornou uma resposta no formato CAN 11 *bits*. Isso ocorreu devido o resistor de final de linha localizado no transceptor CAN e dos resistores de proteção do PTC que eram de 100  $\Omega$ . Segundo a KMP *Drivetrain* (2023), para evitar perdas de dados na rede CAN deve-se ter somente um resistor em cada ponta do barramento, sendo que a resistência equivalente deve estar faixa entre 60  $\Omega$  e 120  $\Omega$ . Também a distância máxima entre o barramento e o dispositivo de aquisição de dados deve-se ser menor do que 500 mm.

O barramento do veículo já possui um resistor de final de linha, sendo assim o resistor do transceptor foi removido e os resistores do PTC foram recalculados obtendo o valor de 10  $\Omega$ . O chicote de interface também foi readequado para atender os 500 mm de comprimento. Na Figura 34 podem ser vistas as respostas no formato CAN 11 *bits* e o sinal coletado.

Figura 34 – Respostas no formato CAN 11 *bits* no monitor da IDE e no osciloscópio



Fonte: O autor (2023)

O teste foi retomado e a ECU começou a retornar as respostas no formato CAN 11 *bits* para o dispositivo de aquisição de dados, assim o teste foi finalizado no veículo com o acompanhamento do recebimento das variáveis no servidor na nuvem. Na Figura 35 pode ser visto o veículo Onix instrumentado com o dispositivo de aquisição de dados.

Figura 35 – Instrumentação do veículo *Onix*



Fonte: O autor (2023)

O teste foi finalizado com o veículo sendo conduzido em um percurso urbano de 650 metros, com diferença de elevação de 15 metros, composto por ruas pavimentadas, com vias

preferenciais e não preferenciais (Figura 36). O acompanhamento do recebimento dos dados no servidor foi realizado com um computador junto ao veículo, este também foi o responsável por fornecer a rede Wi-Fi para o dispositivo de aquisição de dados. O dispositivo apresentou constância, retornando valores dos parâmetros requisitados durante o percurso. Na Figura 37 pode ser visto o teste de condução com o veículo *Onix*.

Figura 36 – Percurso do teste de condução



Fonte: Adaptado de Google Mapas (2023)

Figura 37 – Registro parcial do teste de condução com o veículo *Onix*

Data:	Horário	Rotação [RPM]	Velocidade [km/h]	Posição da Borboleta [%]
04/11/2023	15:01:17	966.0	4.0	21.0
04/11/2023	15:01:24	1708.0	12.0	20.0
04/11/2023	15:01:30	1110.0	13.0	17.0
04/11/2023	15:01:36	1753.0	21.0	57.0
04/11/2023	15:01:43	1049.0	12.0	47.0
04/11/2023	15:01:50	1913.0	34.0	33.0
04/11/2023	15:01:56	859.0	10.0	25.0
04/11/2023	15:02:03	1862.0	23.0	21.0
04/11/2023	15:02:09	888.0	11.0	23.0
04/11/2023	15:02:16	1618.0	20.0	20.0
04/11/2023	15:02:22	1162.0	14.0	17.0
04/11/2023	15:02:28	929.0	11.0	20.0
04/11/2023	15:02:35	1560.0	20.0	20.0
04/11/2023	15:02:42	1652.0	30.0	20.0
04/11/2023	15:02:48	3242.0	21.0	38.0
04/11/2023	15:02:55	1355.0	17.0	30.0
04/11/2023	15:03:02	1501.0	28.0	18.0
04/11/2023	15:03:08	1184.0	21.0	36.0
04/11/2023	15:03:15	2159.0	39.0	25.0
04/11/2023	15:03:22	1760.0	32.0	21.0

Fonte: O autor (2023)



O gráfico e o registro dos valores adquiridos em teste podem ser vistos nos Apêndices A e B. O segundo teste de aplicação ocorreu no veículo *Volkswagem Up* 2014, onde o teste foi repetido no mesmo percurso, de forma a validar o funcionamento do dispositivo de aquisição de dados. Como no teste anterior o dispositivo também apresentou constância, retornando valores dos parâmetros requisitados. Nas Figuras 38 e 39 podem ser vistos o veículo *Up* instrumentado com o dispositivo e o teste de condução realizado.

