

UNIVERSIDADE DE PASSO FUNDO

Bernardo Polese

DISPOSITIVO DE TELEMETRIA VEICULAR
MULTIPROTOCOLO COM TRANSMISSÃO VIA
INTERNET

Passo Fundo

2017

Bernardo Polese

DISPOSITIVO DE TELEMETRIA VEICULAR
MULTIPROTOCOLO COM TRANSMISSÃO VIA
INTERNET

Trabalho apresentado ao curso de Engenharia Elétrica, da Faculdade de Engenharia e Arquitetura, da Universidade de Passo Fundo, como requisito parcial para obtenção do grau de Engenheiro Eletricista, sob orientação do professor Esp. Joan Michel Levandoski.

Passo Fundo

2017

Bernardo Polese

Dispositivo de Telemetria Veicular Multiprotocolo com Transmissão via Internet

Trabalho apresentado ao curso de Engenharia Elétrica, da Faculdade de Engenharia e Arquitetura, da Universidade de Passo Fundo, como requisito parcial para obtenção do grau de Engenheiro Eletricista, sob orientação do professor Esp. Joan Michel Levandoski.

Aprovado em ____ de _____ de _____.

BANCA EXAMINADORA

Prof. Esp. Joan Michel Levandoski - UPF

Prof. Dr. Fernando Passold - UPF

Prof. Dr. Carlos Alberto Ramirez Behaine - UPF

“O fracasso é uma possibilidade. Se as coisas não estão dando errado, você não está inovando o suficiente.”

Elon Musk, CEO da Tesla Motors e SpaceX.

RESUMO

Com a finalidade em agilizar e facilitar a manutenção e diagnóstico de veículos e seus periféricos, é importante acessar seus parâmetros em tempo real. Em alguns casos, especialmente quando se trata dos periféricos, a maioria dos técnicos não possui conhecimento suficiente para compreender o que está acontecendo, e a equipe de engenharia pode ser acionada para solucionar o problema. Do ponto de vista da equipe de engenharia, quando a demanda em campo aumenta não é fácil atravessar todo o país para analisar cada veículo em um curto espaço de tempo. Portanto foi solicitado pela equipe de engenharia da Cielo Telecom Ltda o projeto de um Dispositivo de Telemetria Veicular Multiprotocolo com Transmissão via Internet, o qual possui a função de coletar parâmetros de veículos pesados, que contenham barramentos CAN (*Controller Area Network*) padrão J1939 e J1708 e transmitir esses dados a um software, que executará um servidor TCP para o monitoramento e decodificação da leitura em tempo de execução. O Protocolo CAN é abordado de maneira mais profunda, visto que foi desenvolvido para fins automotivos. Os padrões J1939 e J1708 foram desenvolvidos pela SAE (*Society of Automotive Engineers*) com a finalidade de criar uma norma que abrangesse todos os parâmetros em uma rede veicular. Juntamente com a Interface FMS (*Fleet Management System*) os principais parâmetros foram padronizados, a fim de facilitar a realização de telemetria por terceiros. Os conceitos do Protocolo TCP são apresentados devido à necessidade do desenvolvimento do software que operará como servidor e decodificador da telemetria realizada.

Palavras-Chave: Telemetria, CAN, J1939, J1708, Servidor TCP, Internet

ABSTRACT

In order to perform the maintenance and diagnosis of vehicles and its peripherals in a faster and easier way it is important to get access to its parameters in real time. In some applications, especially when it goes to the peripherals, most of the technicians are not qualified enough to understand what is happening and the engineering team might be required to find out the problem. From the engineers side, when the demand on field increases it is hard to move across the country to analyze each vehicle in a short time. Therefore it was requested by Cielo Telecom engineers the project of a Multiprotocol Vehicle Telemetry with Internet Transmission Device, which works collecting parameters of vehicles that contains CAN J1939 and J1708 buses and broadcasts these data to a TCP Server, which will run a software to monitoring the operation in running-time after the correct decoding of these parameters. The CAN Protocol is discussed in deeper way, because it was developed to automotive purposes. The J1939 and J1708 standards were developed by SAE in order to create standards that cover all parameters in a vehicle network. Together with the FMS Interface, the most relevant parameters have been standardized for the purpose of easiest procedures. The TCP Protocol concepts were presented due to the requirement of the software development, which will work as a server and telemetry decoder.

Keywords: Telemetry, CAN, J1939, J1708, TCP Server, Internet

LISTA DE FIGURAS

Figura 1. Arquitetura Centralizada	18
Figura 2. Arquitetura Distribuida.	19
Figura 3. Camadas do Modelo de Referência OSI.	20
Figura 4. Diferença entre rede sem CAN e rede com CAN.	22
Figura 5. Característica diferencial.....	24
Figura 6. Comparativo entre os formatos CAN 2.0A e CAN 2.0B.....	25
Figura 7. Processo de arbitração do Protocolo CAN.....	27
Figura 8. Hardware para a implementação SAE J1708.....	29
Figura 9. Mensagem J1708.....	30
Figura 10. Mensagem J1708/J1587 contendo dois PIDs.....	31
Figura 11. Descrição do PID 21.	31
Figura 12. Rede J1939.....	32
Figura 13. Identificador J1939.	33
Figura 14. Diferença entre PGN Global e PGN Específico.	34
Figura 15. Exemplo SAE J1939 (PGN FEEE ₁₆)	35
Figura 16. SPN917.	35
Figura 17. Camadas do Modelo TCP/IP.....	38
Figura 18. Diagrama e contextualização do projeto	41
Figura 19. Esquemático de ligação do SN75176BP.....	43
Figura 20. Esquemático de ligação do MCP2551.	44
Figura 21. Módulo Wi-Fi ESP8266 ESP-01	45
Figura 22. Esquemático de ligação do ESP8266 ESP-01.....	45
Figura 23. Ligação DS3231.....	46
Figura 24. Esquemático de ligação do soquete do Cartão SD.....	47
Figura 25. Caixa do protótipo com Display e teclado.	48
Figura 26. Esquemático das fontes de alimentação.....	49
Figura 27. Fluxograma geral do firmware.....	50
Figura 28. Tela de menu.	51
Figura 29. Tela de informações do roteador.....	51
Figura 30. Tecla de mudança de teclado.	51
Figura 31. Fluxograma da aquisição.....	53
Figura 32. Modos de operação do software.....	54

Figura 33. Logs J1939 e J1708.....	55
Figura 34. Gráficos e valores decodificados.	56
Figura 35. PGNs encontrados.....	57
Figura 36. Fluxograma de funcionamento do Simulador J1939/J1708.....	58
Figura 37. Teste do protótipo	60
Figura 38. Servidor iniciado com dispositivo desconectado	61
Figura 39. Conexão a internet e ao servidor.....	61
Figura 40. Servidor iniciado com dispositivo conectado	61
Figura 41. Localização fios J1708 (laranja e cinza) e J1939 (verde e amarelo).....	62
Figura 42. Aferição da telemetria.....	64

LISTA DE TABELAS

Tabela 1. Taxa de transmissão x distância para barramento CAN.....	24
Tabela 2. Principais normas ISO e SAE.....	28
Tabela 3. Parâmetros disponíveis no Simulador J1939 e J1708.	59
Tabela 4. Parâmetros encontrados e seus protocolos	62
Tabela 5. Especificação de decodificação dos parâmetros.....	63

LISTA DE ABREVIATURAS

ABS – *Anti-lock Breaking System*
ACK – *Acknowledgement*
ANSI – *American National Standards Institute*
AT – *Attention*
CAN – *Controller Area Network*
COTS – *Connection Oriented Transport Service*
CRC – *Cyclic Redundancy Check*
CSMA/CA – *Carrier Sense Multiple Access with Collision Avoidance*
DNS – *Domain Name Server*
DLC – *Data Length Code*
ECU – *Electronic Control Unit*
EIA – *Electronic Industries Association*
EOF – *End of Frame*
FMS – *Fleet Management System*
FTP – *File Transfer Protocol*
GPIO – *General Purpose Input/Output*
GmbH – *Gessellschaft mit beschränkter Haftung*
HLEN – *Header Length*
HTTP – *Hypertext Transfer Protocol*
IDE – *Identifier Extension*
IFS – *Inter-frame Extension*
IHL – *Internet Header Length*
IoT – *Internet of Things*
IP – *Internet Protocol*
ISO – *International Organization for Standardization*
I²C – *Inter-Integrated Circuit*
JPEG – *Joint Photographs Experts Group*
LCD – *Liquid Crystal Display*
LDO – *Low Drop-out Voltage*
LLC – *Logic Link Control*
MAC – *Media Access Control*
MID – *Message Identifier*

MP3 – *MPEG-1/2 Audio Layer 3*
NRZ – *Non-Return-to-Zero*
OBD – *On-board Diagnostic*
OSI – *Open Systems Interconnection*
PDU – *Protocol Data Unit*
PG – *Parameter Group*
PGN – *Parameter Group Number*
PID – *Parameter Identifier*
ppm – *partes por milhão*
PWM – *Pulse Width Modulation*
RTC – *Real-Time Clock*
RTR – *Remote Transmission Request*
SAE – *Society of Automotive Engineers*
SD – *Secure Digital*
SOF – *Start of Frame*
SPI – *Serial Peripheral Interface*
SPN – *Suspect Parameter Number*
SRR – *Substitute Remote Request*
TCP – *Transmission Control Protocol*
TCP/IP – *Transmission Control Protocol/Internet Protocol*
TIA – *Telecommunications Industries Association*
TTL – *Transistor-Transistor Logic*
UART – *Universal Asynchronous Receiver/Transmitter*
USART – *Universal Synchronous/Asynchronous Receiver/Transmitter*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS	14
1.2 JUSTIFICATIVA	14
1.3 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 TELEMETRIA	15
2.1.1 História	15
2.1.2 Métodos de Telemetria.....	16
2.1.3 Telemetria no Meio Automotivo	16
2.2 REDE DE COMUNICAÇÃO AUTOMOTIVA	17
2.2.1 ECU.....	18
2.2.2 Arquitetura Centralizada	18
2.2.3 Arquitetura Distribuída	19
2.3 O MODELO OSI.....	19
2.3.1 Conceito	19
2.3.2 Camadas do Modelo OSI	20
2.3.2.1 <i>Camada Física.....</i>	20
2.3.2.2 <i>Camada de Enlace.....</i>	20
2.3.2.3 <i>Camada de Rede.....</i>	21
2.3.2.4 <i>Camada de Transporte</i>	21
2.3.2.5 <i>Camada de Sessão.....</i>	21
2.3.2.6 <i>Camada de Apresentação.....</i>	21
2.3.2.7 <i>Camada de Aplicação.....</i>	21
2.4 O PROTOCOLO CAN.....	22
2.4.1 Histórico	22
2.4.2 Características	23

2.4.3 Camada Física	23
2.4.4 Mensagens	25
2.4.5 Arbitração	26
2.5 PADRÕES DE COMUNICAÇÃO AUTOMOTIVA	28
2.5.1 Classes de Protocolos	28
2.5.2 SAE J1708	29
2.5.2.1 <i>CARACTERÍSTICAS</i>	29
2.5.2.2 <i>MENSAGENS</i>	30
2.5.2.3 <i>SAE J1587</i>	31
2.5.3 SAE J1939	32
2.5.3.1 <i>Características</i>	32
2.5.3.2 <i>Grupos de Parâmetros</i>	33
2.5.3.3 <i>Identificador</i>	33
2.5.3.4 <i>Mensagens</i>	35
2.5.3.5 <i>Interface FMS</i>	36
2.6 TECNOLOGIA TCP/IP	37
2.6.1 Arquitetura Internet	37
2.6.2 O Modelo de Referência TCP/IP	37
2.6.3 O Protocolo IP	39
2.6.3.1 <i>Características</i>	39
2.6.4 O Protocolo TCP	39
2.6.4.1 <i>Características</i>	40
2.6.4.2 <i>Mensagens</i>	40
3 DESENVOLVIMENTO	41
3.1 ESPECIFICAÇÃO DO PROJETO	41
3.2 HARDWARE	42
3.2.1 MICROCONTROLADOR	42

3.2.2 TRANSCEIVERS	43
3.2.2.1 <i>SN75176BP</i>	43
3.2.2.2 <i>MCP2551</i>	44
3.2.3 MÓDULO WI-FI ESP8266	44
3.2.4 RTC	46
3.2.4.1 <i>Maxim DS3231</i>	46
3.2.5 CARTÃO SD	47
3.2.6 INTERFACE COM O USUÁRIO	48
3.2.7 FONTES DE ALIMENTAÇÃO	49
3.3 FIRMWARE	50
3.3.1 Operação Remota	51
3.3.2 Operação Local	52
3.3.3 Aquisição dos Dados	52
3.4 SOFTWARE	54
3.4.1 Modos de operação	54
3.4.2 Logs	55
3.4.3 Gráficos e decodificação	56
3.4.4 PGNs encontrados	57
3.5 SIMULADOR J1939 E J1708	58
4 RESULTADOS	60
4.1 CONEXÃO A INTERNET E SERVIDOR.....	60
4.2 OBTENÇÃO E TRANSMISSÃO DOS PARÂMETROS.....	62
4.3 DECODIFICAÇÃO E APRESENTAÇÃO	63
5 CONSIDERAÇÕES FINAIS	65
REFERÊNCIAS	66
APÊNDICE A – ESQUEMÁTICO ELÉTRICO DO PROTÓTIPO	70
APÊNDICE B – TELA PRINCIPAL DO SOFTWARE DE TELEMETRIA	71

APÊNDICE C – ESQUEMÁTICO ELÉTRICO DO SIMULADOR J1939 E J1708	72
ANEXO A – PARÂMETROS DE DECODIFICAÇÃO DOS DADOS J1939.....	73
ANEXO B – PARÂMETROS DE DECODIFICAÇÃO DOS DADOS J1708	74

1 INTRODUÇÃO

O uso de caminhões caracteriza-se indispensável para a economia brasileira, sendo esses os principais representantes de transporte de carga por todo o território nacional. Aproveitando esse nicho no mercado, as empresas de rastreamento apresentam equipamentos inovadores, capazes de monitorar remotamente inúmeros parâmetros do veículo.

Visando a necessidade de facilitar a telemetria dos caminhões, foi acordado, em 2002, por seis das maiores fabricantes europeias de veículos pesados, a Interface FMS, acordo o qual trata sobre a disponibilização de diversos parâmetros de bordo a terceiros, utilizando o padrão SAE J1939. Hoje, esse padrão opera sobre o protocolo CAN, desenvolvido a partir de 1983 pela Robert Bosch GmbH.

Logo, tornou-se mais fácil obter as informações de bordo e transmiti-las, em conjunto com a telemetria. Todavia, como diagnosticar e interpretar esses parâmetros não é uma tarefa simples, deslocar engenheiros para realizar a manutenção acaba gerando um custo muito alto para a empresa que presta o serviço, considerando que esses perdem tempo em traslado e de desenvolvimento para solucionar problemas em campo.

Assim, o foco desse projeto consiste no desenvolvimento de um dispositivo capaz de realizar a leitura dos parâmetros de veículos que utilizem o protocolo CAN J1939 e J1708 e transmitir esses dados para um servidor TCP utilizando internet, o qual executará um software responsável pela decodificação desses parâmetros, possibilitando que engenheiros da Cielo Telecom possam auxiliar na manutenção sem a necessidade de deslocamento.

Ademais, foi considerada a situação em que a conexão com o servidor não esteja disponível, utilizando assim a gravação dos parâmetros em um cartão de memória, para posteriormente serem importados e analisados no software desenvolvido.

1.1 OBJETIVOS

Os objetivos principais desse trabalho são:

- Desenvolver o dispositivo de diagnóstico com transmissão remota (*hardware* e *firmware*);
- Desenvolver um *software* capaz de obter e importar os dados recebidos por internet e cartão de memória, possibilitando a decodificação e a apresentação em gráficos e relatórios;
- Demonstrar as aplicações e possibilidades do protocolo CAN J1939 e J1708 em conjunto com conexões remotas no ramo automotivo.

1.2 JUSTIFICATIVA

O custo com o deslocamento de engenheiros para locais de manutenção em veículos é elevado para uma empresa, pois, ao realizarem longas viagens, perdem tempo desenvolvendo soluções e ficam ociosos na maior parte do tempo de traslado.

Portanto, com a utilização do equipamento, a operação pode ser acompanhada remotamente, já que os dados serão transmitidos a um servidor TCP executado por um software. No caso de falta de acesso à internet no local da manutenção, um relatório é salvo em um cartão de memória, possibilitando à equipe de engenharia importá-lo nesse mesmo software. Essa equipe responsabiliza-se por analisar os parâmetros obtidos em tempo de execução, caso haja a transmissão, e por desenvolver soluções dedicadas para modelos específicos de veículos.

1.3 ESTRUTURA DO TRABALHO

O primeiro capítulo do presente trabalho refere-se à apresentação, bem como aos objetivos e à justificativa da realização do estudo. No segundo capítulo, com o objetivo de contextualizar o leitor aos temas abordados para a execução do trabalho, será apresentada a fundamentação teórica, a partir de conteúdos obtidos de artigos, teses e livros que discorrem sobre o assunto. O terceiro capítulo trata do desenvolvimento do projeto, ou seja, desde a especificação e análise dos componentes necessários, até a descrição das etapas de desenvolvimento. Por último são apresentados os resultados seguidos pela conclusão do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão abordados assuntos concernentes ao desenvolvimento do projeto proposto, os quais buscam contextualizar o leitor desde sua origem até sua importância para o desenvolvimento satisfatório do protótipo, com figuras que ilustram pontos carentes de maior destaque.

Utilizou-se literaturas abrangentes do assunto, documentos históricos, livros, artigos de revistas e congressos, trabalhos e dissertações, além de normas que tratam especificamente dos tópicos relacionados.

2.1 TELEMETRIA

A telemetria é o processo em obter informações de uma localização qualquer e transmitir esses dados para um local conveniente a fim de ser examinado e armazenado (LOZANO-NIETO, 1999).

2.1.1 História

A telemetria surgiu no ano de 1845 a partir da transmissão de dados entre o *Winter Palace* e a sede do exército russo. Simultaneamente, foi desenvolvido o telémetro, utilizado para transmissão automática da velocidade de balas de canhões (MAYO-WELLS, 1963).