Figura 38 – Aquisição dos dados do teste final no veículo *Up*.



Fonte: O autor (2023)

Figura 39 – Registro parcial do teste de condução com o veículo *Up*

CAN1 ☆ 📷 ☁				
Data:	Horário	Rotação [RPM]	Velocidade [km/h]	Posição da Borboleta [%]
05/11/2023	15:10:07	716.0	0.0	31.0
05/11/2023	15:10:13	936.0	13.0	25.0
05/11/2023	15:10:19	1199.0	21.0	21.0
05/11/2023	15:10:27	1088.0	12.0	20.0
05/11/2023	15:10:35	1015.0	34.0	19.0
05/11/2023	15:10:41	977.0	10.0	19.0
05/11/2023	15:10:48	1700.0	15.0	75.0
05/11/2023	15:10:55	940.0	16.0	18.0
05/11/2023	15:11:01	952.0	20.0	19.0
05/11/2023	15:11:07	947.0	14.0	19.0
05/11/2023	15:11:13	936.0	11.0	19.0
05/11/2023	15:11:21	942.0	20.0	18.0
05/11/2023	15:11:27	947.0	30.0	18.0
05/11/2023	15:11:33	917.0	21.0	18.0
05/11/2023	15:11:39	1199.0	21.0	31.0
05/11/2023	15:11:45	1088.0	12.0	25.0
05/11/2023	15:11:51	1015.0	34.0	21.0
05/11/2023	15:11:57	1236.0	39.0	20.0

Fonte: O autor (2023)

## 6 CONCLUSÃO

A partir do estudo realizado no referencial teórico, foi possível compreender a importância dos sistemas eletrônicos embarcados e dos protocolos de comunicação de um veículo. Diante da disponibilidade da rede CAN, se utilizou desse recurso para realizar o desenvolvimento de um *hardware* e *firmware* de um dispositivo de aquisição de dados. Os parâmetros definidos para a aquisição foram definidos pelo autor, de forma que pudessem ser verificados no painel de instrumentos do veículo.

Um dos objetivos propostos para esse projeto foi a implementação de uma comunicação sem fio (Wi-Fi), para realizar a transmissão de dados com o servidor na nuvem. Essa escolha limitou o usuário a possuir uma rede de internet no veículo, para poder acompanhar os dados adquiridos em tempo real. Nesses casos, a leitura posterior dos dados permaneceu como uma alternativa viável, assegurando que as informações fossem devidamente registradas e acessíveis.

Os testes foram realizados em dois veículos diferentes, de forma a validar o funcionamento do dispositivo de aquisição de dados desenvolvido. A partir do dispositivo, foi possível monitorar os valores dos parâmetros em tempo real do veículo. O dispositivo cumpriu com o objetivo principal, pois os valores lidos da velocidade instantânea, da rotação do motor e da posição da borboleta foram armazenados com sucesso no servidor na nuvem.

Embora o dispositivo de aquisição de dados tenha funcionado com êxito, o entendimento da rede CAN em conjunto com o *driver* TWAI do ESP32 foi uma barreira para ser superada ao longo do desenvolvimento do projeto, mostrando a importância da pesquisa realizada nos livros e *datasheets* do fabricante dos componentes utilizados. Em suma, o dispositivo realizou a conexão com o veículo de forma estável. Os demais objetivos estipulados foram alcançados com êxito, validando o dispositivo desenvolvido como uma ferramenta para realizar a leitura dos parâmetros de um veículo.

De forma que o objetivo geral do trabalho foi alcançado com êxito. Diante disso, para um melhor aproveitamento do desenvolvimento do projeto, sugere-se:

- Realizar estudo de uma possível aplicação comercial do produto/serviço;
- Utilizar uma rede de internet, independente do veículo ou de um telefone móvel;
- Explorar outras funções de diagnose disponíveis, como um *scanner* automotivo;
- Desenvolver uma placa de circuito impresso compacta, com conector OBD II embutido no gabinete do dispositivo;
- Realizar a adequação do protocolo de comunicação para a linha média, pesada e fora de estrada, como ônibus, caminhões e máquinas agrícolas.

## REFERÊNCIAS BIBLIOGRÁFICAS

CAPELLI, A. **Eletroeletrônica automotiva - injeção eletrônica, arquitetura do motor e sistemas embarcados**. 1. ed. São Paulo: Editora Saraiva, 2010. 369 p. ISBN:978-85-365-1845-9. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536518459/>. Acesso em: 26 mar. 2023.

CAVALCANTE, Larissa H.; **Sistema de monitoramento automotivo via rede can**. 2018. 71 p. Dissertação (Graduação) - Curso de Engenharia Mecatrônica, Instituto Federal de Santa Catarina, Florianópolis, 2018. Disponível em: <https://repositorio.ifsc.edu.br/handle>. Acesso em: 27 mar 2023.

CHARETTE, Robert. **This Car Runs on Code**. 2009. Disponível em: <https://spectrum.ieee.org/this-car-runs-on-code>. Acesso em: 20 maio 2023.

COMUNICACIÓN, M.D. **Reparación de automotrices PCM-Tableros, Módulos Body y de Transmisión**. Disponível em: <https://www.electronicayservicio.com>. Acesso em: 24 mar 2023.

COPPERHILLTECH. **CAN Bus Mini Breakout Board**. Disponível em: <https://copperhilltech.com/can-bus-mini-breakout-board/>. Acesso em: 02 out. 2023.

CUNHA, H. R. d. C. **Sistema de diagnóstico obd2/eobd para viaturas automóveis**. 2011. Dissertação (Mestrado) — Universidade de Aveiro, Aveiro, Portugal, 2011.

DEUTSCHES INSTITUT FÜR NORMUNG. DIN 72251 **Road vehicles - Voltages for electrical equipment of power-driven vehicle** —. Germany: DIN, 2001.

DEWESOFT. **What Is CAN Bus (Controller Area Network) and How It Compares to Other Vehicle Bus Networks** Disponível em: <https://dewesoft.com/daq/what-is-can-bus/>. Acesso em: 27 mar 2023.

ESPRESSIF. **ESP32-S2-Saola-1**. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/hw-reference/esp32s2/user-guide-saola-1-v1.2.html/>. Acesso em: 27 ago 2023.

FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. **Redes de computadores**. Porto Alegre: Editora AMGH, 2013. 887 p E-book. ISBN 9788580551693. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580551693/>. Acesso em: 01 abr. 2023.

GUIMARÃES, Alexandre de A. **Eletrônica Embarcada Automotiva**. Edição única. São Paulo: Editora Saraiva, 2007. 329 p E-book. ISBN 9788536518503. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536518503/>. Acesso em: 25 mar. 2023.

HOW2ELECTRONICS. **Interfacing MCP2515 CAN Module**. Disponível em: <https://how2-electronics.com/interfacing-mcp2515-can-bus-module/>. Acesso em: 20 ago. 2023.

INSTITUTE FOR PRINTED CIRCUITS. **Generic Standard on Printed Board Design**. IPC-2221, 2003. Disponível em: [https://www.eng.lbl.gov/~shuman/NEXT/CURRENT\\_DESIGN/TP/MATERIALS/IPC2221A\(L\).pdf](https://www.eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC2221A(L).pdf). Acesso em: 20 ago. 2023.

INTERNATIONAL STANDARDIZATION ORGANIZATION. ISO 15765: **Road vehicles — Diagnostics on Controller Area Networks (CAN)** —. Geneva: ISO, 2005.

INTERNATIONAL STANDARDIZATION ORGANIZATION. ISO 15031-3: **Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics** —. Geneva: ISO, 2016.

INTERNATIONAL STANDARDIZATION ORGANIZATION. ISO 11898: **Road vehicles — Controller area network (CAN)**. Geneva: ISO, 2003.

KMP DRIVETRAIN **Practical tips: CAN-Bus**. Disponível em: <https://www.kmpdrivetrain.com/paddleshift/practical-tips-can-bus/>. Acesso em: 04 nov 2023.