Em 1874, foi apresentado um sistema de telemetria para transmissão de três parâmetros (profundidade de neve, temperatura e pressão barométrica) das montanhas do *Mont Blanc* por mais de 350 quilômetros para Paris utilizando impulsos elétricos em uma linha de transmissão (MAYO-WELLS, 1963).

Apesar dessas demonstrações de seu potencial, não houve apelo para a utilização da telemetria na Europa. Entretanto, com a inauguração do Canal do Panamá em 1914 e seus extensos sistemas de telemetria, os quais reportavam níveis da água e fenômenos físicos, o brilhante futuro dessa tecnologia foi enfatizado ainda mais, como relatado em vários artigos técnicos (MAYO-WELLS, 1963).

2.1.2 Métodos de Telemetria

Segundo Lozano-Nieto (1999), existem diferentes meios para executar a telemetria: ópticos, mecânicos, hidráulicos, elétricos, entre outros. Atualmente, o método mais utilizado é baseado em sinais elétricos, sendo vantajoso sobre os demais por não se limitar à distância entre as áreas a serem analisadas, além de ser facilmente adaptado em infraestruturas já existentes.

A telemetria elétrica pode ser dividida em duas categorias: Telemetria cabeada ou sem fio.

A cabeada é tecnologicamente a solução mais simples, podendo ser implementada em estruturas prontas. Já a telemetria sem fio é mais complexa: requer um estágio de radiofrequência. Apesar de sua complexidade, é amplamente mais utilizada, já que a informação pode ser transmitida por longas distâncias, permitindo a implementação em locais com difícil acesso. Pode também transmitir dados em alta velocidade e tem capacidade suficiente para transmitir em diversos canais, se necessário (LOZANO-NIETO, 1999).

2.1.3 Telemetria no Meio Automotivo

Diversas áreas da ciência utilizam o conceito de telemetria, como a medicina, biologia e agricultura. Já no ramo automotivo, a telemetria se popularizou devido às competições automobilísticas, principalmente a *Formula One*, sendo utilizada como fator decisivo nas competições (TEIXEIRA, DE OLIVEIRA e HELLENO, 2014).

Engenheiros empregam a telemetria para adquirir uma vasta quantidade de dados durante treinos e corridas, e os utilizam para obter a melhor performance possível. Os parâmetros obtidos incluem: acelerações nos três eixos (força G), temperaturas, velocidades das rodas e deslocamento da suspensão. Inclusive, em caso de acidentes, a causa pode ser descoberta a partir da análise desses dados (TANABE, 2015).

A telemetria veicular é presente também no ramo comercial. Em utilização conjunta aos rastreadores automotivos, são instalados variados sensores e um controlador, com a finalidade de extrair informações, como identificação do motorista, consumo de combustível, distância percorrida, locais visitados, velocidade, eventos como freadas e acelerações bruscas, portas abertas, e mais. (TEIXEIRA, DE OLIVEIRA e HELLENO, 2014).

A vantagem desse sistema é que as informações podem ser transmitidas em tempo real, contribuindo na redução de custo em manutenções, consumo de combustível e acidentes.

Todavia, ao adicionar sensores, o custo se torna elevado, além da possibilidade de influenciarem o funcionamento dos circuitos eletrônicos do próprio veículo (TEIXEIRA e TORMIER, 2015).

Alternativamente, se torna viável a utilização da própria rede de comunicação do veículo, visto que essa possui todos os parâmetros necessários para a execução da telemetria, eliminando a utilização de sensores externos e intervenção no sistema elétrico do veículo (TEIXEIRA, DE OLIVEIRA e HELLENO, 2014).

Para a realização de uma telemetria válida e eficiente, parâmetros do veículo devem ser adquiridos e analisados, como:

- Velocidade do tacógrafo;
- Velocidade das rodas;
- Velocidade do motor;
- Odômetro;
- Temperatura do líquido refrigerante do motor;
- Posição do pedal de acelerador;
- Consumo de combustível;
- Nível de combustível;
- Carga do motor.

Portanto, todos esses parâmetros descritos acima serão coletados, decodificados e apresentados ao usuário do dispositivo.

2.2 REDE DE COMUNICAÇÃO AUTOMOTIVA

Hoje, os veículos apresentam um intenso fluxo de informações relativas ao seu funcionamento (TEIXEIRA, DE OLIVEIRA e HELLENO, 2014).

Para transmissão desses dados pelo veículo, são necessários vários módulos eletrônicos, responsáveis pela leitura das entradas, gerenciamento dos protocolos de comunicação, intercâmbio de dados e acionamento das saídas. Os modelos de módulos eletrônicos são denominados ECU (*Electronic Control Unit*), ou Unidade de Controle Eletrônico (GUIMARÃES e SARAIVA, 2003).

2.2.1 ECU

As ECUs são de extrema importância na arquitetura de um veículo, pois são encarregadas de realizar o controle de temporizadores, motor, freios, entre outros (GUIMARÃES e SARAIVA, 2003).

Com o avanço tecnológico dos veículos, novas funções foram implementadas e controladas de forma eletrônica. Conseqüentemente o número de ECUs aumentou, sendo que um carro de passageiros mediano pode conter cerca de trinta ECUs. Já um veículo topo de linha pode possuir mais de duzentas ECUs implementadas (LOPES, 2009).

As ECUs estão em constante comunicação entre elas, como por exemplo o painel, que regularmente necessita saber o estado dos mais variados periféricos do veículo (CANNEWSTELLER.COM, 2011 apud BALDISSERA, 2011), e o ABS (*Anti-lock Braking System*), que, em caso de falha, comunica a ECU do motor para limitar a potência disponibilizada (LOPES, 2009).

Os sistemas de controle de uma aplicação embarcada podem ser conectados em diversas formas, sendo chamadas de Arquitetura Elétrica. No setor automotivo se destacam duas: Arquitetura Centralizada e Arquitetura Distribuída (GUIMARÃES, 2007).

2.2.2 Arquitetura Centralizada

É classificado como Arquitetura Centralizada quando é encontrado apenas uma ECU responsável por receber todos os sinais de entrada, processá-los e comandar as respectivas saídas de controle do sistema. (GUIMARÃES e SARAIVA, 2003).

Apresenta como principal vantagem um *hardware* simples. Em contrapartida, a grande quantidade de cabeamento e a dificuldade na realização de uma manutenção caracterizam pontos negativos desse sistema (VARGAS, 2007). A Figura 1 representa este conceito de arquitetura.

Figura 1. Arquitetura Centralizada

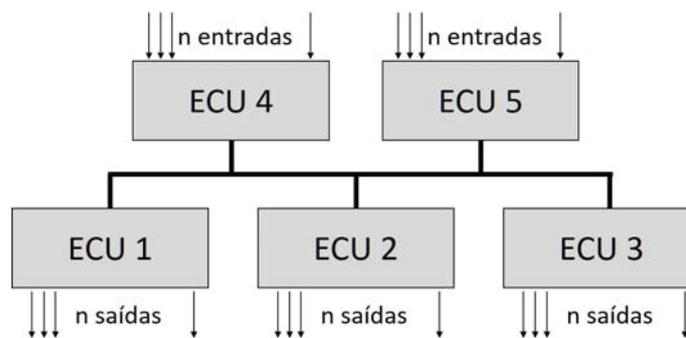


Fonte: Adaptado de GUIMARÃES, 2007

2.2.3 Arquitetura Distribuída

Ao dividir as tarefas em várias ECUs interligadas em um mesmo sistema de controle, é apresentada a Arquitetura Distribuída, resultando em uma redução de cabeamento, ampliação do sistema com facilidade e modularização do projeto. Contudo, este método implica na utilização de um protocolo de comunicação (GUIMARÃES e SARAIVA, 2003). A Figura 2 representa esse tipo de arquitetura.

Figura 2. Arquitetura Distribuída.



Fonte: Adaptado de GUIMARÃES, 2007

2.3 O MODELO OSI

Buscando implementar um sistema aberto a partir de padronizações, a ISO (*International Organization for Standardization*) especificou o Modelo de Referência OSI (*Open Systems Interconnection*), ou simplesmente Modelo OSI (BRISA, 1994).

2.3.1 Conceito

Baseado no conceito de camadas sobrepostas, no qual cada camada executa um conjunto bem definido de funções, o Modelo OSI opera no princípio de usuário e prestador de serviços, sendo que cada uma das camadas do modelo é prestadora de serviços à camada imediatamente superior, e usuária dos serviços prestados pela camada imediatamente inferior (BRISA, 1994). Este modelo é composto por sete camadas, como ilustrado na Figura 3.

Figura 3. Camadas do Modelo de Referência OSI.

Diversos Tipos de Aplicação

Aplicação	7	Camadas Relacionadas com a Aplicação
Apresentação	6	
Sessão	5	
Transporte	4	
Rede	3	Camadas Relacionadas com Aspectos de Transmissão
Enlace	2	
Física	1	

Fonte: Adaptado de BRISA, 1994

2.3.2 Camadas do Modelo OSI

O Modelo OSI é estruturado em sete camadas, sendo as três camadas inferiores responsáveis pela interconexão de sistemas ou de equipamentos individuais, relacionadas a aspectos de transmissão. A camada de transporte, por sua vez, provê a comunicação fim-a-fim entre processos individuais. Por último, as três camadas superiores prestam serviços relacionados com a natureza da aplicação (BRISA, 1994).

2.3.2.1 Camada Física

É a camada mais baixa, responsável pela definição e reconhecimento de um bit. O projeto de rede deve garantir que quando um nó enviar um bit de valor 1, ele seja recebido pelo outro nó como um bit 1, e não como um bit 0. Esta também é a camada onde estão definidos os meios de transmissão, inclusive o tamanho dos cabos, interfaces de conexões (terminações de cabos) e tensão (BALDISSERA, 2011).

2.3.2.2 Camada de Enlace

De acordo com BRISA (1994), a camada de enlace tem por objetivo realizar a transferência de dados sobre uma conexão física de maneira confiável. Ela deve prover funções e procedimentos que permitam ativar, manter e desativar um enlace físico, possuindo mecanismos de detecção e, se aplicável, de correção de erros da camada física. Para estes fins, a camada de enlace é subdividida em duas outras camadas: LLC (*Logic Link Control*) e MAC (*Media Access Control*).

2.3.2.3 *Camada de Rede*

Esta camada é responsável por atribuir um endereço único e global para os dispositivos conectados nesta, e prover direções de a qualquer outro ponto da rede. De modo geral, a qualidade do serviço fornecido (retardo, tempo em trânsito, instabilidade, etc.) também é uma questão da camada de rede (BALDISSERA, 2011).

2.3.2.4 *Camada de Transporte*

A camada de transporte garante que as mensagens sejam entregues sem erros, em sequência e sem perdas ou duplicações. Ela elimina para os protocolos de camadas superiores qualquer preocupação a respeito da transferência de dados entre eles e seus pares (MICROSOFT, 2013).

2.3.2.5 *Camada de Sessão*

Segundo Gallo e Hancock (2005) esta camada é vista como responsável por coordenar o fluxo de dados entre os nós. Nela, são implementadas regras para sincronização das trocas de mensagens, e averigua quais procedimentos tomar em caso de falhas.

2.3.2.6 *Camada de Apresentação*

É a camada responsável pela compressão, criptografia e tradução dos dados para um formato padrão: JPEG (*Joint Photographs Experts Group*) e MP3 (*MPEG-1/2 Audio Layer 3*) são alguns exemplos (BALDISSERA, 2011).

2.3.2.7 *Camada de Aplicação*

A camada de aplicação contém uma série de protocolos comumente necessários aos usuários. Ela é responsável por fornecer uma interface aos softwares de computador, permitindo que estes sejam escritos apenas uma vez e utilizados em diferentes tipos de rede (TANENBAUM, 2003). Alguns exemplos são: Telnet, FTP, HTTP, DNS (*Domain Name Server*). Também é a camada número cinco do modelo TCP/IP (*Transmission Control Protocol/Internet Protocol*).

2.4 O PROTOCOLO CAN

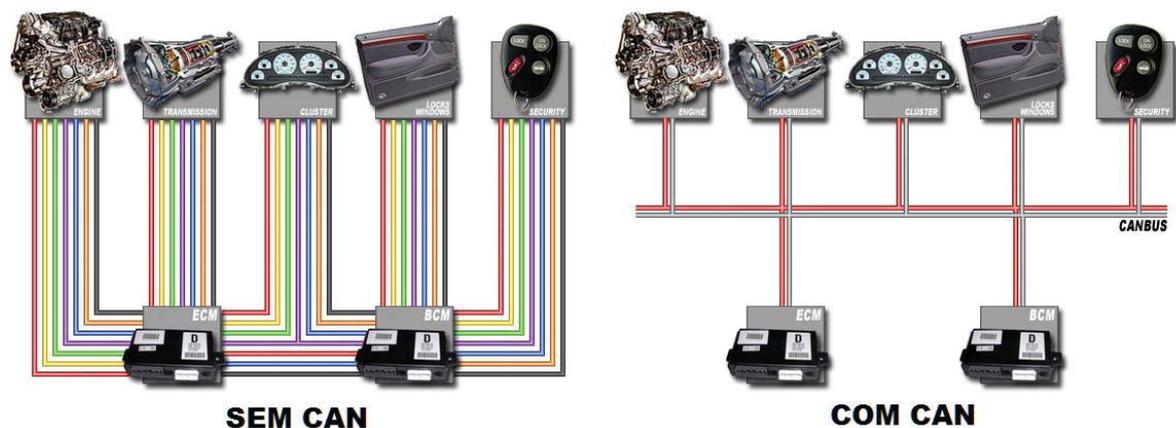
Desenvolvido primeiramente para aplicações automotivas, e implementando as duas camadas mais inferiores do Modelo OSI (física e enlace), o Protocolo CAN é conhecido por um sistema de comunicação prático e robusto em aplicações que necessitem alta confiabilidade de transmissão (BALDISSERA, 2011).

2.4.1 Histórico

Com o aumento das funções de controle implementadas nos veículos e o crescente número de ECUs instaladas a bordo (LOPES, 2009), a quantidade de cabos para conectar todos os componentes cresceu, já que a maioria das conexões automotivas antes do desenvolvimento do Protocolo CAN era do tipo híbrida, combinando topologias Ponto a Ponto e Malha (CANNEWSTELLER.COM, 2011 apud BALDISSERA, 2011).

Logo, a partir da necessidade de produzir automóveis mais confiáveis, seguros, eficientes e ao mesmo tempo reduzir o peso e complexidade do cabeamento, foi apresentado pela empresa Robert Bosch GmbH em 1986, o Protocolo CAN (PAZUL, 1999). A Figura 4 ilustra as diferenças entre uma rede sem CAN e uma rede com CAN.

Figura 4. Diferença entre rede sem CAN e rede com CAN.



Fonte: Adaptado de CANBUSKIT, 2016

2.4.2 Características

O Protocolo CAN se caracteriza por apresentar, segundo Farsi (1999), Lugli e Santos (2009):

- Uma interface serial de alta velocidade configurável para operar em taxas de até 1 Mbit/s;
- Meio físico de baixo custo, visto que o Protocolo CAN opera sobre um par de cabos trançados, reduzindo custos em peso e cabeamento;
- Tamanho de mensagens reduzidos, implicando em baixa latência, quando comparado a outros sistemas;
- Rápido tempo de resposta, posto que não necessita de permissão do barramento para transmitir;
- Operação no conceito multimestre, ou seja, cada módulo pode se tornar mestre a qualquer momento e passar a transmitir no barramento;
- Esquema de arbitragem não destrutiva, isto é, sempre que dois nós começam a transmitir mensagens ao mesmo tempo, é garantido que a mensagem com mais alta prioridade será enviada;
- Utiliza um elaborado esquema de tratamento de erros que resulta na retransmissão das mensagens que não são apropriadamente recebidas.

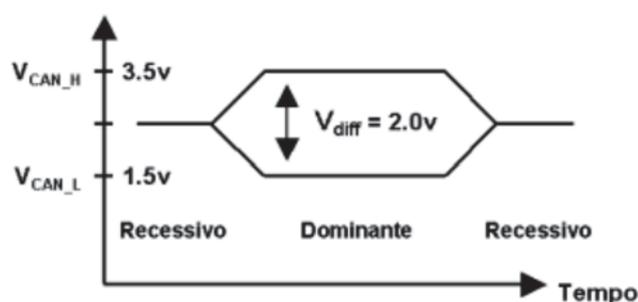
2.4.3 Camada Física

A codificação de bit do Protocolo CAN é definida como sendo do tipo NRZ (*Non-Return-to-Zero*), ou seja, o nível do sinal pode permanecer constante durante todo tempo de bit. Portanto, é importante para o sincronismo que medições sejam feitas para assegurar o tempo máximo permitido entre duas bordas de sinal (BALDISSERA, 2011).

Os dados transmitidos através do barramento são caracterizados por não possuírem nível lógico alto ou baixo, mas sim uma representação denominada bits dominantes e recessivos, em virtude de que as informações são interpretadas pela diferença de potencial entre os fios CAN *High* e CAN *Low*. Essas características são atribuídas pelos protocolos ISO 11898 e ISO 11519, que definem a implementação das duas camadas mais inferiores do Modelo OSI. As camadas restantes são deixadas em aberto para serem elaboradas por desenvolvedores de software. (BALDISSERA, 2011).

Considerando os fios elétricos como meio de transmissão, esses devem ser trançados, e não blindados, visto que essa configuração atenua fortemente os efeitos causados por interferências eletromagnéticas, uma vez que qualquer ação sobre um dos fios será sentida também pelo outro. Desta forma, ambos os sinais irão flutuar para o mesmo sentido com a mesma intensidade, não alterando a tensão diferencial entre o CAN *High* e o CAN *Low*. Logo, devido a essa configuração física, o barramento CAN é classificado como par trançado diferencial (GUIMARÃES, 2007). A Figura 5 ilustra essa característica diferencial.

Figura 5. Característica diferencial.



Fonte: GUIMARÃES, 2003

A taxa de transmissão é inversamente proporcional ao comprimento do barramento. A norma ISO 11898 especifica que um *transceiver* deve ser capaz de fornecer uma velocidade de 1 Mbps em barramento de 40 metros (RICHARDS, 2002). Na Tabela 1 é ilustrada essa relação:

Tabela 1. Taxa de transmissão x distância para barramento CAN.