MARQUES, Marco A. **CAN Automotivo Sistema de Monitoramento**. 2004. 163 p. Dissertação (Mestrado) - Mestrado em Ciências e Engenharia Elétrica, Universidade Federal De Itajubá, Itajubá 2004. Disponível em: <https://repositorio.unifei.edu.br/jspui/bitstream/>. Acesso em: 28 mar 2023.

MICROCHIP. **High-Speed CAN Transceiver**. Disponível em: <http://ww1.microchip.com/downloads/en/devicedoc/21667d.pdf>. Acesso em: 16 abr. 2023.

MICROCHIP. **Stand-alone can controller with SPI interface**. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-ANController-with-SPI-20001801J.pdf>. Acesso em: 16 abr. 2023.

MORAES, Alexandre F. de. **Redes sem Fio - Instalação, Configuração e Segurança - Fundamentos**. Editora Saraiva, 2020 291 p. E-book. ISBN 9788536522074. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788536522074/> Acesso em: 01 abr. 2023.

NXP. **TJA1050 High speed CAN transceiver**. Disponível em: <https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf>. Acesso em: 20 ago. 2023.

SILVA, Fernanda Rosa da; SOARES, Juliane A.; SILVA, Lídia P C. **Redes sem fio**. Porto Alegre: Editora Sagah 2021. 291 p. E-book. ISBN 9786556901374. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556901374/>. Acesso em: 01 abr. 2023.

SNAP-ON. **Manual do software de comunicação OBD global do veículo**. Disponível em: [https://br.sun-workshopsolutions.com/EAZ0025B49A\\_1\\_A\\_OBD\\_Global\\_2/](https://br.sun-workshopsolutions.com/EAZ0025B49A_1_A_OBD_Global_2/). Acesso em: 26 mar. 2023.

TEXAS INSTRUMENTS. **SN65HVD23x 3.3-V CAN Bus Transceivers**. Disponível em: [https://www.ti.com/lit/ds/symlink/sn65hvd230.pdf?ts=1699198567105&ref\\_url=https%253A%252F%252Fwww.google.com%252F/](https://www.ti.com/lit/ds/symlink/sn65hvd230.pdf?ts=1699198567105&ref_url=https%253A%252F%252Fwww.google.com%252F/). Acesso em: 20 ago. 2023.

UNIVERSIA. **Os dados que transformam a Fórmula 1: é assim que se ganha uma corrida hoje em dia**. Disponível em: <https://www.universia.net/br/actualidad/orientacion-academica/os-dados-que-transformam-a-formula-1-e-assim-que-se-ganha-uma-corrida-hoje-em-dia/>. Acesso em 26 mar. 2023.


UNDERWRITERS LABORATORIES. UL-94. **Tests for Flammability of Plastic Materials for Parts in Devices and Appliances**. – Estados Unidos: UL, 2023.

## ANEXO A – FOLHA DE DADOS DA FAMÍLIA DO ESP32-S2-WROOM-I

### 1 Module Overview

**Note:**

Check the link or the QR code to make sure that you use the latest version of this document:  
[https://www.espressif.com/documentation/esp32-s2-wroom\\_esp32-s2-wroom-i\\_datasheet\\_en.pdf](https://www.espressif.com/documentation/esp32-s2-wroom_esp32-s2-wroom-i_datasheet_en.pdf)