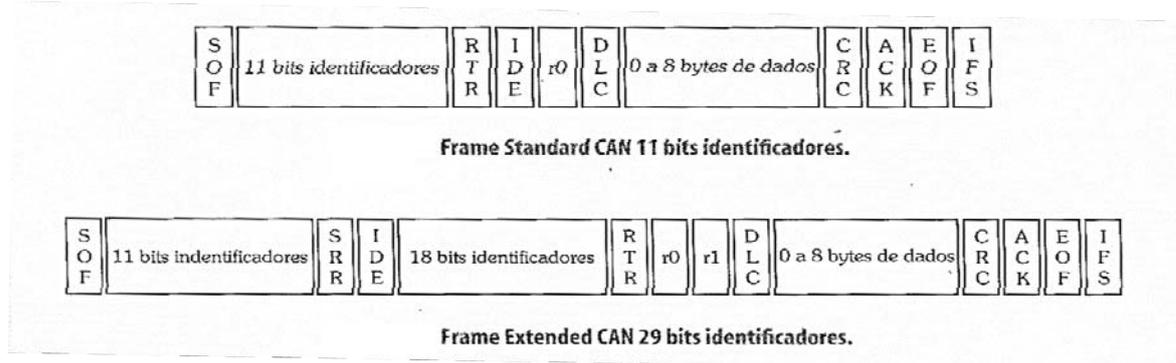
Taxa de transmissão (kbit/s)	Comprimento do barramento (m)
1000	40
500	130
250	270
125	530
100	620
50	1300
20	3300
10	6700
5	10000

Fonte: LUGLI, 2009

2.4.4 Mensagens

De acordo com Baldissera (2011), as mensagens do Protocolo CAN podem assumir dois formatos: CAN 2.0A, ou *standard*, e CAN 2.0B, comumente conhecido como formato estendido. A principal diferença entre esses dois formatos é o tamanho dos identificadores, 11 bits e 29 bits, respectivamente. A Figura 6 ilustra os quadros de mensagens para ambos os formatos.

Figura 6. Comparativo entre os formatos CAN 2.0A e CAN 2.0B



Fonte: LUGLI, 2009

Ao analisar a Figura 6, observa-se que ambos os formatos iniciam com um bit de início, o SOF (*Start of Frame*), seguido imediatamente pelo identificador de 11 bits, o qual também define a prioridade de cada mensagem. O menor valor binário possui a maior prioridade de transmissão (LUGLI e SANTOS, 2009).

O bit RTR (*Remote Transmission Request*), para o formato CAN 2.0A, é um bit que indica que informação está sendo solicitada. Este bit é seguido de mais dois bits em estado dominante (r0 e r1). O bit SRR (*Substitute Remote Request*), para o formato CAN 2.0B cumpre a função de substituir esse bit por um estado sempre dominante (SOUZA, 2010).

O marco da divisão entre os dois formatos se encontra no bit IDE (*Identifier Extension*). Caso esse encontre-se no estado recessivo, indica que os próximos dezoito bits serão a continuação do identificador para construir o formato estendido, caso contrário a mensagem será no formato padrão (BALDISSERA, 2011).

A quantidade de bytes a ser transmitida é apresenta no campo DLC (*Data Length Code*). Logo em seguida, o campo CRC (*Cyclic Redundancy Check*) é responsável pela técnica de detecção de erros (SOUZA, 2010).

Quando o nó recebe a mensagem correta, ele sobrescreve o bit recessivo do campo ACK (*Acknowledgement*) na mensagem original com um bit dominante, indicando que recebeu uma mensagem corretamente (LUGLI e SANTOS, 2009).

Por fim, o campo EOF (*End of Frame*) é responsável por indicar o final da transmissão, e os bits do campo IFS (*Inter-frame Space*), tem a função de disponibilizar tempo necessário ao microcontrolador mover corretamente os dados para um buffer de recepção de mensagem (BALDISSERA, 2011).

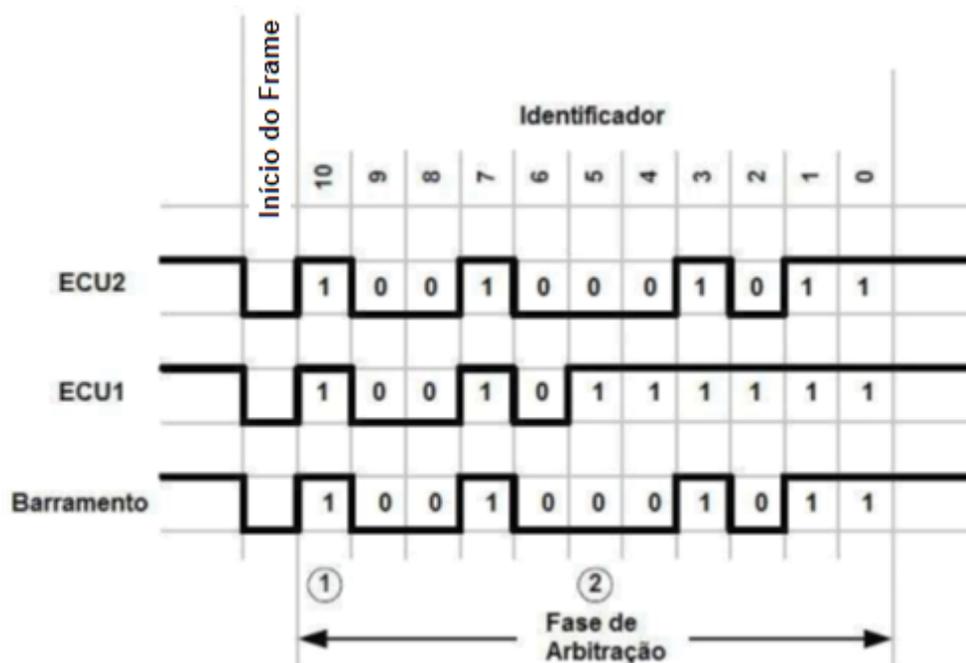
2.4.5 Arbitração

O Protocolo CAN é baseado em mensagens, e não em endereços, ao contrário de outros protocolos, como o *Ethernet*. Embutidos nas mensagens CAN, estão a prioridade e o conteúdo a ser transmitido (FARSI, 1999). No momento em que um nó deseja transmitir, ele recebe acesso ao barramento, inicia a transferência de uma mensagem e todos os outros nós se tornam receptores. Portanto, todos os nós da rede recebem a mensagem. Tendo recebido a mensagem corretamente, os nós iniciam a verificação para determinar se a mensagem recebida é relevante ou não, para o dispositivo em particular (LOPES, 2009).

A prioridade com que uma mensagem é transmitida relativamente à outra é especificada pelos seus identificadores (LUGLI e SANTOS, 2009). Como o bit dominante tem prioridade sobre o recessivo, mensagens com identificadores menores tem maior prioridade para acessar o barramento (LOPES, 2009).

A arbitragem é requerida quando ocorre acesso simultâneo ao barramento por diferentes nós. O método utilizado é o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), em que a prioridade da mensagem é detectada no identificador CAN. Os conflitos são resolvidos através da arbitragem bit a bit dos identificadores das mensagens. Cada nó observa a rede utilizando o mecanismo *bitwise*, em que o estado dominante (0) se sobrepõe ao estado recessivo (1) (LUGLI e SANTOS, 2009). A Figura 7 apresenta o processo de arbitração do protocolo.

Figura 7. Processo de arbitração do Protocolo CAN.



Fonte: MANNISTO e DAWSON 2003.

Conforme mostrado na Figura 7, a comparação dos identificadores da ECU2 e ECU1, que possuem, respectivamente os valores binários 10010001011 (1163 decimal) e 10010111111 (1215 decimal), e utilizando o método de arbitração do Protocolo CAN, é notado que a partir do bit 5, em que é percebida a diferença entre os dois identificadores, o de menor valor, ECU2 vence a arbitração e segue sendo transmitido no barramento.

2.5 PADRÕES DE COMUNICAÇÃO AUTOMOTIVA

A indústria automobilística e várias organizações de padronizações industriais, como a ISO e a SAE vêm trabalhando há anos com grandes montadoras no desenvolvimento de padrões de diagnóstico veicular (GUIMARÃES, 2007). Na Tabela 2, destacam-se algumas das principais normas usadas em diagnose veicular.

Tabela 2. Principais normas ISO e SAE.

	Principais normas				
ISO	11519	11898	14229	14230	15765
SAE	J1708	J1850	J1939	J1979	J2284

Fonte: GUIMARÃES, 2007

2.5.1 Classes de Protocolos

Seguindo os critérios da SAE, os protocolos de comunicação são divididos em grupos, conforme sua taxa de transmissão.

Protocolos Classe A, relacionados às funções de conforto de um veículo, possuem uma taxa transmissão de até 10 kbps. Nesta faixa está incluso o padrão SAE J1708, muito utilizado em veículos pesados. Já os protocolos Classe B, operam em taxas de transmissão entre 10 kbps a 125 kbps, geralmente relacionados ao controle dos sistemas de entretenimento de um veículo. Finalmente, os protocolos Classe C, responsáveis pelo controle de sistemas de segurança, operam em taxas de transmissão entre 125 kbps e 1 Mbps (GUIMARÃES, 2007).

Cabe salientar que além destas classes mencionadas, segundo Baldissera (2011), existem classes de protocolos de:

- Diagnóstico embarcado;
- *Mobile Media*;
- *Safety Bus*;
- *Drive by-wire*

Veículos pesados, como caminhões, ônibus, máquinas agrícolas e de construção, possuem arquiteturas de computação distribuídas, formadas por no mínimo uma rede de controle (J1939) e uma rede de diagnóstico (J1708) (MARINHO e CORDEIRO, 2014). Esses padrões englobam os parâmetros de interesse desse projeto, para a realização da telemetria.

2.5.2 SAE J1708

Visando padronizar o hardware e o protocolo para comunicação entre os módulos eletrônicos de veículos pesados, a SAE publicou em 1986, o Padrão SAE J1708. Este que segundo Julicher (2008), tem como objetivos:

- Minimizar os custos com hardware;
- Fornecer flexibilidade para expansão sem impactar nos sistemas existentes;
- Utilizar um hardware padronizado;
- Disponibilizar aos fabricantes a possibilidade de customização da conexão.

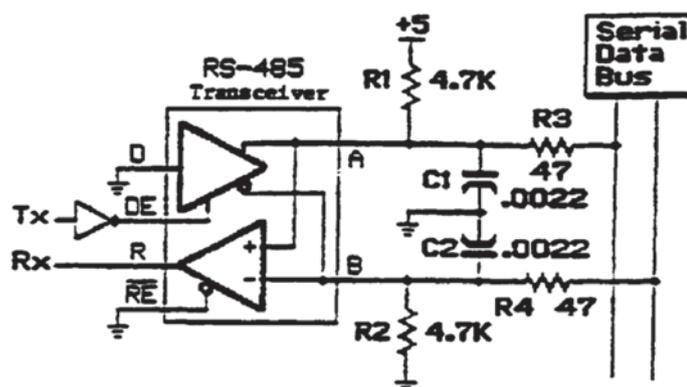
2.5.2.1 CARACTERÍSTICAS

O padrão SAE J1708, assim como o protocolo CAN, implementa as duas camadas inferiores do Modelo de Referência OSI - camada de enlace e física. Também foi definido que este padrão apresente um software básico de aplicação, embora o documento de aplicação SAE J1587 (o qual será abordado na seção 2.5.2.3) defina os principais dados e funções a serem transmitidos no barramento (TEXAS INSTRUMENTS, 1993).

De acordo com a SAE, as características físicas são definidas por um par diferencial de cabos trançados com comprimento máximo de 40 metros e uma quantidade limitada em 20 nós no barramento (TEXAS INSTRUMENTS, 1993).

Devido as suas características elétricas, como tensão diferencial, além do baixo custo e grande disponibilidade, a SAE definiu o hardware para a implementação do protocolo J1708, o padrão RS-485 (JULICHER, 2008). A Figura 8 apresenta o circuito típico proposto pela especificação SAE J1708.

Figura 8. Hardware para a implementação SAE J1708.



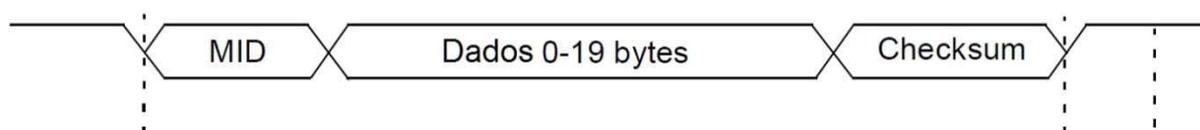
Segundo Saastamoinen (2008), e analisando a Figura 8, as informações transmitidas no barramento são interpretadas a partir da diferença de potencial entre os pontos “A” e “B”. O nível lógico alto é lido quando o ponto “A” é pelo menos 0.2 V mais positivo que o ponto “B”, e nível lógico baixo quando o contrário.

O tempo de bit (*bit time*) especificado é de 9600 bps (*bits per second*), ou seja, os níveis lógicos alto e baixo, não devem exceder 104.2 μ s (microssegundos), tornando o protocolo SAE J1708 consistente ao padrão UART (*Universal Asynchronous Receiver/Transmitter*). (SAASTAMOINEN, 2008).

2.5.2.2 MENSAGENS

As mensagens J1708 possuem um comprimento variável entre 3 a 21 caracteres de comprimento, sendo compostas de um MID (*Message Identifier*), seguido por até 19 caracteres de dados e encerrando com um *checksum* (SAASTAMOINEN, 2008). A Figura 9 apresenta o conteúdo de uma mensagem J1708.

Figura 9. Mensagem J1708.



Fonte: JULICHER, 2008

Segundo Julicher (2008) o MID é responsável por permitir o acesso do dispositivo ao barramento J1708 e definir a prioridade da mensagem a ser transmitida.

De acordo com a SAE (2002) para a identificação dos MIDs válidos em aplicações para veículos pesados (MID entre 128 e 255) é necessário utilizar a norma SAE J1708 em conjunto com a norma SAE J1587, visto que a primeira é responsável pela implementação do hardware e o protocolo básico, enquanto a SAE J1587 é responsável pela camada de transporte e aplicação do Modelo OSI, contendo principalmente, os parâmetros presentes no protocolo e informações necessárias para a realização da decodificação dos dados lidos diretamente ao barramento em valores legíveis em unidades padrões, como RPM, km/h, no caso da velocidade do motor e do veículo, respectivamente.

2.5.2.3 SAE J1587

O SAE J1587 é um protocolo de diagnóstico automotivo desenvolvido pela SAE para a maior parte dos veículos pesados e médios fabricados após 1985. É um protocolo para ser utilizado em conjunto com a norma SAE J1708 (SAE, 2002).

Segundo Kvaser (2016), o Padrão SAE J1587 descreve o formato de uma mensagem, assim como os parâmetros, consistindo em sua estrutura o MID, o PID (*Parameter Identification*), o campo de dados e um *checksum*. O comprimento de uma mensagem é limitado em 21 caracteres de acordo com a norma J1708, porém é permitido o envio de mensagens maiores, utilizando o conceito COTS (*Connection Oriented Transport Service*). Uma mensagem J1587 contendo dois PIDs (21 e 12) é demonstrada na Figura 10.

Figura 10. Mensagem J1708/J1587 contendo dois PIDs.

MID	PID	DATA	PID	DATA1	DATA2	CHECKSUM
128	21	186	12	05	48	112

Fonte: Adaptador de KVASER, 2016

Analisando a Figura 10, juntamente com a norma SAE 1587 (SOCIETY OF AUTOMOTIVE ENGINEERS, 2002) é possível identificar os parâmetros contidos na mensagem e decodificá-la. O primeiro valor, MID 128, corresponde ao identificador da ECU do motor. Em seguida é apresentado o PID 21 com valor 186, e o PID 12 com valores em dois caracteres, 05 e 48 ou seja, é possível conhecer a temperatura interna da ECU do motor e o status dos freios, respectivamente. Na Figura 11 é apresentado a descrição do PID 21, a qual facilita a interpretação da temperatura interna da ECU do motor.

Figura 11. Descrição do PID 21.

A.21 Engine ECU Temperature—Internal air temperature of the engine ECU.

Parameter Data Length: 1 Character
 Data Type: Signed Short Integer
 Bit Resolution: 2.5 °F
 Maximum Range: -320.0 to 317.5 °F
 Transmission Update Period: 1.0 s
 Message Priority: 8

Fonte: SOCIETY OF AUTOMOTIVE ENGINEERS, 2002

De acordo com a Figura 11, cada bit corresponde a 2.5 °F, com um *offset* de 320 °F, ou seja, para decodificação da temperatura do motor, é possível utilizar a Equação 1:

$$\text{Temperatura do motor (°F)} = \text{dado} * 2,5 - 320 \quad (1)$$

Portanto, de acordo com a Equação 1, a temperatura do motor é de 145 °F. Por último a mensagem contém um valor de *checksum*, que quando somado aos outros valores da mensagem e realizado uma operação AND com 255, deve retornar o valor zero. Somente assim a mensagem pode ser considerada válida (KVASER, 2016).

2.5.3 SAE J1939

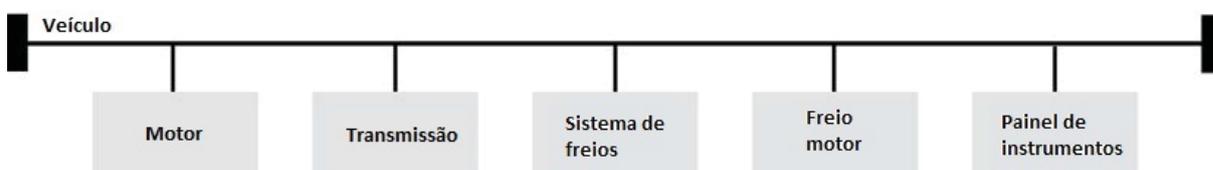
A fim de definir a comunicação entre as ECUs em uma rede veicular, a SAE publicou em 1998, baseado no Protocolo CAN 2.0B, o conjunto de especificações SAE J1939 (LUGLI e SANTOS, 2009).

Atualmente, este padrão é direcionado exclusivamente a veículos pesados, em aplicações de diagnóstico e controle. Devido a sua popularidade, passou também a ser adotado em aplicações agrícolas (ISO 11789), navais e aéreas (NMEA 2000) (BALDISSERA, 2011).

2.5.3.1 Características

O padrão SAE J1939 determina as camadas de rede, transporte, sessão e apresentação, já que o Protocolo CAN define as duas camadas mais inferiores, restando apenas a camada de aplicação para os desenvolvedores (VOOS, 2008). Da mesma forma, é definido a maneira em que os dados serão compartilhados entre as ECUs, como a prioridade, tamanho, escala e *offset* das mensagens (SIMMA SOFTWARE, 2016). Na Figura 12, é apresentada uma rede típica J1939.

Figura 12. Rede J1939.



Fonte: Adaptado de JUNGER, 2010

As principais características do padrão SAE J1939, segundo Junger (2010):

- Baseado no Protocolo CAN 2.0B (Formato estendido);
- Taxa de transmissão de 250 kbit/s;
- Comunicação ponto-a-ponto;
- Manutenção da rede;
- Características de diagnóstico;
- Definição de grupos de parâmetros para veículos comerciais;
- Suporte para grupos de parâmetros específicos do fabricante.

2.5.3.2 Grupos de Parâmetros

A exclusividade do Padrão SAE J1939 está em seus PGs (*Parameter Groups*), identificadores únicos atribuídos a cada mensagem, com a finalidade de identificar a mensagem e seu conteúdo (SIMMA SOFTWARE, 2016) .

Usualmente, um grupo de parâmetros possui comprimento especificado entre 8 até 1785 bytes, embora grupos de parâmetros com mais de 8 bytes requerem um protocolo de transporte para a realização da transmissão (JUNGER, 2010).

2.5.3.3 Identificador

Devido a utilização do Protocolo CAN 2.0B em sua essência, o Padrão SAE J1939 possui um identificador de 29 bits, conhecido como PGN (*Parameter Number Group*), responsável por especificar principalmente o endereço de origem e de destino e a prioridade (JUNGER, 2010). Na Figura 13 é apresentada a composição do identificador SAE J1939.

Figura 13. Identificador J1939.

Prioridade	Página de dados estendida	Página de dados	Formato PDU	PDU Específico	Endereço de origem
3 bit	1 bit	1 bit	8 bit	8 bit	8 bit

Fonte: Adaptado de JUNGER, 2010

A partir da análise da Figura 13, conclui-se que o identificador J1939 é dividido em seis blocos:

- Prioridade, o qual define que quanto menor seu valor binário, maior a prioridade da mensagem;
- Página de dados estendida e página de dados, os quais segundo Baldissera (2011) não possuem funções especificadas para o Padrão SAE J1939, sendo lidos como zero;
- Formato da Unidade de Protocolo de Dados (PDU), caracterizando o subconjunto de bits responsável pela identificação de um PGN. Quando o bloco Formato PDU recebe um valor maior ou igual a 240 a transmissão é realizada em formato de *broadcast*, ou seja, a mensagem possui um PGN Global e todos os nós da rede irão receber a mensagem. Quando esse valor é inferior a 240, denomina-se à mensagem um PGN Específico, pois a mensagem será transmitida a um dispositivo em particular (*peer-to-peer*) (JUNGER, 2010).
- O PDU Específico; segundo Junger (2010), é utilizado apenas quando a mensagem for global, sendo definido como Extensão de Grupo (GE). Caso contrário, é lido sempre como zero. A Figura 14 ilustra a diferença entre PGNs globais e específicos.

Figura 14. Diferença entre PGN Global e PGN Específico.



Fonte: Adaptado de JUNGER, 2010

- Endereço de origem; segundo Baldissera (2010), é o bloco de oito bits o qual contém o endereço da Aplicação de Controle (CA). Cabe salientar que uma ECU pode conter uma ou mais CAs, entretanto cada uma possui um endereço específico associado ao nome do dispositivo (JUNGER, 2010).

Na Figura 15, é apresentado um exemplo retirado da norma SAE J1939-71, exemplificando a relação entre PGs e os parâmetros a serem obtidos, no caso, relacionado a ECU responsável pela Temperatura do Motor.

Figura 15. Exemplo SAE J1939 (PGN FEEE₁₆)

Name:	Engine temperature 1 – ET1	
Transmission rate:	1s	
Data length:	8 bytes	
Extended Data Page	0	
Data page:	0	
PDU format:	254	
PDU specific:	238	
Default priority:	6	
PG Number:	65,262 (00FEEE ₁₆)	
Description of data:		
Byte:	1	Engine Coolant Temperature
	2	Engine Fuel Temperature 1
	3,4	Engine Oil Temperature 1
	5,6	Engine Turbocharger Oil Temperature
	7	Engine Intercooler Temperature
	8	Engine Intercooler Thermostat Opening

Fonte: SOCIETY OF AUTOMOTIVE ENGINEERS, 2003.

2.5.3.4 Mensagens

Segundo Axiomatic (2006), todos os parâmetros utilizados na rede J1939 são descritos pelo comitê da SAE. Para designar um parâmetro específico, a SAE J1939-71 definiu o SPN (*Suspect Parameter Number*), um número de 19 bits com informações detalhadas como comprimento do dado, tipo do dado, resolução, *offset*, variação e um *tag* para referência. Geralmente, SPNs que compartilham características comuns são transmitidos na rede utilizando o mesmo PGN.

Na Figura 16, é apresentado um exemplo na realização da busca de um parâmetro específico, distância total percorrida em alta resolução (SPN917) na Norma SAE J1939-71.

Figura 16. SPN917.

spn917 - Distância Total Percorrida em Alta Resolução

Comprimento do dado:	4 bytes
Resolução:	5 m/bit , 0 offset
Variação:	0 to 21,055,406 km
Tipo:	Medido
SPN:	917
PGN:	[65217]

Fonte: SOCIETY OF AUTOMOTIVE ENGINEERS, 2006

Segundo Baldissera (2011), SPNs não implementados, via de regra, possuem seus bytes preenchidos com o valor hexadecimal FF.

2.5.3.5 Interface FMS

Buscando facilitar às empresas independentes a telemetria de veículos pesados, seis das maiores empresas de caminhões da Europa: Daimler, MAN, Scania, DAF, Volvo (Renault) e IVECO, acordaram no ano de 2002 a Interface FMS (FMS-STANDARD, 2016).

As informações contidas na *Interface FMS* seguem a codificação da Norma SAE J1939. Em contrapartida, a quantidade de dados disponibilizados é dependente não apenas do fabricante, como também do modelo do veículo. Logo, não necessariamente todos os parâmetros listados abaixo se encontrarão disponíveis em um mesmo veículo. Os parâmetros abrangidos, segundo a Interface FMS são descritos:

- Velocidade do Veículo (baseada na roda e tacógrafo);
- Chave da embreagem (on/off);
- Chave do freio (on/off);
- Controle de cruzeiro (on/off);
- Tomada de força (estado/modo);
- Posição do pedal de acelerador (0-100%);
- Total de combustível utilizado;
- Nível de combustível;
- Giros do motor
- Peso nos eixos
- Total de horas do motor
- Versão do software da Interface-FMS
- Identificador do veículo
- Informações do tacógrafo
- Distância do veículo em alta resolução
- Temperatura do líquido refrigerante do motor

(BALDISSERA, 2011).

As fabricantes, visando evitar ligações realizadas diretamente no barramento do veículo e conseqüentemente, ocasionar a perda de qualquer garantia sobre o veículo, realizaram a padronização de um conector, visto que segundo a norma, a Interface FMS é descrita como o único local para uma conexão de dados segura para a rede interna do veículo (HODAC, 2004 apud BALDISSERA, 2011).

2.6 TECNOLOGIA TCP/IP

O TCP/IP (*Transmission Control Protocol/Internet Protocol*) é o conjunto de informações e padrões de comunicação, que fornece uma linguagem comum para a interconexão entre redes de computadores, e permite a utilização da tecnologia de comutação de pacotes, cuja finalidade é facilitar o transporte de informações (CAVALCANTI, 1997).

2.6.1 Arquitetura Internet

Apresentada pelo Departamento de Defesa do governo americano e escolhida como o padrão obrigatório de comunicação entre sistemas da época, a Arquitetura Internet atualmente é largamente utilizada para interconexão de sistemas computacionais heterogêneos.

Evidencia a grande simplicidade de implementação de seus protocolos, que mesmo assim atendem aos requisitos de interconexão exigidos pela maioria dos sistemas (BRISA, 1994).

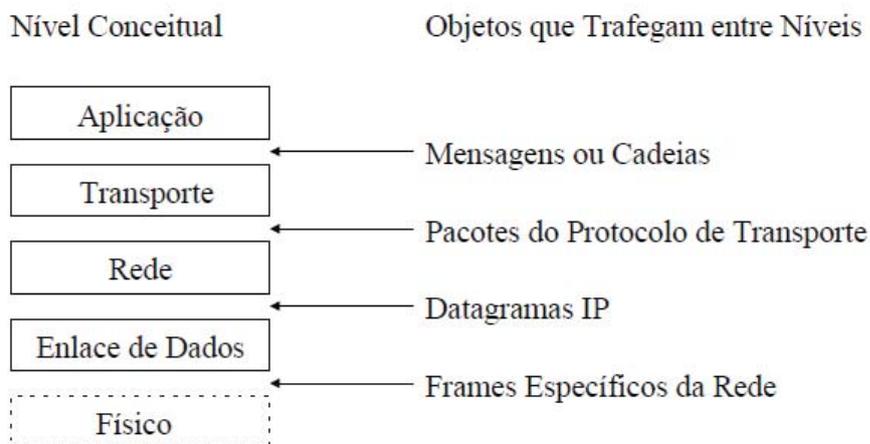
Diferentemente do Modelo de Referência OSI, apresentado na Seção 2.3, na Arquitetura Internet, buscou-se definir um protocolo próprio para cada camada, bem como a interface de comunicação entre duas camadas adjacentes, priorizando pela funcionalidade (BRISA, 1994).

Segundo Brisa (1994), a Arquitetura Internet é composta por dois protocolos principais: o IP (*Internet Protocol*), responsável pelo encaminhamento de pacotes de dados entre diversas sub-redes, e o TCP (*Transmission Control Protocol*), que por sua vez, é responsável pelo transporte fim-a-fim confiável de mensagem de dados entre dois sistemas. Logo, o conjunto TCP/IP, pode oferecer um serviço de alta confiabilidade.

2.6.2 O Modelo de Referência TCP/IP

De acordo com Comer (2000), a pilha de protocolos TCP/IP é organizada em cinco camadas, sendo quatro camadas de *software* construídas sobre uma quinta camada de *hardware*. A Figura 17 ilustra como as cinco camadas do Modelo TCP/IP estão dispostas.

Figura 17. Camadas do Modelo TCP/IP



Fonte: Adaptador de COMER, 2000

Chaves (2003) define as cinco camadas do Modelo TCP/IP da seguinte maneira:

Camada de aplicação: O nível mais alto, onde são executadas aplicações através de serviços disponíveis na rede TCP/IP. Aplicações interagem com a camada de transporte para o envio e recebimento de dados, definindo o modelo, seja por uma sequência de mensagens individuais ou uma cadeia contínua de bytes.

Camada de transporte: Responsável pelo estabelecimento e controle do fluxo de dados entre dois *hosts*. A camada de transporte fornece a comunicação entre as aplicações de maneira confiável, ou seja, garantir que as informações sejam entregues sem erros e na sequência correta.

Camada de Rede: Encarregada da comunicação entre os *hosts*. Essa camada aceita uma requisição de envio de pacote oriunda da camada de transporte, com a identificação do *host* para onde o pacote deve ser transmitido. Encapsula o pacote em um datagrama IP e preenche o cabeçalho do datagrama com os endereços lógicos de origem e destino. Determina também se o datagrama deve ser entregue diretamente, ou enviado para um *gateway*.

Camada de Enlace de dados: A camada mais baixa no nível de software, a camada de enlace é responsável por aceitar datagramas IP, encapsulá-los em *frames*, preencher o cabeçalho dos mesmos com os endereços físicos de origem e destino, dentre outros dados e transmiti-los para uma rede específica.

Camada física: Recebe os frames convertidos em sinais eletrônicos pela camada de enlace, e os conduz até a próxima interface de rede, que pode ser a do *host* de destino ou a do *gateway* da rede.

2.6.3 O Protocolo IP

Postel (2002) define o Protocolo IP como a base para os outros protocolos da pilha TCP/IP, como o TCP, abordado na seção 2.6.4, e apresenta, segundo Brisa (1994), as funções de transporte dos blocos de dados, denominados datagramas.

Datagramas IP, segundo Chaves (2003) podem ser definidos como blocos de dados transmitidos de uma determinada origem para um destino, de tal forma que origens e destinos são *hosts* identificados por endereços lógicos de tamanho fixo.

2.6.3.1 Características

O Protocolo IP, apresenta a possibilidade de fragmentar e reconstruir datagramas, permitindo transmitir em redes que suportem diferentes tamanhos por blocos de dados. Prove funções necessárias para compartilhar datagramas IP para um destino determinado em redes interconectadas. Ademais, não implementa controle de fluxo, sequenciamento ou tarefas de responsabilidade dos protocolos de níveis mais altos, como o TCP (CHAVES, 2003).

Porém, o protocolo incorpora a função de roteamento, ou seja, determina se o datagrama deve ser entregue diretamente ao seu destino, caso pertençam a mesma rede, ou entregue ao *gateway*, sendo esse o responsável por enviar o datagrama ao destino (CHAVES, 2003).

De acordo com Comer (2000), o IP baseia-se no conceito de entrega de datagramas “sem garantias”, logo, inclui um conjunto de regras sobre o processamento dos datagramas, como, quando e como uma mensagem de erro deve ser gerada e as condições nas quais os mesmos devem ser descartados.

2.6.4 O Protocolo TCP

Segundo Postel (1981), o TCP é um protocolo de comunicação que prove conexões entre máquinas (*hosts*), podendo estar ligadas a diferentes redes de computadores (BRISA, 1994) de forma confiável, ou seja, é um protocolo orientado à conexão. Stevens (1994), define que o termo “orientado à conexão” significa que duas aplicações, utilizando um protocolo que detém esta característica, devem estabelecer uma conexão bidirecional, antes de efetuar a troca de dados.

A princípio, de acordo com Brisa (1994), o TCP deve ser capaz de operar sobre um largo espectro de sistemas de comunicação, desde linhas ponto-a-ponto até redes comutadas por pacotes.

2.6.4.1 Características

O TCP é um protocolo confiável, pois antes de um *host* enviar dados a outro, ocorre o reconhecimento relativo à chegada de dados (CHAVES, 2003). Utilizando a técnica *sliding window*, o protocolo realiza o controle de fluxo durante as conexões, evitando sobrecarga dos processos durante a recepção de dados. Além disso, o TCP exerce o controle de erros sobre os dados recebidos através da utilização de *checksum* (BRISA, 1994).

De acordo com Brisa (1994), para um *host* utilizar os serviços TCP, primeiramente ele se associa a uma porta de serviço TCP, que quando ligada a um endereço IP, constitui um *socket*. Similarmente a um endereço IP que um *host* recebe em uma rede internet, único e exclusivo, dentro de uma estação, cada porta é identificada por um número único. Portanto, o *socket* possui uma identificação única dentro de uma rede internet.

2.6.4.2 Mensagens

Para a troca de dados entre estações utilizando o protocolo TCP, é utilizada a unidade chamada de segmentos, os quais definem o estabelecimento da conexão, a transferência de dados, envio de ACK e encerramento da conexão (BRISA, 1994).

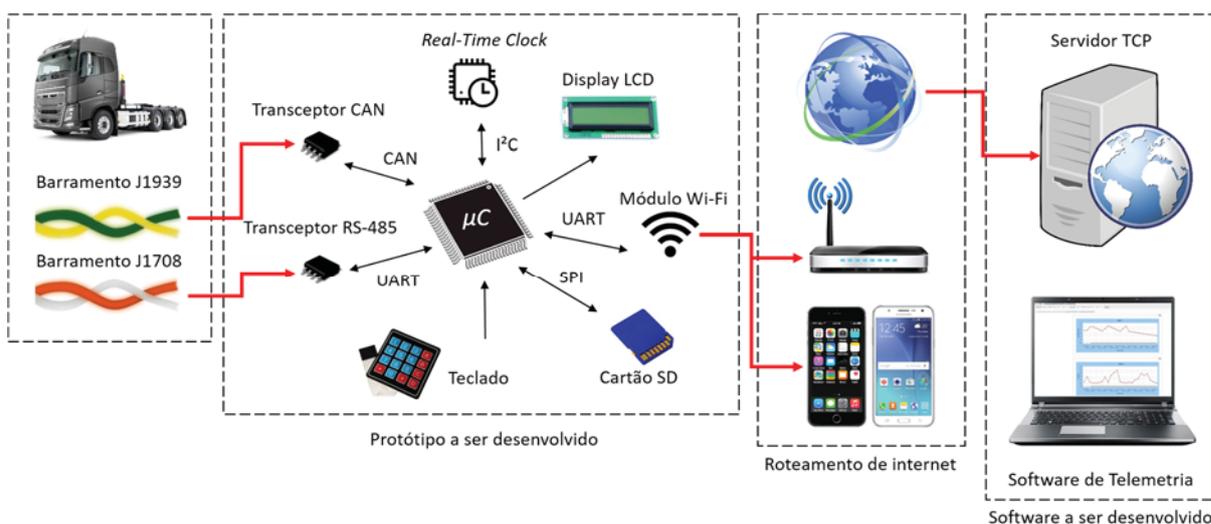
3 DESENVOLVIMENTO

O projeto advém da necessidade em reduzir custos da Cielo Telecom com o deslocamento de engenheiros para realização de manutenções que envolvam a análise dos dados presentes no barramento de dados veiculares, visto que essa operação envolve custos com diárias, além do tempo perdido para a solução. Dessa forma, espera-se tornar essa análise remota, e conseqüentemente aumentar a velocidade na solução de problemas, uma vez que ao invés de deslocar um ou dois engenheiros, remotamente essa atividade pode ser realizada por uma equipe inteira.