#### 1.1 Features

<p><b>CPU and On-Chip Memory</b></p> <ul style="list-style-type: none"> <li>• ESP32-S2 embedded, Xtensa® single-core 32-bit LX7 microprocessor, up to 240 MHz</li> <li>• 128 KB ROM</li> <li>• 320 KB SRAM</li> <li>• 16 KB SRAM in RTC</li> </ul> <p><b>Wi-Fi</b></p> <ul style="list-style-type: none"> <li>• 802.11 b/g/n</li> <li>• Bit rate: 802.11n up to 150 Mbps</li> <li>• A-MPDU and A-MSDU aggregation</li> <li>• 0.4 μs guard interval support</li> <li>• Operating frequency: 2412 ~ 2484 MHz</li> </ul> <p><b>Peripherals</b></p> <ul style="list-style-type: none"> <li>• GPIO, SPI, LCD, UART, I2C, I2S, Camera interface, IR, pulse counter, LED PWM, TWAI® (compatible with ISO 11898-1, i.e. CAN Specification 2.0), full-speed USB OTG, ADC, DAC, touch sensor, temperature sensor</li> </ul>	<p><b>Integrated Components on Module</b></p> <ul style="list-style-type: none"> <li>• 40 MHz crystal oscillator</li> <li>• 4 MB SPI flash</li> </ul> <p><b>Antenna Options</b></p> <ul style="list-style-type: none"> <li>• On-board PCB antenna (ESP32-S2-WROOM)</li> <li>• External antenna via a connector (ESP32-S2-WROOM-I)</li> </ul> <p><b>Operating Conditions</b></p> <ul style="list-style-type: none"> <li>• Operating voltage/Power supply: 3.0 ~ 3.6 V</li> <li>• Operating ambient temperature: -40 ~ 85 °C</li> </ul> <p><b>Certification</b></p> <ul style="list-style-type: none"> <li>• RF certification: See certificates for <a href="#">ESP32-S2-WROOM</a> and <a href="#">ESP32-S2-WROOM-I</a></li> <li>• Green certification: RoHS/REACH</li> </ul> <p><b>Test</b></p> <ul style="list-style-type: none"> <li>• HTOL/HTSL/uHAST/TCT/ESD</li> </ul>
---	--

#### 1.2 Description

ESP32-S2-WROOM and ESP32-S2-WROOM-I are two powerful, generic Wi-Fi MCU modules that have a rich set of peripherals. They are an ideal choice for a wide variety of application scenarios related to Internet of Things (IoT), such as embedded systems, smart home, wearable electronics, etc.

ESP32-S2-WROOM comes with a PCB antenna (ANT). ESP32-S2-WROOM-I comes with a connector for an external antenna (CONN).

They both feature a 4 MB external SPI flash. The information in this datasheet is applicable to both modules.

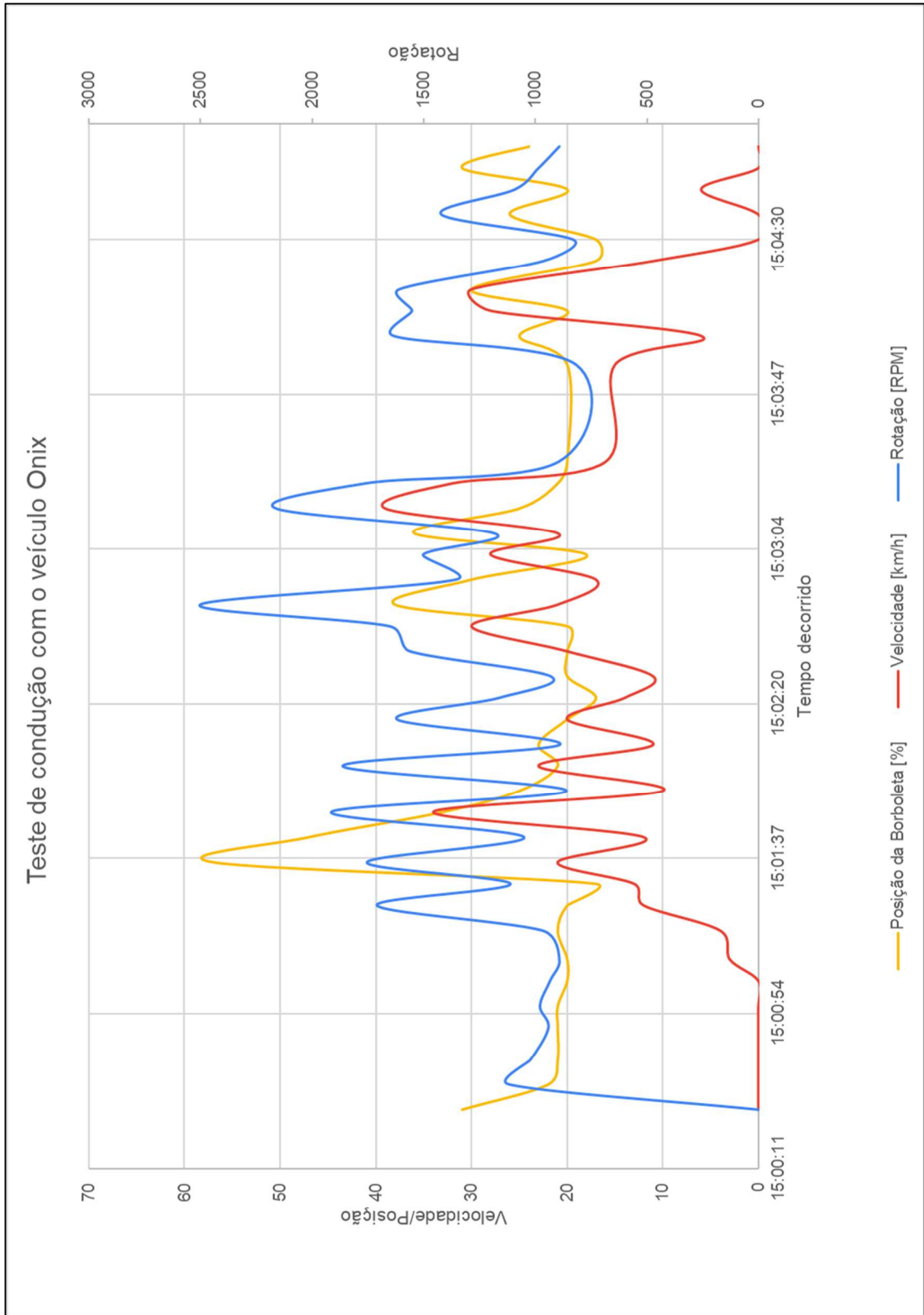
Fonte: Espressif (2022)