3.1 ESPECIFICAÇÃO DO PROJETO

O dispositivo é conectado aos barramentos J1939 e/ou J1708 de veículos pesados. A partir da conexão, o usuário selecionará pelo teclado a opção em realizar a transmissão em leitura para um servidor TCP via internet, através de um roteador Wi-Fi, como por exemplo um celular, ou salvá-la em um Cartão SD (*Secure Digital*). Na Figura 18, é apresentado o diagrama do projeto, desde a conexão com o caminhão e seus barramentos até as interfaces de comunicação com o microcontrolador e hardwares de saída (Display, Módulo Wi-Fi e Cartão SD).

Figura 18. Diagrama e contextualização do projeto



Fonte: Próprio Autor

3.2 HARDWARE

O hardware do Dispositivo de Telemetria Veicular Multiprotocolo com Transmissão via Internet é responsável por intermediar todas as informações entre o veículo e o software de decodificação e apresentação.

Basicamente, suas principais funcionalidades incluem: ler as informações dos barramentos CAN J1939 e J1708 através dos respectivos transceptores; transmitir as informações via Wi-Fi até o roteador conectado para que seja encaminhado via internet para o software; e salvar as informações no cartão SD. Sendo que todas essas tarefas são comandadas pelo teclado e apresentadas no display. No Apêndice A é apresentado o esquemático completo do dispositivo.

3.2.1 MICROCONTROLADOR

Para atender a todas as necessidades de desenvolvimento do dispositivo, e analisando a Figura 18 é necessário utilizar um microcontrolador que apresente as seguintes características:

- Suporte ao Protocolo CAN 2.0B;
- Dois módulos USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) para interface RS-485 e com o Módulo Wi-Fi;
- Comunicação I²C (*Inter-Integrated Circuit*) para interface com o RTC;
- Comunicação SPI (*Serial Peripheral Interface*) para interface com o Cartão SD;
- Grande quantidade de pinos, visto que serão implementados um Teclado e um Display para interface com o usuário, além de outros pinos utilizados para as interfaces de comunicação;
- Quantidade suficiente de memória de programa para desenvolvimento do firmware, pois diversos módulos serão utilizados.

Com as características do microcontrolador a ser escolhido, foi optado pelo microcontrolador PIC18F66K80, o qual cumpre com todas as características solicitadas, possuindo 64 pinos e 64 kbytes de memória de programa, o suficiente para a implementação do firmware.

3.2.2 TRANSCEIVERS

Como o microcontrolador PIC18F66K80 opera apenas com sinais TTL (*Transistor-Transistor Logic*) em seu barramento, é necessário, para a realização da comunicação externa a utilização de *transceivers* (transceptores), componentes responsáveis pela implementação da camada física para vários padrões de comunicação e barramentos seriais (MAXIM INTEGRATED, 2016).

A utilização dos *transceivers* no projeto se faz necessária, posto que o protocolo RS-485, utilizado no padrão SAE J1708, e o protocolo CAN, utilizado no SAE J1939, operam em níveis de tensões diferentes ao TTL utilizado pelo microcontrolador.

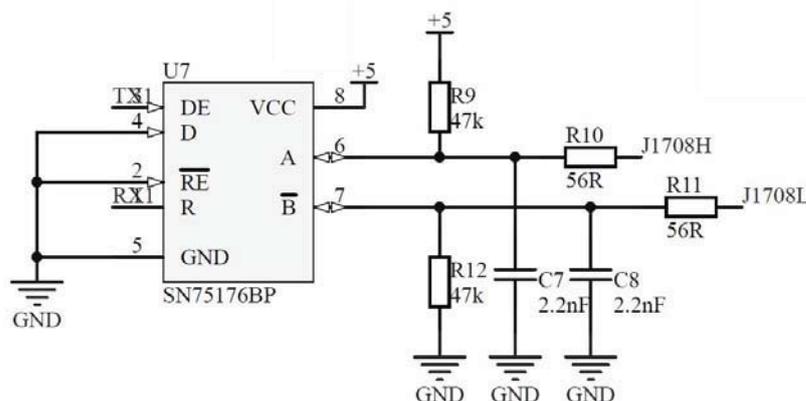
Portanto, para cada protocolo foi utilizado um *transceiver* específico, conforme o protocolo utilizado. Para o padrão J1708, foi utilizado o *transceiver* SN75176BP da Texas Instruments. Para o padrão J1939, que opera sobre o protocolo CAN, foi utilizado o *transceiver* MCP2551 da Microchip.

3.2.2.1 SN75176BP

Esse dispositivo é um *transceiver* que atende os requisitos da norma ANSI TIA/EIA-485-A, a qual define as características elétricas do padrão RS-485.

O circuito integrado apresenta limitador de corrente e desligamento automático por elevação de temperatura. Suporta em seu barramento a inserção de até 32 dispositivos operando na velocidade de até 10 Mbps. É disponibilizado em encapsulamento de 8 pinos. Na Figura 19 é ilustrado o esquemático de ligação do SN75176BP no circuito desenvolvido.

Figura 19. Esquemático de ligação do SN75176BP.



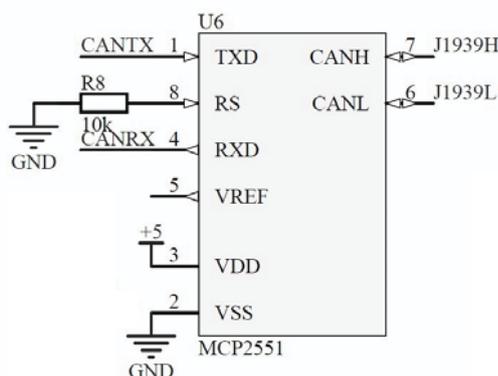
Fonte: Próprio Autor

3.2.2.2 MCP2551

O circuito integrado MCP2551 é um *transceiver* CAN de alta velocidade, que opera com taxas de transmissão de até 1Mb/s, e cumpre os requisitos de camada física descritos na norma ISO 11898, embora para o circuito utilizado no projeto, não foi utilizado o resistor de 120Ω, em virtude que o dispositivo operará como um *sniffer*, ou seja, apenas analisando os dados presentes no barramento, sem alterar suas características de impedância.

O MCP2551 é oferecido no encapsulamento de 8 pinos, possui proteção contra surtos de tensão e curto-circuito no barramento e desligamento automático ocasionado por elevação da temperatura além de suportar a conexão de até 112 nós na rede, compatível com sistemas de 12V e 24V. O circuito utilizado é ilustrado na Figura 20.

Figura 20. Esquemático de ligação do MCP2551.



Fonte: Próprio autor

3.2.3 MÓDULO WI-FI ESP8266

Para a transmissão da leitura em tempo de execução, foi optado pela tecnologia Wi-Fi, visto que é uma tecnologia de fácil acesso, e a maioria dos *smartphones* pode operar como um roteador, fornecendo, assim, conexão do dispositivo ao servidor TCP desejado para a transmissão dos dados.

Com a expansão da ideologia IoT (*Internet of Things*), novos dispositivos conectáveis foram apresentados no mercado como o Onion Omega2® e o Openpicus FlyportPRO®; todavia, o que mais se destacou foi o ESP8266, da empresa Espressif: um dispositivo acessível e muito bem difundido no meio eletrônico, devido à sua facilidade de operação e alto rendimento.

O ESP8266 é um SoC (*System-on-a-chip*) com tecnologia Wi-Fi embarcada disponibilizado em ampla variedade de modelos. Entretanto, o escolhido para o projeto é o ESP-01 por apresentar as seguintes características:

- Conexão ao microcontrolador pela interface UART, utilizando comandos AT;
- Tensão de operação de 3.3 V;
- Alcance de até 90 metros;
- Compacto, contendo apenas 8 pinos e antena no próprio encapsulamento;
- Alta disponibilidade no mercado.

Nas Figura 21 Figura 21 é apresentado o Módulo ESP8266 ESP-01:

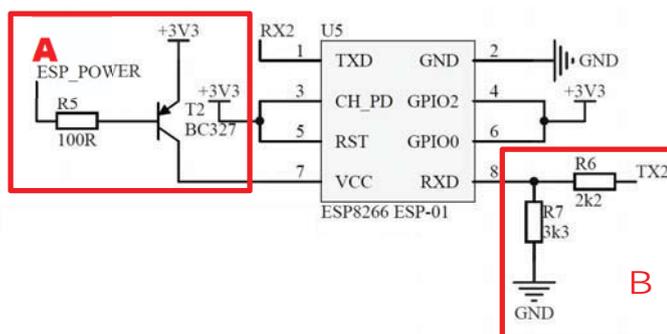
Figura 21. Módulo Wi-Fi ESP8266 ESP-01



Fonte: Próprio autor

Por ser alimentado com tensão 3.3V, suas saídas e entradas operam nesse nível de tensão. Logo, como o microcontrolador opera com 5V, foi utilizado um divisor de tensão para rebaixar o nível de tensão sinal enviado do microcontrolador ao módulo, evitando eventuais danos (Figura 22-B). Também foi utilizado um transistor BC327 para ligar e desligar o módulo (Figura 22-A), conforme ilustrado na Figura 22.

Figura 22. Esquemático de ligação do ESP8266 ESP-01: A – Acionamento com transistor; B – Divisor resistivo.



Fonte: Próprio Autor

3.2.4 RTC

O RTC (*Real-time clock*), é um dispositivo que, após alimentado, mantém o controle de tempo. Geralmente é apresentado no formato de circuito integrado, apresentando baixo consumo de energia e interface para microcontroladores.

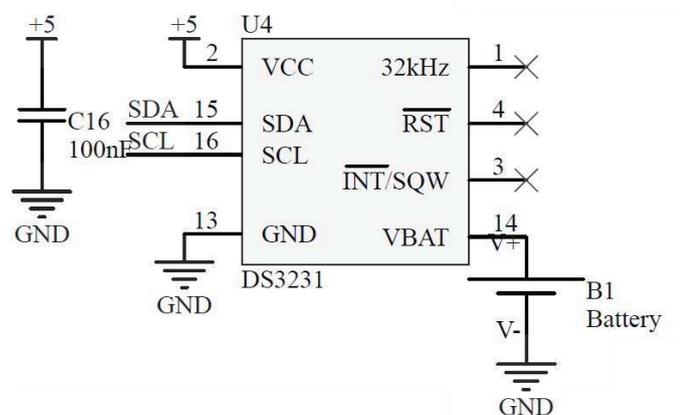
Para o desenvolvimento do projeto, foi considerada a utilização devido à necessidade em saber o momento exato em que foi realizada a telemetria, principalmente quando essa é realizada localmente, sendo salva no cartão de memória. Devido a disponibilidade no mercado e precisão foi selecionado o RTC Maxim DS3231.

3.2.4.1 Maxim DS3231

O DS3231 é um RTC de baixo custo, com compensação da frequência do cristal oscilador interno por temperatura, resultando em uma enorme precisão, aproximadamente 2ppm (partes por milhão) em condições adequadas de temperatura (0°C a 40°C), ou seja, uma variação de aproximadamente 0.18 segundos por dia (MAXIM INTEGRATED, 2016).

Disponibiliza além da data e hora, a temperatura e a possibilidade de programar dois alarmes e uma onda quadrada de saída. Possui também o chaveamento automático para a bateria reserva quando uma falta de alimentação é detectada, e utiliza uma interface I²C de alta velocidade (400 kHz). Na Figura 23 é apresentado o esquema de ligação do DS3231 no circuito utilizado.

Figura 23. Ligação DS3231.



Fonte: Próprio autor.

3.2.5 CARTÃO SD

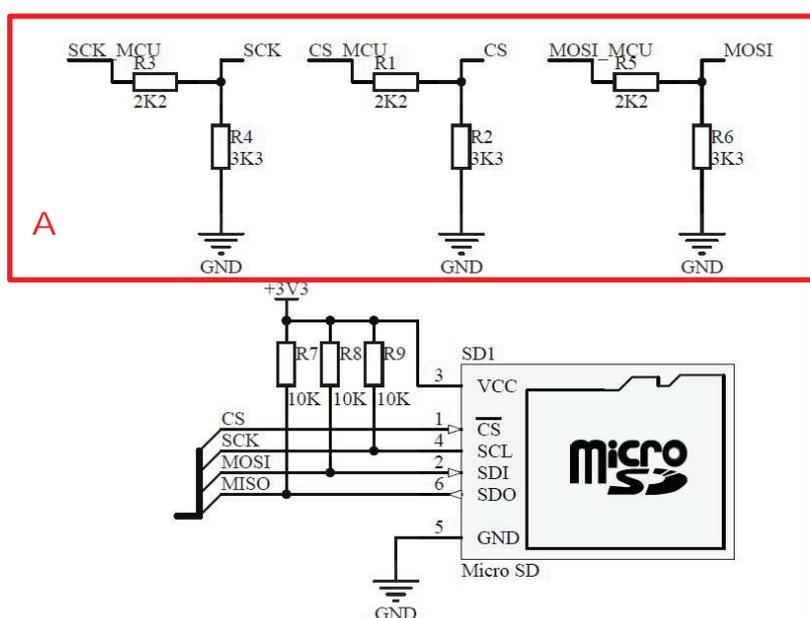
O Cartão SD (*Secure Digital*) é resultado do grupo fundado em janeiro de 2000 pela Panasonic, SanDisk e Toshiba, com o objetivo de estabelecer padrões e facilitar o desenvolvimento da tecnologia de um cartão de memória, esse chamado de *SD Memory Card* (NAKANO, 2014).

Segundo Nakano (2014), a tecnologia *SD Card* evoluiu a tal ponto em que a maioria dos dispositivos que necessitem armazenamento e transferência de arquivos (*smartphones*, câmeras de foto e vídeo, computadores, *tablets*) possuem suporte para tal tecnologia de maneira acessível.

O uso do Cartão SD no projeto se faz necessário devido à falta de acesso à internet no momento da realização da leitura. Com o seu uso, a leitura pode ser efetuada e posteriormente analisada no software.

A interface do Cartão SD com o microcontrolador é realizada por meio da comunicação SPI, o qual utiliza níveis de tensão de 3.3V, portanto foi utilizada a mesma fonte desenvolvida para o módulo Wi-Fi além da redução do nível de tensão dos pinos de comunicação a fim de evitar eventuais danos. Na Figura 24 é ilustrada ligação do soquete do Cartão SD com o microcontrolador e seus divisores resistivos (Figura 24-A) para assegurar o nível de tensão adequado.

Figura 24. Esquemático de ligação do soquete do Cartão SD: A – Divisores de tensão resistivos.



3.2.6 INTERFACE COM O USUÁRIO

Visando um melhor controle das operações a serem realizadas, como digitação dos parâmetros da rede e do servidor o qual será conectado o dispositivo, são necessários hardwares para a interface com o usuário. O teclado e o *display* são dispositivos típicos para essa aplicação, pois permitem o controle e visualização em tempo real das ações tomadas pelo usuário do dispositivo.

Logo, para a visualização das operações foi optado por um *Display* LCD 16x4, o qual contém área suficiente para a apresentação de menus e de funções disponíveis. Devido ao desenvolvimento em conjunto com a empresa Cielo Telecom, foi utilizada uma caixa adaptada para esse fim, contendo Display e teclado, conforme apresentado na Figura 25.

Figura 25. Caixa do protótipo com Display e teclado.



Fonte: Próprio autor

3.2.7 FONTES DE ALIMENTAÇÃO

A maioria dos circuitos eletrônicos apresenta uma ou mais fontes de alimentação, visto que é necessário adaptar os níveis de tensão para os dispositivos a serem energizados.

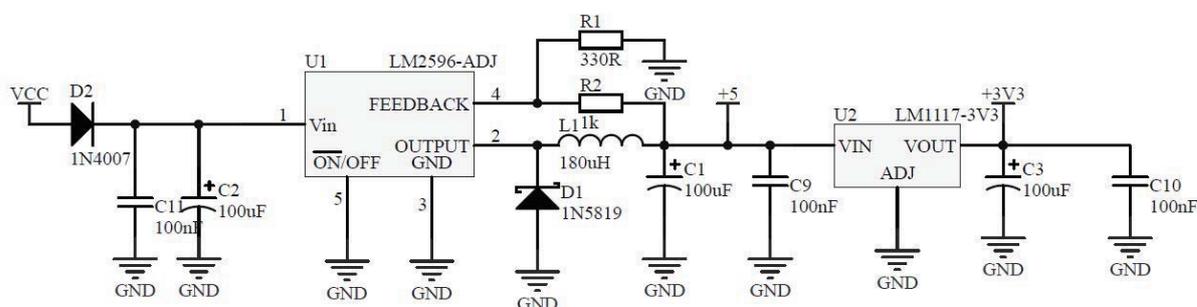
O protótipo, por apresentar variados componentes de hardware, inclusive alguns necessitando alimentação com diferentes níveis de tensão (3.3V e 5V), é composto de duas fontes de alimentação, nos níveis de tensão especificados anteriormente.

Como veículos pesados apresentam saídas de tensão de 12V e 24V, e o circuito apresenta um consumo relativamente alto, ocasionado principalmente pelo Display LCD, que pode atingir um consumo de 264 mA (AGTECHNOLOGIES, 2015), e do Módulo Wi-Fi que pode consumir até 215 mA (ESPRESSIF SYSTEMS, 2013), a utilização de um regulador linear, ao menos como a fonte principal não apresenta nenhuma vantagem, pois a potência dissipada é dada pela relação da diferença de tensão entre a entrada e a saída sobre a corrente consumida. Portanto, para utilização como fonte principal, responsável por alimentar todos os componentes com 5V e a fonte de alimentação 3.3V, a qual suprirá os outros componentes desse nível de tensão, foi optado por um regulador chaveado.

Reguladores chaveados são caracterizados por apresentarem alta eficiência combinada a uma alta corrente de saída, requisitos desse projeto. O regulador escolhido foi o LM2596-ADJ da Texas Instruments, ajustado em 5V, o qual fornece correntes de até 3A com eficiência energética próxima de 80% (TEXAS INSTRUMENTS, 2016).

Já para a fonte de alimentação 3.3V foi selecionado o regulador linear LM1117 da Texas Instruments, circuito integrado que fornece corrente suficiente para alimentar os componentes 3.3V. O esquemático das duas fontes de alimentação é apresentado na Figura 26.

Figura 26. Esquemático das fontes de alimentação.

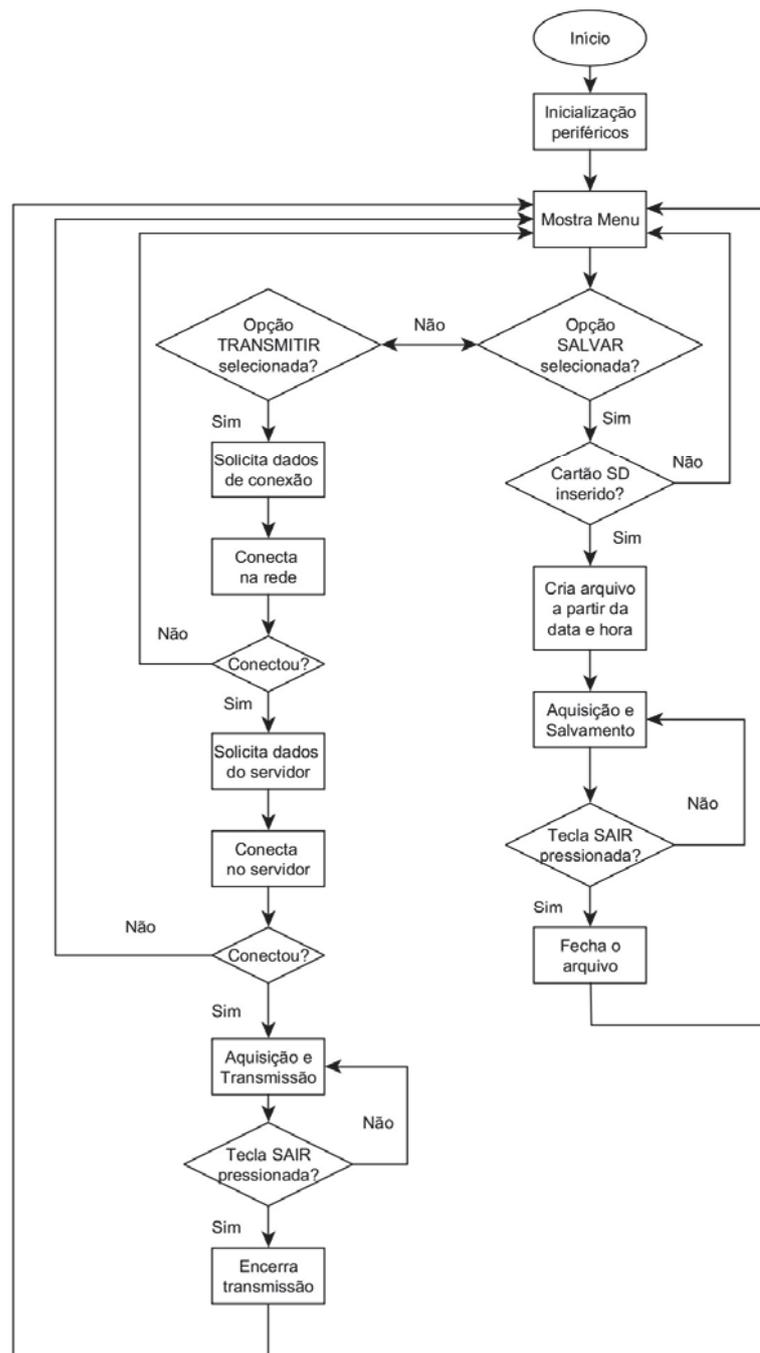


Fonte: Próprio autor

3.3 FIRMWARE

O firmware é responsável pela configuração dos periféricos utilizados para a realização da leitura dos parâmetros no veículo, inicialização da conexão com o servidor TCP e salvamento da leitura no cartão de memória quando não for estabelecida a conexão. Nas Figura 27, é apresentado o fluxograma de funcionamento, seguido por suas explicações detalhadas.

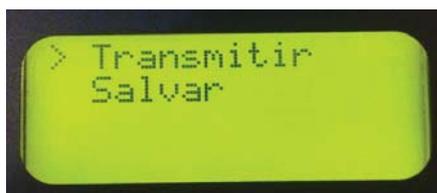
Figura 27. Fluxograma geral do firmware.



Fonte: Próprio Autor

A inicialização dos periféricos configura todos os módulos a serem utilizados pelo microcontrolador como: GPIO, CAN, I2C, SPI, UART, *Timers*, Display e Interrupções. Após essa inicialização é apresentado o Menu, onde o usuário deve escolher entre as opções: Transmitir ou Salvar, sendo responsáveis pela operação remota e local, respectivamente. A tela de menu é apresentada na Figura 28.

Figura 28. Tela de menu.



Fonte: Próprio autor

3.3.1 Operação Remota

Selecionando a opção “Transmitir” o usuário é direcionado para a janela onde deverá informar os dados do roteador a se conectar – nome da rede e senha. A operação é similar à realizada em computadores, embora para o dispositivo do projeto, além da senha o usuário deve informar o nome da rede na tela apresentada na Figura 29.

Figura 29. Tela de informações do roteador.



Fonte: Próprio autor

Para as operações de digitação foi implementada a função de teclado alfanumérico com símbolos, selecionáveis a partir de tecla dedicada no teclado, apresentada na Figura 30.

Figura 30. Tecla de mudança de teclado.



Fonte: Próprio autor

O modo de digitação pode ser acompanhado a partir do identificador no canto superior da tela, sendo eles:

- A – Letras maiúsculas;
- a – Letras minúsculas
- 1 – Numérico e símbolos.

Após digitação dos dados da rede Wi-Fi, clicando “Enter” é realizada a operação de conexão ao roteador. Se a opção é realizada com sucesso a próxima tela, conexão ao servidor é apresentada. Nessa tela é solicitado ao usuário digitar o IP e a Porta do servidor.

Cabe salientar que se utilizado um servidor com IP e porta fixa, essa janela pode ser eliminada, já realizando a operação de conexão ao servidor. Com a conexão estabelecida, os dados são adquiridos e transmitidos conforme será explicado na Seção 3.3.3.

3.3.2 Operação Local

A operação local é muito similar à remota, diferindo apenas no que tange ao meio de transmissão. Onde na anterior, os dados eram transmitidos do dispositivo ao servidor através da internet. Na operação local, os dados são gravados em um arquivo criado em um cartão de memória.

3.3.3 Aquisição dos Dados

A etapa mais importante do projeto, juntamente com a decodificação realizada pelo software é a aquisição dos dados, pois esse processo coordena a ordem das aquisições, controla os *buffers* de entrada e saída, e prevê o funcionamento contínuo mesmo com erros de recepção, mensagens corrompidas e ausência de um dos barramentos.

Todo o processo de aquisição é controlado por um temporizador, visando sistematizar a recepção, transmissão ou salvamento. Para cancelamento da operação, conforme apresentado no fluxograma da Figura 27, é necessário pressionar a tecla “Sair” do teclado. Portanto, enquanto a tecla não é pressionada, a operação ocorre continuamente. Para ambos os modos de operação (Remoto e local), o processo de aquisição é o mesmo.

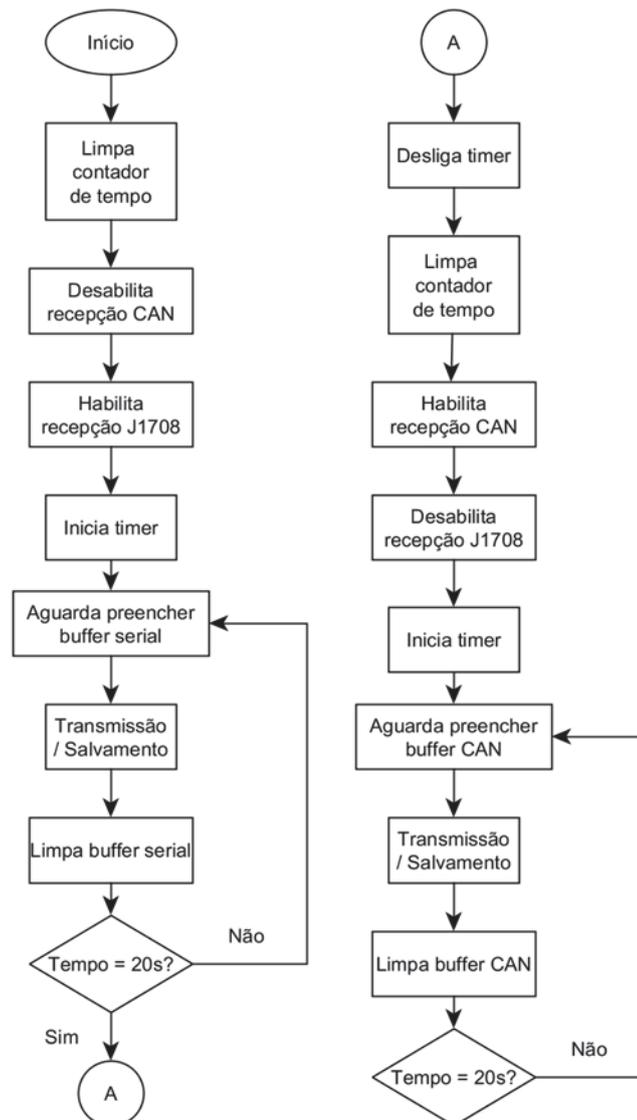
Por opção, foi adotado iniciar a recepção dos dados J1708: O processo inicia limpando os valores do *Timer*, desabilitando a recepção CAN J1939 e habilitando a recepção serial J1708. A inicialização do temporizador dá início ao processo, que durará 20 segundos, tempo

suficiente para coletar dados necessários, transmitir ou salvar, dependendo da função selecionada. Não necessariamente esse processo (aquisição e transmissão/salvamento) ocorrerá uma vez, já que o processo ocorre por tempo e não quantidade de dados.

Cabe salientar que, para cada fim de processo, o *buffer* é limpo, permitindo que mais dados cheguem ao dispositivo.

Passados os 20 segundos, o temporizador é desligado, e a recepção é alternada para a CAN e novamente, após a inicialização do *Timer* o processo ocorre com duração de 20 segundos, e ficará alternado entre J1708 e CAN, enquanto o usuário não pressionar a tecla “Sair”. A Figura 31 apresenta o fluxograma do processo de aquisição.

Figura 31. Fluxograma da aquisição.



3.4 SOFTWARE

A partir da necessidade em transmitir os dados obtidos e realizar o monitoramento, foi desenvolvido um software que opere como servidor TCP e apresente ao usuário, que acompanhará remotamente, os dados presentes no barramento de dados do veículo, sejam eles na forma como são transmitidos (codificados), e na forma decodificada e em gráficos, de acordo com as normas SAE J1939-71 e SAE J1708/1587. O software é capaz de importar os arquivos de leitura do cartão de memória, e apresentá-los em gráficos e valores decodificados. Finalmente, foi implementada as funções de exportação dos *logs* gerados.

Todo software foi desenvolvido utilizando a linguagem de programação C#. A seguir, são apresentadas algumas características do software, já no Apêndice B é apresentado a tela principal do software.

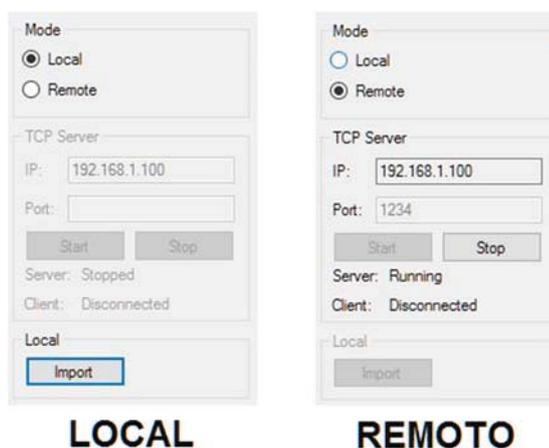
3.4.1 Modos de operação

Da mesma maneira que o dispositivo, o software também opera em dois modos: local e remoto. Sendo essa a primeira configuração a ser feita para obter os dados do veículo.

Selecionando o modo local, o próximo passo é importar o arquivo gerado pelo dispositivo clicando na opção “Import”. Caso a opção selecionada seja a “Remote”, o IP é carregado automaticamente, incumbindo o usuário a informar a porta do servidor a ser iniciado. Esses dados devem coincidir com os dados do servidor digitados no dispositivo.

Se os dados estão de acordo, o servidor pode ser iniciado e aguardar a conexão do dispositivo, para o início da telemetria. Na Figura 32 são apresentadas as duas telas de operação.

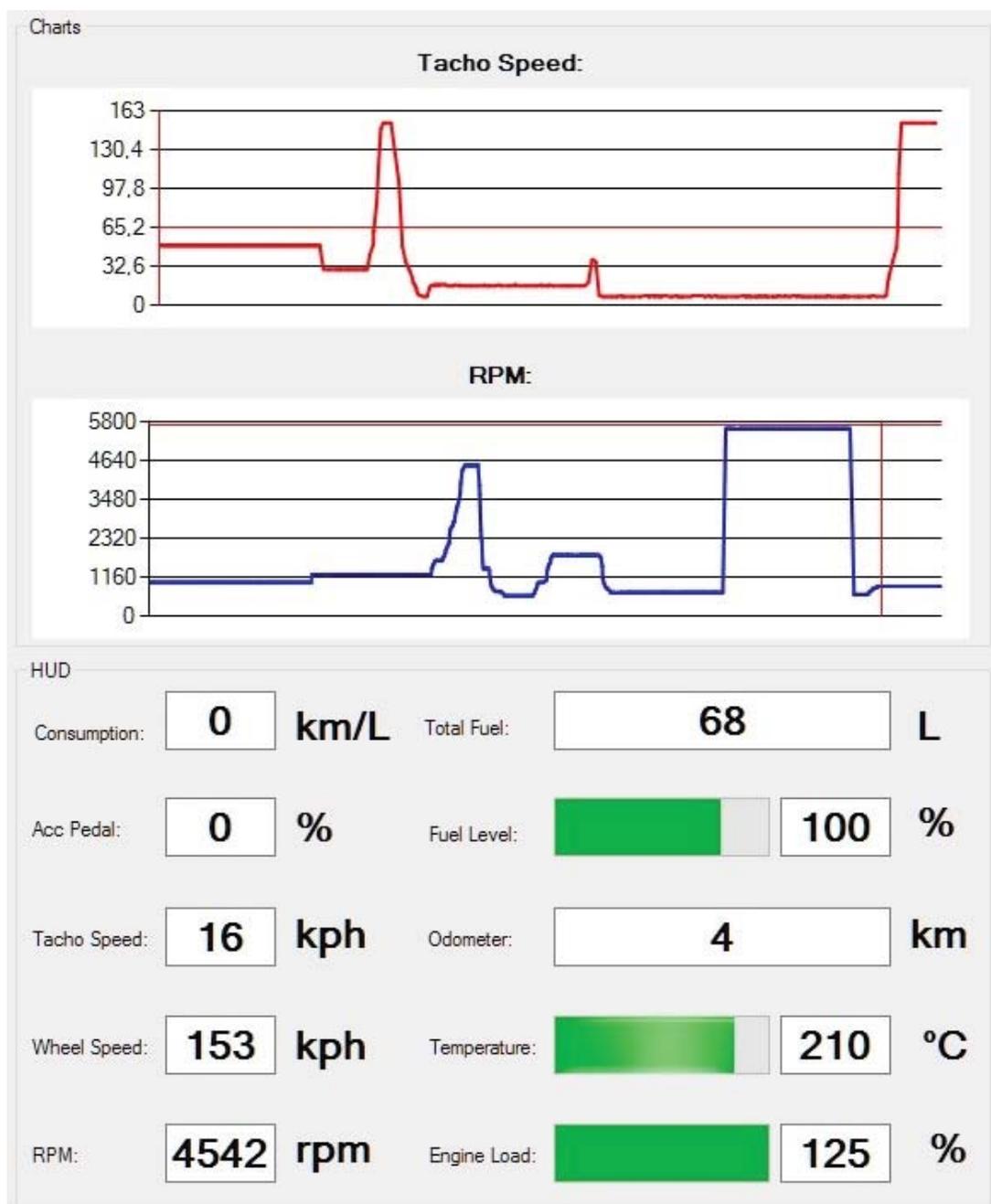
Figura 32. Modos de operação do software.



3.4.3 Gráficos e decodificação

Para fornecer uma visão interativa dos dados adquiridos durante a telemetria, foram desenvolvidas as telas de gráficos e decodificação, que apresentam os parâmetros definidos na Seção 2.1.3 em suas respectivas unidades. Os valores apresentados são decodificados a partir das normas J1939/71 e J1708/J1587. Na Figura 34 é apresentada a janela de gráficos e valores decodificados.

Figura 34. Gráficos e valores decodificados.



Fonte: Próprio autor

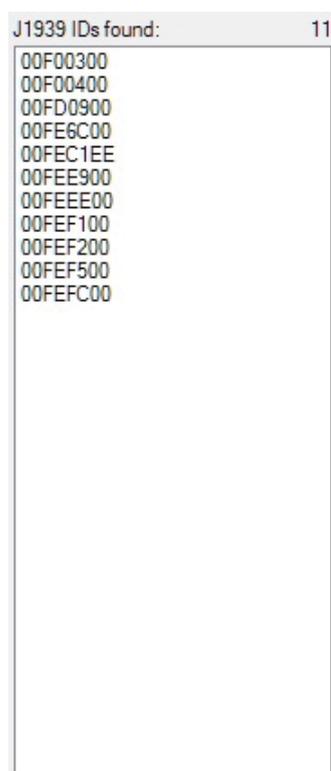
3.4.4 PGNs encontrados

Visando facilitar a identificação para o usuário dos parâmetros que estão sendo obtidos foi desenvolvida uma janela que apresenta todos os PGNs J1939 encontrados durante a telemetria.

Essa visualização é útil para o conhecimento dos PGNs disponíveis em cada modelo de veículo, posto que mesmo veículos da mesma marca ou modelo apresentam diferenças nos parâmetros disponíveis em seu barramento de dados.