## ANEXO B – LISTA DE PARÂMETROS PID DE UM VEÍCULO

PID	Nome, curto	Nome do parâmetro, longo	PID	Nome, curto	Nome do parâmetro, longo
\$03	FUELSYS1	SISTEMA DE COMBUSTÍVEL 1	\$1B	O2S24 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 4 (V)
\$03	FUELSYS2	SISTEMA DE COMBUSTÍVEL 2	\$1B	O2S42 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 4 – SENSOR 2 (V)
\$04	LOAD_PCT	VALOR DA CARGA CALCULADO (%)	\$1B	SHRTFT24 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B2-S4 (%)
\$05	ECT (°C)	TEMPERATURA DO FLUIDO DE ARREFECIMENTO DO MOTOR (°C)	\$1B	SHRTFT42 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B4-S2 (%)
\$06	SHRTFT1 (%)	BANCO DE PREPARAÇÃO DE COMBUSTÍVEL DE CURTO PRAZO 1 (%)	\$1C	OBDSUP	REQUISITOS OBD
\$07	LONGFT1 (%)	BANCO DE PREPARAÇÃO DE COMBUSTÍVEL DE LONGO PRAZO 1 (%)	\$1D	O2SLOC11	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 1
\$08	SHRTFT2 (%)	BANCO DE PREPARAÇÃO DE COMBUSTÍVEL DE CURTO PRAZO 2 (%)	\$1D	O2SLOC12	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 2
\$09	LONGFT2 (%)	BANCO DE PREPARAÇÃO DE COMBUSTÍVEL DE LONGO PRAZO 2 (%)	\$1D	O2SLOC21	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 1
\$0A	FRP (kPa)	MEDIDOR DA PRESSÃO DO COMBUSTÍVEL (kPa)	\$1D	O2SLOC22	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 2
\$0B	MAP (kPa)	MAP DE ADMISSÃO (kPa)	\$1D	O2SLOC31	BANCO DO SENSOR DE OXIGÊNIO 3 – SENSOR 1
\$0C	RPM (1/min.)	VELOCIDADE DO MOTOR (1/min)	\$1D	O2SLOC32	BANCO DO SENSOR DE OXIGÊNIO 3 – SENSOR 2
\$0D	VSS (km/h)	VELOCIDADE DO VEÍCULO (km/h)	\$1D	O2SLOC41	BANCO DO SENSOR DE OXIGÊNIO 4 – SENSOR 1
\$0E	SPARKADV (°)	AVANÇO DO PONTO DE IGNIÇÃO (°)	\$1D	O2SLOC42	BANCO DO SENSOR DE OXIGÊNIO 4 – SENSOR 2
\$0F	IAT (°C)	TEMPERATURA DO AR DE ADMISSÃO (°C)	\$1E	PTO_STAT	STATUS DE RETIRADA DE POTÊNCIA (PTO)
\$10	MAF (g/s)	TAXA DE FLUXO DE AR (g/s)	\$1F	RUNTM (s)	TEMPO DESDE A PARTIDA DO MOTOR (s)
\$11	TP (%)	POSIÇÃO ABSOLUTA DO ACCELERADOR (%)	\$21	MILL_DIST (km)	DISTÂNCIA ENQUANTO MIL ESTÁ ATIVA (km ou milhas)
\$12	AIR_STAT	STATUS DO AR SECUNDÁRIO COMANDADO	\$22	FRP_REL (kPa)	PRESSÃO RELATIVA COMBUSTÍVEL (kPa)
\$13	O2SLOC11	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 1	\$23	FRP (kPa)	MEDIDOR DA PRESSÃO DO COMBUSTÍVEL (kPa)
\$13	O2SLOC12	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 2	\$24	EQ_RAT11	PROPORÇÃO DE EQUIVALÊNCIA B1-S1 (:1)
\$13	O2SLOC13	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 3	\$25	EQ_RAT12	PROPORÇÃO DE EQUIVALÊNCIA B1-S2 (:1)
\$13	O2SLOC14	BANCO DO SENSOR DE OXIGÊNIO 1 – SENSOR 4	\$26	EQ_RAT13	PROPORÇÃO DE EQUIVALÊNCIA B1-S3 (:1)
\$13	O2SLOC21	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 1	\$27	EQ_RAT14	PROPORÇÃO DE EQUIVALÊNCIA B1-S4 (:1)
\$13	O2SLOC22	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 2	\$28	EQ_RAT21	PROPORÇÃO DE EQUIVALÊNCIA B2-S1 (:1)
\$13	O2SLOC23	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 3	\$28	EQ_RAT31	PROPORÇÃO DE EQUIVALÊNCIA B3-S1 (:1)
\$13	O2SLOC24	BANCO DO SENSOR DE OXIGÊNIO 2 – SENSOR 4	\$29	EQ_RAT22	PROPORÇÃO DE EQUIVALÊNCIA B2-S2 (:1)