Portanto, através dessas informações, juntamente com a norma J1939-71 é possível analisar parâmetros desconhecidos. Um exemplo dessa janela é apresentado na Figura 35.

Figura 35. PGNs encontrados.



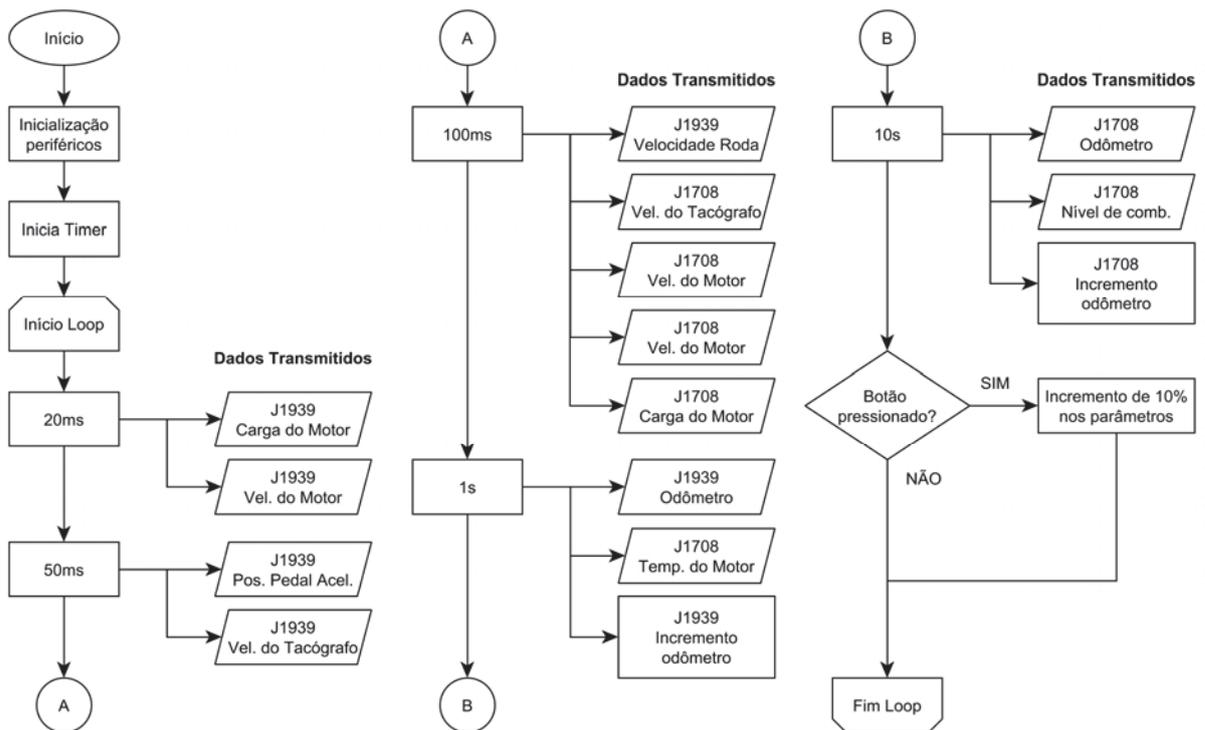
Fonte: Próprio autor

3.5 SIMULADOR J1939 E J1708

Visando e auxiliar o desenvolvimento e os testes do protótipo foi desenvolvido um Simulador J1939 e J1708, dispositivo que cumpre a função de simular um barramento ativo de um veículo pesado, transmitindo informações em conformidade com as normas SAE J1939, SAE J1708 e Interface FMS.

O simulador opera em dois modos: estático e dinâmico, sendo selecionados a partir de um botão na lateral da caixa do dispositivo. O primeiro transmite dados fixos no barramento enquanto o dispositivo estiver ligado. Já o segundo modo, a cada 10 segundos os parâmetros de velocidade do tacógrafo, rodas e motor são incrementados em 10%. Cabe salientar que o parâmetro do odômetro está sempre em constante atualização de acordo com a velocidade do tacógrafo, independentemente do modo utilizado, tornando assim os dados confiáveis para a simulação. A transmissão de dados explicada acima é apretnada no fluxograma da Figura 36.

Figura 36. Fluxograma de funcionamento do Simulador J1939/J1708



Fonte: Próprio autor

No simulador são transmitidos os principais parâmetros veiculares, os quais são mais populares nos veículos pesados. Porém, por ser uma ferramenta de auxílio em desenvolvimento, novos parâmetros podem ser incluídos a partir de novas atualizações. Os parâmetros disponibilizados pelo simulador são apresentados na Tabela 3.

Tabela 3. Parâmetros disponíveis no Simulador J1939 e J1708.

Parâmetros disponibilizados	J1939	J1708
Odômetro	X	X
Velocidade do tacógrafo	X	X
Velocidade do motor	X	X
Nível de combustível	-	X
Pedal de acelerador	X	-
Carga do motor	X	X
Temperatura	X	-
Total de combustível	-	X
Velocidade da roda	X	-

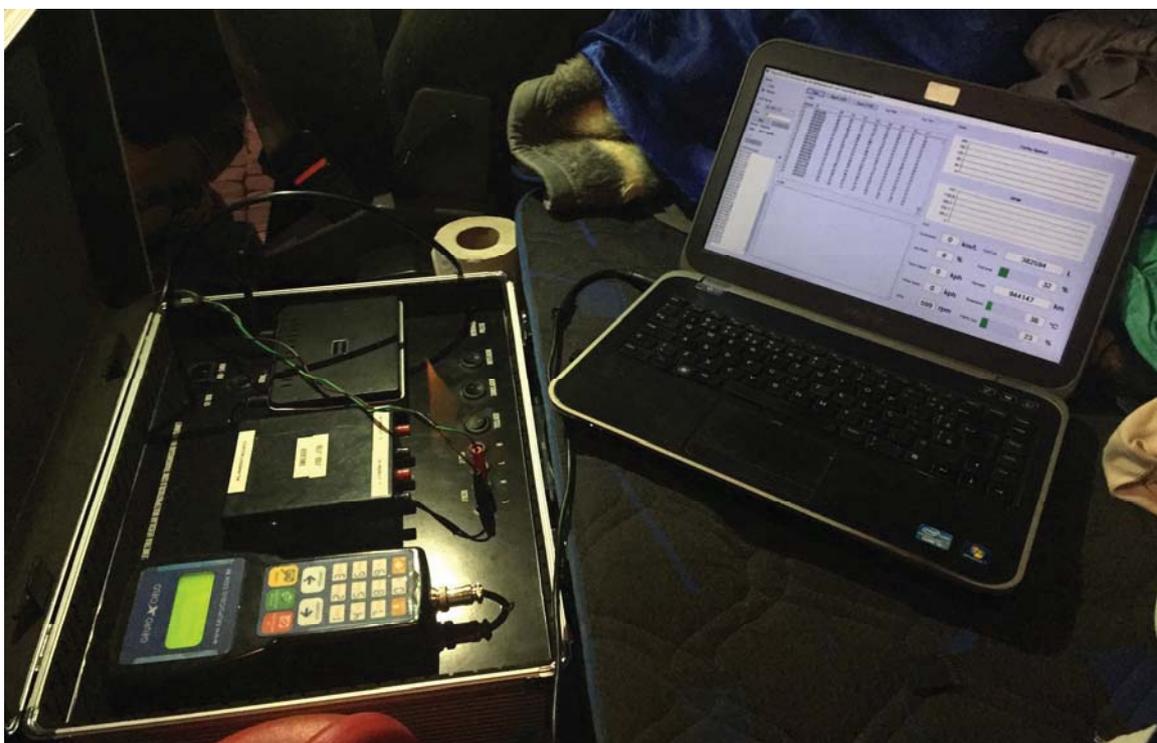
Fonte: Próprio autor

Em um primeiro momento o Simulador fornece dados J1939 e J1708, pois em veículos pesados, são os protocolos mais comuns. No entanto, outros protocolos podem ser utilizados, desde que seja realizada uma atualização no hardware utilizado, pois o atual foi desenvolvido unicamente para protocolos CAN e J1708. O esquemático elétrico do Simulador J1939 e J1708 se encontra no Apêndice C.

4 RESULTADOS

A validação do protótipo desenvolvido é obtida a partir do correto funcionamento das três etapas seguintes: conexão à internet e ao servidor; transmissão dos parâmetros obtidos nos veículos e simulador; e a decodificação realizada no software. Todas as ações são visualizadas no próprio software, pois o mesmo, realiza a interface de funcionamento do dispositivo. Para os testes realizados em campo foram utilizados os veículos Volvo FH 440 e Scania G380. Na Figura 37 é apresentado o protótipo em funcionamento, realizando a telemetria em um Scania G380.

Figura 37. Teste do protótipo



Fonte: Próprio autor

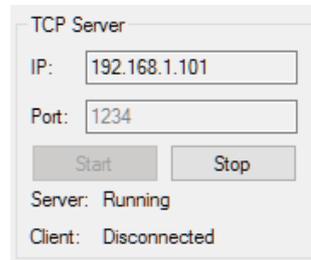
4.1 CONEXÃO A INTERNET E SERVIDOR

A conexão é realizada a partir do dispositivo, seguindo os passos descritos na Seção 3.3.3 e acompanhado no software, pois o mesmo informa quando o dispositivo se conectou ao mesmo.

Primeiramente o software é inicializado na operação remota. O IP adquirido é carregado automaticamente, restando ao usuário apenas informar a porta e iniciar o servidor clicando em

“Start”. Na mesma janela é apresentado o status do servidor e se a conexão com o dispositivo foi estabelecida ou não. A Figura 38 ilustra essa tela.

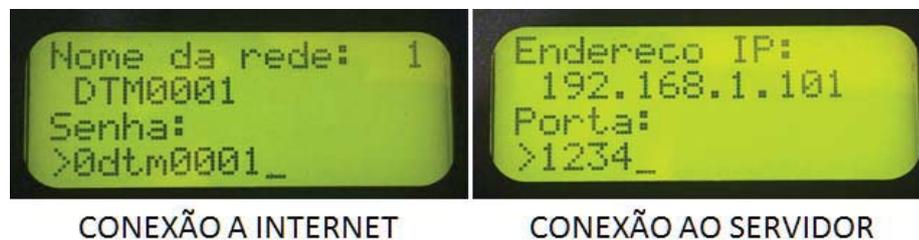
Figura 38. Servidor iniciado com dispositivo desconectado



Fonte: Próprio autor

No dispositivo o usuário é encarregado de digitar a nome e a senha da rede em que deseja se conectar e, caso a conexão seja bem-sucedida, informar o endereço IP e a porta, conforme apresentado no software. Na Figura 39 são apresentadas essas etapas realizadas no dispositivo.

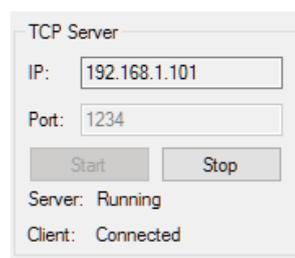
Figura 39. Conexão a internet e ao servidor



Fonte: Próprio autor

Caso os dados estejam de acordo, a conexão irá ocorrer corretamente e o software informará que o dispositivo se conectou e a telemetria poderá ser iniciada, conforme ilustrado na Figura 40.

Figura 40. Servidor iniciado com dispositivo conectado

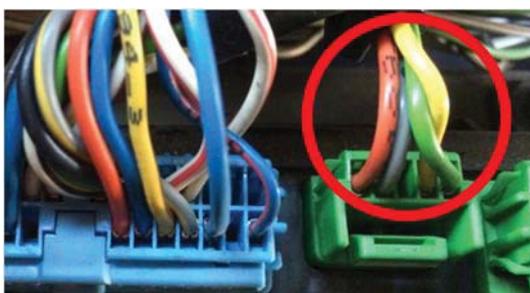


Fonte: Próprio autor

4.2 OBTENÇÃO E TRANSMISSÃO DOS PARÂMETROS

Para obter os parâmetros veiculares, o dispositivo é ligado diretamente ao barramento de dados do veículo, usualmente pelo conector de diagnostico presente na maioria dos veículos, porém, alguns dados não são disponibilizados a partir dessa conexão. Logo a ligação será feita diretamente aos fios J1708 e J1939, presentes sob a caixa de fusíveis do veículo, conforme apresentado na Figura 41.

Figura 41. Localização fios J1708 (laranja e cinza) e J1939 (verde e amarelo)



Fonte: Próprio autor

Após a ligação nos barramentos de dados, o software em execução e o dispositivo conectado, a telemetria é iniciada, e os dados coletados são apresentados na interface do software. Foi buscado através da telemetria, obter os parâmetros descritos na Seção 2.1.3, independente do protocolo de comunicação utilizado no veículo. Na Tabela 4 são apresentados em qual dos protocolos foram encontrados os parâmetros desejados.

Tabela 4. Parâmetros encontrados e seus protocolos

Parâmetro	Volvo FH 440	Scania G380
Vel. Tacógrafo	J1939/J1708	J1939
Vel. Rodas	J1939	J1939
Vel. Motor	J1939/J1708	J1939
Odômetro	J1939/J1708	J1939
Temp. Liq. Refrig. Motor	J1708	J1939
Posição pedal acelerador	J1939/J1708	J1939
Cons. Combustível	J1708	J1939
Nível combustível	J1708	J1939
Carga do motor	J1939/J1708	J1939

Fonte: Próprio autor

Além dos parâmetros desejados, analisando detalhadamente os arquivos de telemetria, outros foram encontrados, como:

- Horímetro (relógio);
- Temperatura do óleo da transmissão e ambiente;
- Horas de uso do motor;
- Marcha atual;
- Chave do pedal de freio e embreagem;

Foi observado que dentre os veículos testados, apenas o da marca Volvo possui o barramento de dados J1708. Ademais é ressaltado a importância da análise detalhada dos dados obtidos, visto que parâmetros não listados podem ser encontrados.

4.3 DECODIFICAÇÃO E APRESENTAÇÃO

Conforme os dados são enviados, o software os decodifica e apresenta de maneira legível e em gráficos, permitindo o usuário acompanhar remotamente a telemetria realizada.

Todos os parâmetros são decodificados de acordo com a normas SAE J1939-71 e SAE J1708/J1587. Na Tabela 5 são apresentados os parâmetros desejados do projeto e suas especificações para decodificação.

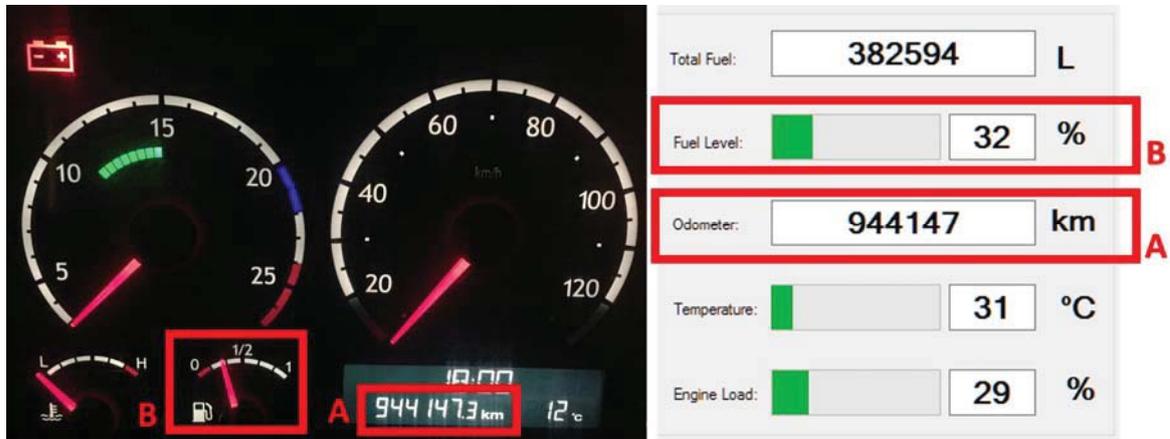
Tabela 5. Especificação de decodificação dos parâmetros

Parâmetro	J1939		J1708	
	Resolução do bit	Offset	Resolução do bit	Offset
Vel. Tacógrafo	0.00390625 km/h	0	0.805 km/h	0
Vel. Rodas	0.00390625 km/h	0	0.805 km/h	0
Vel. Motor	0.125 rpm	0	0.25 rpm	0
Odômetro	5 m	0	0.161 km	0
Temp. Liq. Refrig. Motor	1 °C	-40 °C	1 °F	0
Posição pedal acelerador	0.4 %	0	0.4 %	0
Cons. Combustível	0.5 L	0	0.473 L	0
Nível combustível	0.4 %	0	0.5 %	0
Carga do motor	1 %	0	0.5 %	0

Fonte: Próprio autor

A precisão da telemetria foi aferida comparando os dados apresentados no software com os valores visualizados no painel dos veículos, conforme ilustrado na Figura 42.

Figura 42. Aferição da telemetria: A - Odômetro; B - Nível de combustível



Fonte: Próprio autor

5 CONSIDERAÇÕES FINAIS

A partir dos objetivos definidos, o desenvolvimento do protótipo satisfaz as especificações do projeto, como a leitura dos barramentos de dados veiculares CAN J1939 e J1708, conexão e transmissão dos dados para o software servidor e a decodificação realizada por este último. Em paralelo com o desenvolvimento do dispositivo, o software permitiu realizar a interface entre o protótipo e o usuário, proporcionando uma visualização de status e da telemetria, sendo ela com os dados codificados ou decodificados.

A possibilidade de testar o dispositivo em veículos proporcionou, além da validação da decodificação realizada pelo software, a importância das normas SAE J1939 e SAE 1708, pois ambas são de suma importância no processo, e sem elas a decodificação não poderia ser facilmente realizada. Ademais, com base nos dados de telemetria, outros parâmetros podem ser decodificados, pois, embora não constem na norma Interface FMS, utilizando as normas da SAE, é possível decodificá-los. Sendo assim, para um trabalho futuro fica a possibilidade de aprimorar o software para decodificar esses outros parâmetros.