\$14	O2S11(V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 1 (V)	\$29	EQ_RAT32	PROPORÇÃO DE EQUIVALÊNCIA B3-S2 (:1)
\$14	SHRTFT11 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B1-S1 (%)	\$2A	EQ_RAT23	PROPORÇÃO DE EQUIVALÊNCIA B2-S3 (:1)
\$15	O2S12 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 2 (V)	\$2A	EQ_RAT41	PROPORÇÃO DE EQUIVALÊNCIA B4-S1 (:1)
\$15	SHRTFT12 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B1-S2 (%)	\$2B	EQ_RAT24	PROPORÇÃO DE EQUIVALÊNCIA B2-S4 (:1)
\$16	O2S13 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 3 (V)	\$2B	EQ_RAT42	PROPORÇÃO DE EQUIVALÊNCIA B4-S2 (:1)
\$16	SHRTFT13 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B1-S3 (%)	\$2C	EGR_PCT (%)	EGR COMANDADO (%)
\$17	O2S14 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 4 (V)	\$2D	EGR_ERR (%)	ERRO DO EGR (%)
\$17	SHRTFT14 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B1-S4 (%)	\$2E	EVAP_PCT (%)	PURGA EVAPORATIVA COMANDADA (%)
\$18	O2S21 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 1 (V)	\$2F	FLI (%)	ENTRADA DE NÍVEL DO COMBUSTÍVEL (%)
\$18	O2S31 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 3 – SENSOR 1 (V)	\$30	WARM_UPS	NO. DE AQUECIMENTOS DESDE QUE OS DTCs FORAM APAGADOS
\$18	SHRTFT21 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B2-S1 (%)	\$31	CLR_DIST	DISTÂNCIA A PARTIR DOS DTCs APAGADOS
\$18	SHRTFT31 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B3-S1 (%)	\$32	EVAP_VP (Pa)	PRESSÃO DE VAPOR DO SISTEMA EVAP (Pa)
\$19	O2S22 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 2 (V)	\$33	BARO (kPa)	PRESSÃO BAROMÉTRICA (kPa)
\$19	O2S32 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 3 – SENSOR 2 (V)	\$34	O2S11 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 1 (mA)
\$19	SHRTFT22 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B2-S2 (%)	\$35	O2S12 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 2 (mA)
\$19	SHRTFT32 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B3-S2 (%)	\$36	O2S13 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 3 (mA)
\$1A	O2S23 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 3 (V)	\$37	O2S14 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 1 – SENSOR 4 (mA)
\$1A	O2S41 (V)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 4 – SENSOR 1 (V)	\$38	O2S21 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 1 (mA)
\$1A	SHRTFT23 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B2-S3 (%)	\$38	O2S31 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 3 – SENSOR 1 (mA)
\$1A	SHRTFT41 (%)	PREPARAÇÃO DO COMBUSTÍVEL EM CURTO PRAZO B4-S1 (%)	\$39	O2S22 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 2 (mA)
			\$39	O2S32 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 3 – SENSOR 2 (mA)
			\$3A	O2S23 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 3 (mA)
			\$3A	O2S41 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 4 – SENSOR 1 (mA)
			\$3B	O2S24 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 2 – SENSOR 4 (mA)
			\$3B	O2S42 (mA)	BANCO DE TENSÃO DO SENSOR DE OXIGÊNIO 4 – SENSOR 2 (mA)

Fonte: Adaptado de SNAP-ON (2013)

## APÊNDICE A – GRÁFICO DO TESTE DE CONDUÇÃO COM O VEÍCULO ONIX



Fonte: O autor (2023)

## APÊNDICE B – REGISTRO DO TESTE DE CONDUÇÃO COM O VEÍCULO ONIX

CAN1 ☆ 📁 ☁				
Data:	Horário	Rotação [RPM]	Velocidade [km/h]	Posição da Borboleta [%]
04/11/2023	15:00:27	0	0	31
04/11/2023	15:00:34	1115	0	22
04/11/2023	15:00:41	1020	0	21
04/11/2023	15:00:50	940	0	21
04/11/2023	15:00:56	978	0	21
04/11/2023	15:01:03	933	0	20
04/11/2023	15:01:09	891	3	20
04/11/2023	15:01:17	966	4	21
04/11/2023	15:01:24	1708	12	20
04/11/2023	15:01:30	1110	13	17
04/11/2023	15:01:36	1753	21	57
04/11/2023	15:01:43	1049	12	47
04/11/2023	15:01:50	1913	34	33
04/11/2023	15:01:56	859	10	25
04/11/2023	15:02:03	1862	23	21
04/11/2023	15:02:09	888	11	23
04/11/2023	15:02:16	1618	20	20
04/11/2023	15:02:22	1162	14	17
04/11/2023	15:02:28	929	11	20
04/11/2023	15:02:35	1560	20	20
04/11/2023	15:02:42	1652	30	20
04/11/2023	15:02:48	2500	21	38
04/11/2023	15:02:55	1355	17	30
04/11/2023	15:03:02	1501	28	18
04/11/2023	15:03:08	1184	21	36
04/11/2023	15:03:15	2159	39	25
04/11/2023	15:03:22	1760	32	21
04/11/2023	15:03:28	895	16	20
04/11/2023	15:03:55	816	15	20
04/11/2023	15:04:03	1633	6	25
04/11/2023	15:04:10	1551	28	20
04/11/2023	15:04:16	1608	30	30
04/11/2023	15:04:24	963	11	17
04/11/2023	15:04:30	829	0	17
04/11/2023	15:04:37	1419	0	26
04/11/2023	15:04:44	1091	6	20
04/11/2023	15:04:50	984	0	31
04/11/2023	15:04:56	891	0	24

Fonte: O autor (2023)