Foi constatado que o barramento CAN J1939 está presente em todos os veículos pesados testados, e acompanhando outros testes junto a empresa Cielo Telecom, muitos outros veículos possuem esse barramento. Já o barramento J1708 é uma característica exclusiva dos modelos da marca Volvo, sendo que aos poucos está caindo em desuso, principalmente nos veículos lançados a partir do ano de 2017, dando espaço a um novo barramento, esse que com uma breve análise, foi concluído que opera sobre o protocolo CAN, embora, diferente do padrão J1939, transmite as informações a 500 kbps. Além do mais, não foram encontradas bibliografias sobre esse protocolo, impossibilitando a inclusão nesse projeto.

Visando expandir as possibilidades do dispositivo, é sugerido a compatibilidade com outros barramentos de dados: como CAN2.0A em variadas taxas de transmissão e sistema OBDII, padrão universal para diagnóstico de veículos leves. No que tange ao software, a possibilidade de uma decodificação semiautomática, a partir de gráficos e testes em tempo real, facilitaria a interpretação de parâmetros ausentes nas normas.

REFERÊNCIAS

AGTECHNOLOGIES. **AGM 1604A-204 Specification**. [S.l.]. 2015.

AXIOMATIC. What is SAE J1939? **Axiomatic Global Electronic Solutions**, 2006. Disponível em: <<http://www.axiomatic.com/whatissaej1939.pdf>>. Acesso em: 27 de Agosto de 2016.

BALDISSERA, E. N. **Decodificador de dados usando o protocolo CAN - Padrão SAE J1939**. Universidade de Passo Fundo. Passo Fundo. 2011.

BRISA. **Arquiteturas de Redes de Computadores - OSI e TCP/IP**. São Paulo: Makron Books, 1994.

CANBUSKIT. What is CAN BUS?, 2016. Disponível em: <<http://canbuskit.com/what.php>>. Acesso em: 27 de Agosto de 2016.

CAVALCANTI, J. C. A INTERNET, o modelo nacional e uma proposta de enfoque para uma política de tarifas em sua operação no país. **Revista de Economia Política**, v. 17, n. 2 (66), p. 130-144, Abril - Junho 1997.

CHAVES, M. H. P. C. **Análise de estado de tráfego de redes TCP/IP para aplicação em detecção de intrusão**. INPE. São José dos Campos. 2003.

COMER, D. E. **Internetworking with TCP/IP Principles, Prot.** 4^a. ed. Upper Saddle River, New Jersey, USA: Prentice Hall, v. 1, 2000.

ESPRESSIF SYSTEMS. **Espressif Smart Connectivity Platform: ESP8266**. [S.l.]. 2013.

FARSI, M.; RATCLIFF, K.; BARBOSA, M. An overview of controller area networking. **Computing & Control Engineering Journal**, v. 10, n. 3, p. 113-120, 1999.

FMS-STANDARD. Information about the FMS-Standard. **FMS-Standard**, 2016. Disponível em: <<http://www.fms-standard.com/Truck/index.htm>>. Acesso em: 29 de Agosto de 2016.

GALLO, M. A.; HANCOCK, W. M. **Computer communications and networking technologies**. [S.l.]: Course Technology Press, 2005.

GUIMARÃES, A. D. A. **Eletrônica Automotiva Embarcada**. São Paulo: Érica, 2007.

GUIMARÃES, A. D. A.; SARAIVA, A. M. **Um roteiro de implementação de uma rede CAN (Controller Area Network)**. Conferência Internacional de Engenharia Automotiva - SIMEA. [S.l.]: [s.n.]. 2003.

JULICHER, J. **AN1230 - SAE J1708/J1587 Communications with the EUSART**. Microchip Technology Inc. [S.l.]. 2008.

JUNGER, M. **Introduction to J1939 - Application Note**. Vector Infomatik GmbH. Stuttgart, Alemanha. 2010.

KVASER. Introduction to SAE J1587. **Kvaser**, 2016. Disponível em: <<https://www.kvaser.com/about-can/can-standards/j1587/>>. Acesso em: 25 de Setembro de 2016.

KVASER. Introduction to SAE J1708. **Kvaser**, 2016. Disponível em: <<https://www.kvaser.com/about-can/can-standards/j1708/>>. Acesso em: 20 de Setembro de 2016.

LOPES, C. A. C. **CAN - Controller Area Network**. Universidade Estadual de Londrina. Londrina. 2009.

LOZANO-NIETO, A. Telemetry. In: WEBSTER, J. G. **The Measurement, Instrumentation, and Sensors Handbook**. [S.l.]: CRC Press, 1999. Cap. 87.

LUGLI, A. B.; SANTOS, M. M. D. **Sistemas Fieldbus para automação industrial: DeviceNet, CANOpen, SDS e Ethernet**. 1ª. ed. São Paulo: Érica, 2009.

MANNISTO, D.; DAWSON, M. **An Overview of Controller Area Network (CAN) Technology**. Raport instytutowy, Machine Bus Corporation. [S.l.]. 2003.

MARINHO, A. T.; CORDEIRO, W. S. C. **Desenvolvimento de uma Box Truck para realização de testes elétricos em componentes automotivos**. Universidade Tecnológica Federal do Paraná. Curitiba. 2014.

MAXIM INTEGRATED. Transceivers, 2016. Disponível em: <<https://www.maximintegrated.com/en/products/interface/transceivers.html>>. Acesso em: 12 de Setembro de 2016.

MAYO-WELLS, W. J. The Origins of Space Telemetry. **Technology and Culture**, v. 4, n. 4, p. 499-514, 1963.

MICROSOFT. Definição das sete camadas do modelo OSI e explicação de suas funções, 2013. Disponível em: <<https://support.microsoft.com/pt-br/kb/103884>>. Acesso em: 12 de Outubro de 2016.

NAKANO, K. **SD Standards and SD Technology**. CeBIT. India: [s.n.]. 2014.

PAZUL, K. **AN713 - Controller Area Network (CAN) Basics**. Microchip Technology Inc. [S.l.]. 1999.

POSTEL, J. **Transmission Control Protocol. Darpa Internet Program. Protocol Specification**. University of Southern California. Marina del Rey, California, USA. 1981.

SAASTAMOINEN, R. **Analysis of the SAE J1708 protocol**. Mälardalen University. Västerås, Sweden. 2008.

SIMMA SOFTWARE. J1939 Introduction. **Simma Software**, 2016. Disponível em: <<http://www.simmasoftware.com/j1939.html>>. Acesso em: 7 de Agosto de 2016.

SOCIETY OF AUTOMOTIVE ENGINEERS. **Electronic Data Interchange Between Microcomputer Systems in Heavy-Duty Vehicle Applications**. [S.l.]. 2002.

SOCIETY OF AUTOMOTIVE ENGINEERS. **Vehicle Application Layer - J1939-71**. [S.l.]. 2003.

STEVENS, W. R. **TCP/IP Illustrated**. 2ª. ed. [S.l.]: Addison-Wesley Professional, v. 1, 2011.

TEIXEIRA, F. C. R.; TORMIER, D. R. Utilização de Telemetria para Diagnóstico Automotivo à Distância. **SIMEA**, v. 2, n. 1, Setembro de 2015.

TANABE, R. New World Encyclopedia. **New World Encyclopedia**, 2015. Disponível em: <<http://www.newworldencyclopedia.org/p/index.php?title=Telemetry&oldid=992066>>.

Acesso em: 15 de Agosto de 2016.

TANENBAUM, A. S. **Redes de Computadores**. 4ª. ed. São Paulo: Editora Campus, 2003.

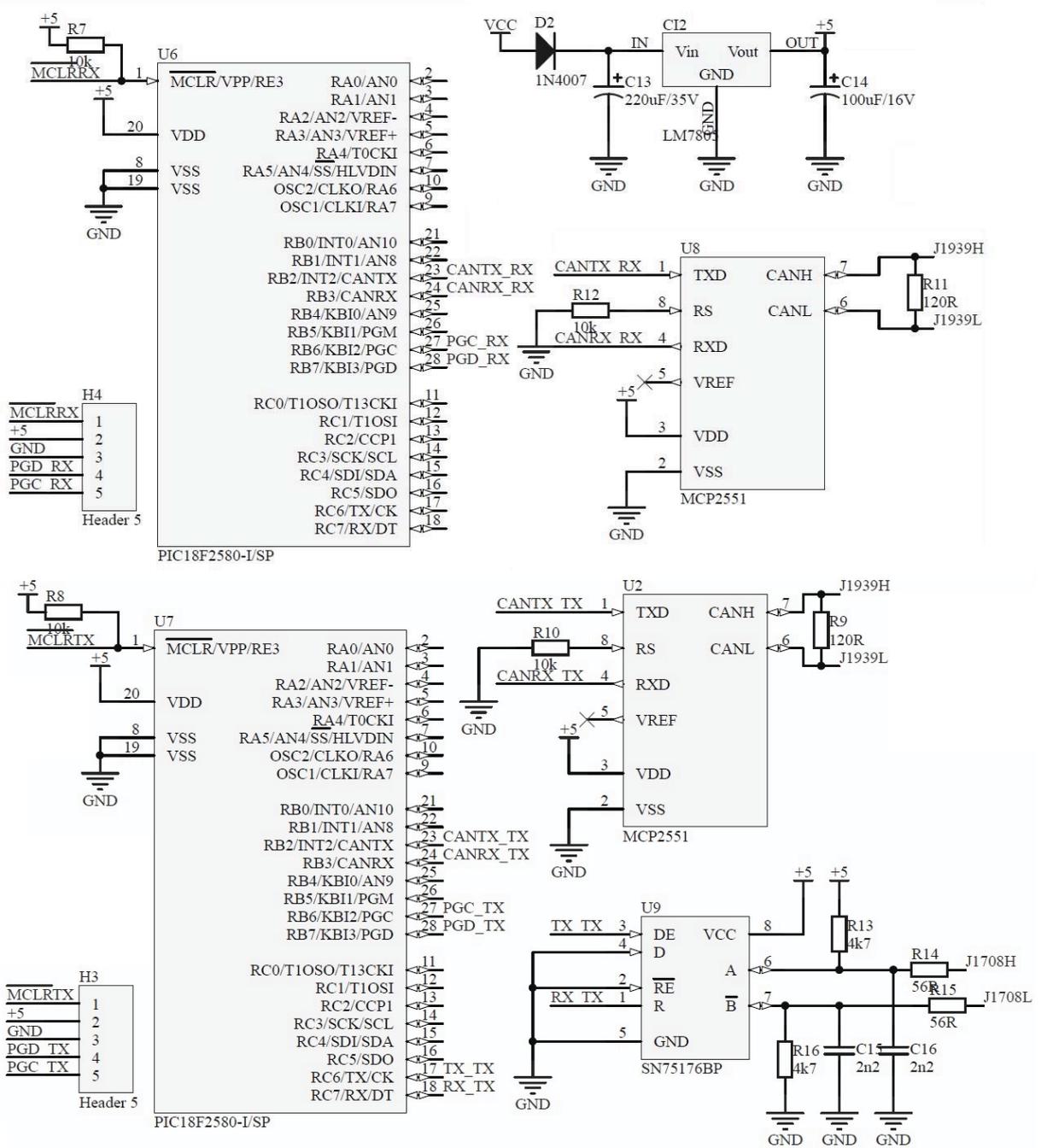
TEIXEIRA, F. C. R.; DE OLIVEIRA, M. C.; HELLENO, A. L. Telemetria Automotiva via Internet Móvel. **Revista Ciência e Tecnologia**, v. 16, n. 28/29, 2014.

TEXAS INSTRUMENTS. **LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator**. [S.l.]. 2016.

VARGAS, M. T. **Computador de Bordo Automotivo**. Universidade de Passo Fundo. Passo Fundo. 2007.

VOOS, W. **A Comprehensible Guide to J1939**. 1ª. ed. [S.l.]: Copperhill Media Corporation, 2008.

APÊNDICE C – ESQUEMÁTICO ELÉTRICO DO SIMULADOR J1939 E J1708



ANEXO A – PARÂMETROS DE DECODIFICAÇÃO DOS DADOS J1939

spn84 - Wheel-Based Vehicle Speed - Speed of the vehicle as calculated from wheel or tailshaft speed.

Data Length: 2 bytes
 Resolution: 1/256 km/h per bit , 0 offset
 Data Range: 0 to 250.996 km/h
 Type: Measured
 Suspect Parameter Number: 84
 Parameter Group Number: [65265]

spn91 - Accelerator Pedal Position 1 - The ratio of actual position of the analog engine speed/torque request input device

Data Length: 1 byte
 Resolution: 0.4 %/bit , 0 offset
 Data Range: 0 to 100 %
 Type: Measured
 Suspect Parameter Number: 91
 Parameter Group Number: [61443]

spn92 - Percent Load At Current Speed - The ratio of actual engine percent torque (indicated) to maximum indicated torque available at the current engine speed, clipped to zero torque during engine braking.

Data Length: 1 byte
 Resolution: 1 %/bit , 0 offset
 Data Range: 0 to 250 %
 Operating Range: 0 to 125%
 Type: Status
 Suspect Parameter Number: 92
 Parameter Group Number: [61443]

spn96 - Fuel Level - Ratio of volume of fuel to the total volume of fuel storage container.

Data Length: 1 byte
 Resolution: 0.4 %/bit , 0 offset
 Data Range: 0 to 100 %
 Type: Measured
 Suspect Parameter Number: 96
 Parameter Group Number: [65276]

spn110 - Engine Coolant Temperature - Temperature of liquid found in engine cooling system.

Data Length: 1 byte
 Resolution: 1 deg C/bit , -40 deg C offset
 Data Range: -40 to 210 deg C
 Type: Measured
 Suspect Parameter Number: 110
 Parameter Group Number: [65262]

spn190 - Engine Speed - Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.

Data Length: 2 bytes
 Resolution: 0.125 rpm/bit , 0 offset
 Data Range: 0 to 8,031.875 rpm
 Type: Measured
 Suspect Parameter Number: 190
 Parameter Group Number: [61444]

spn250 - Total Fuel Used - Accumulated amount of fuel used during vehicle operation.

Data Length: 4 bytes
 Resolution: 0.5 L/bit , 0 offset
 Data Range: 0 to 2,105,540,607.5 L
 Type: Measured
 Suspect Parameter Number: 250
 Parameter Group Number: [65257]

spn917 - High Resolution Total Vehicle Distance - Accumulated distance traveled by the vehicle during its operation.

NOTE - See SPN 245 for alternate resolution.

Data Length: 4 bytes
 Resolution: 5 m/bit , 0 offset
 Data Range: 0 to 21,055,406 km
 Type: Measured
 Suspect Parameter Number: 917
 Parameter Group Number: [65217]

spn1624 - Tachograph vehicle speed - Speed of the vehicle registered by the tachograph.

Data Length: 2 bytes
 Resolution: 1/256 km/h per bit , 0 offset
 Data Range: 0 to 250.996 km/h
 Type: Measured
 Suspect Parameter Number: 1624
 Parameter Group Number: [65132]

ANEXO B – PARÂMETROS DE DECODIFICAÇÃO DOS DADOS J1708

A.84 Road Speed—Indicated vehicle velocity.

Parameter Data Length: 1 Character
 Data Type: Unsigned Short Integer
 Bit Resolution: 0.805 km/h (0.5 mph)
 Maximum Range: 0.0 to 205.2 km/h (0.0 to 127.5 mph)
 Transmission Update Period: 0.1 s
 Message Priority: 1
 Format:
 PID Data
 84 a
 a— Road speed

A.190 Engine Speed—Rotational velocity of crankshaft.

Parameter Data Length: 2 Characters
 Data Type: Unsigned Integer
 Bit Resolution: 0.25 rpm
 Maximum Range: 0.0 to 16383.75 rpm
 Transmission Update Period: 0.1 s
 Message Priority: 1
 Format:
 PID Data
 190 a a
 a a— Engine speed

A.96 Fuel Level—Ratio of volume of fuel to the total volume of the primary fuel storage container.

Parameter Data Length: 1 Character
 Data Type: Unsigned Short Integer
 Bit Resolution: 0.5%
 Maximum Range: 0.0 to 127.5%
 Transmission Update Period: 10.0 s
 Message Priority: 6
 Format:
 PID Data
 96 a
 a— Fuel level

A.91 Percent Accelerator Pedal Position—Ratio of actual accelerator pedal position to maximum pedal position.

Parameter Data Length: 1 Character
 Data Type: Unsigned Short Integer
 Bit Resolution: 0.4%
 Maximum Range: 0.0 to 102.0%
 Transmission Update Period: 0.1 S
 Message Priority: 3
 Format:
 PID Data
 91 a
 a— Percent accelerator pedal position

A.110 Engine Coolant Temperature—The temperature of liquid found in engine cooling system.

Parameter Data Length: 1 Character
 Data Type: Unsigned Short Integer
 Bit Resolution: 1.0 °F
 Maximum Range: 0.0 to 255.0 °F
 Transmission Update Period: 1.0 s
 Message Priority: 4
 Format:
 PID Data
 110 a
 a— Engine coolant temperature

A.245 Total Vehicle Distance—Accumulated distance travelled by vehicle during its operation.

Parameter Data Length: 4 Characters

Data Type: Unsigned Long Integer

Bit Resolution: 0.161 km (0.1 mi)

Maximum Range: 0.0 to 691 207 984.6 km (0.0 to 429 496 729.5 mi)

Transmission Update Period: 10.0 s

Message Priority: 7

Format:

PID Data

245 n a a a a

n— Number of parameter data characters = 4

a a a a— Total vehicle distance

A.92 Percent Engine Load—Ratio of current output torque to maximum torque available at the current engine speed.

Parameter Data Length: 1 Character

Data Type: Unsigned Short Integer

Bit Resolution: 0.5%

Maximum Range: 0.0 to 127.5%

Transmission Update Period: 0.1 s

Message Priority: 3

Format:

PID Data

92 a

a— Percent engine load

A.250 Total Fuel Used—Accumulated amount of fuel used during vehicle operation.

Parameter Data Length: 4 Characters

Data Type: Unsigned Long Integer

Bit Resolution: 0.473 L (0.125 gal)

Maximum Range: 0.0 to 2 032 277 476 L (0.0 to 536 870 911.9 gal)

Transmission Update Period: On request

Message Priority: 8

Format:

PID Data

250 n a a a a

n— Number of parameter data characters = 4

a a a a— Total fuel used