Ministério
da Educação

Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia de
Santa Catarina

CÂMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA

BACHARELADO EM ENGENHARIA MECATRÔNICA

LARISSA HINCKEL CAVALCANTE

SISTEMA DE MONITORAMENTO AUTOMOTIVO VIA REDE CAN

FLORIANÓPOLIS, JULHO DE 2018

INSTITUTO FEDERAL
SANTA CATARINA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CÂMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA

BACHARELADO EM ENGENHARIA MECATRÔNICA

LARISSA HINCKEL CAVALCANTE

SISTEMA DE MONITORAMENTO AUTOMOTIVO VIA REDE CAN

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para a obtenção do título de Bacharel em Engenheira Mecatrônica

Professor Orientador: Adriano Regis

FLORIANÓPOLIS, JULHO DE 2018

Ficha de identificação da obra elaborada pelo autor.

Cavalcante, Larissa H.
Sistema de Monitoramento Automotivo via Rede CAN /
Larissa H. Cavalcante ; orientação de Adriano Regis. -
Florianópolis, SC, 2018. 70 p.

Trabalho de Conclusão de Curso (TCC) - Instituto
Federal de Santa Catarina, Câmpus Florianópolis.
Bacharelado em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.

1. Rede CAN. 2. OBD-II. 3. Monitoramento Automotivo.
4. Comunicação Remota. I. Regis, Adriano. II. Instituto
Federal de Santa Catarina. Departamento Acadêmico
de Metal Mecânica. III. Título.

SISTEMA DE MONITORAMENTO AUTOMOTIVO VIA REDE CAN

LARISSA HINCKEL CAVALCANTE

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Engenharia Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Bacharelado em Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 04 de julho de 2018.

Banca examinadora:

_____

Prof. Adriano Regis, Me.

Orientador

Instituto Federal de Santa Catarina

_____

Prof. Jean Paulo Rodrigues, Dr.

Instituto Federal de Santa Catarina

_____

Prof. Marcelo Vandresen, Dr.

Instituto Federal de Santa Catarina

INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante **LARISSA HINCKEL CAVALCANTE**, matrícula n° **161007347-9**, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado **SISTEMA DE MONITORAMENTO AUTOMOTIVO VIA REDE CAN**, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 04 de Julho de 2018.

Prof. Orientador do TCC: Adriano Regis

Dedico esse trabalho a todos aqueles que,

com sua paciência e afeto,

me acompanharam nessa jornada.

AGRADECIMENTOS

Essa é a conclusão de mais uma jornada de aprendizado, que não será a última, mas que talvez seja a final dentro dessa instituição de ensino que tanto me engrandeceu. Durante a árdua trajetória de desenvolvimento desse Trabalho de Conclusão de Curso, muitas foram as pessoas que contribuíram para sua realização. Porém, não seria justo considerar apenas 5 meses de trabalho, quando foram os muitos anos de apoio recebido que me permitiram chegar até aqui.

Primeiramente, gostaria de agradecer em especial à minha família, que sempre me deu todo o apoio e as condições necessárias para realizar meus sonhos. João, Eliane, Lélia e Laisa, um obrigado não seria suficiente para reconhecer a importância de vocês na minha vida.

Em segundo lugar, quero fazer um agradecimento aos mestres que passaram pelo meu caminho e compartilharam parte do seu conhecimento com tanta bondade. Obrigado, principalmente, aos professores do IFSC, que caminharam comigo nessa longa jornada.

Não posso deixar de citar os nomes das pessoas que se fizeram bastante presentes durante esses 5 meses de desenvolvimento do meu TCC:

Bruna, obrigada pelo seu apoio e carinho incondicional que me deram força para continuar até o final.

Felippe, querido amigo que esteve sempre disponível e disposto a ajudar, sem suas contribuições esse trabalho não seria o mesmo.

Vinicius e Lucas, colegas que plantaram a ideia inicial desse projeto e foram sempre solícitos a oferecer assistência.

Por último, mas não menos importante, obrigado Adriano pela orientação clara, concisa e sábia, durante esse período de tensão.

Com a felicidade de finalizar esse projeto, fica meu agradecimento de coração a todos vocês.

Aprender é a única coisa de que a mente nunca se cansa,

nunca tem medo e nunca se arrepende.

(Leonardo da Vinci)

RESUMO

Em virtude da carência de produtos acessíveis para a supervisão de veículos no Brasil e visando a implementação de ferramentas que aumentem a segurança dos motoristas, foi proposto o desenvolvimento de um sistema de monitoramento automotivo que possibilite ao usuário o acesso às informações sobre o funcionamento do seu automóvel. Através do estudo da rede CAN, decidiu-se a melhor abordagem para a obtenção das informações via porta OBD-II do carro, construindo-se um hardware embarcado para aquisição de dados do veículo em tempo real e acesso à localização geográfica do mesmo. Uma comunicação remota foi implementada com a intenção de enviar as informações para um aplicativo, também desenvolvido nesse projeto, de forma a facilitar a visualização dos resultados pelo usuário. O protótipo foi testado em dois veículos de modelos diferentes, com a intenção de validar o sistema, e obteve resultados satisfatórios, atendendo à todas as funcionalidades propostas. Os valores adquiridos pelo hardware, e transmitidos via *bluetooth* ao celular do usuário, corresponderam aos mostrados no painel de cada veículo testado. O sistema de monitoramento facilita a detecção de falhas no automóvel, possibilitando ao motorista realizar a manutenção do carro e prevenir acidentes graves, e também garante o acesso às variáveis de estado dos módulos veiculares, permitindo assim, por exemplo, a aquisição de dados referentes a um determinado trajeto. Dessa forma, é possível mensurar a variação de dados como velocidade e consumo de combustível durante um tempo desejado, monitorando o desempenho do motorista e o rendimento do veículo.

Palavras-Chave: Rede CAN. OBD-II. Monitoramento Automotivo. Comunicação Remota.

ABSTRACT

In order to address the lack of accessible products to gauge vehicles' systems in Brazil and to implement tools to increase driver safety, this work proposes the development of an automotive monitoring system that allows the user to access their vehicle's operation information. The scope of the Project was determined through the study of the Controller Area Network (CAN) protocol. The objective was to obtain a vehicle's operational information using the OBD-II port of the car, then an embedded hardware was developed to acquire the vehicle's data in real time and access its geographical information. A remote communication was also implemented in the prototype with the intention of sending the vehicle's data to a mobile application, an application that was also developed in this project, in order for the user to visualize the data. The prototype was tested in two vehicles of different models so as to validate the system and obtain satisfactory results while taking into account all the proposed functionalities. The values acquired by the hardware, transmitted via bluetooth to the user's cell phone, corresponded to those shown in the panel of each tested vehicle. This monitoring system facilitates the detection of troubles in the car, enabling the driver to perform car maintenance and prevent serious accidents. It also makes it possible to access the state variables of the vehicle modules, thus allowing, for example, the acquisition of specific route information. In this way, it is possible to measure the variation of data such as speed and fuel consumption during a desired time, making it possible for the driver to monitor their performance as well the performance of the vehicle.

Keywords: Controller Area Network. OBD-II. Automotive Monitoring System. Remote Communication.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK – Acknowledge

ACU – Airbag Control Unit

ATMEGA328p – microcontrolador fabricado pela empresa ATMEL

BCM – Brake Control Module

BCM – Body Control Module

CAN – Controller Area Network

CARB – California Air Resources Board

CCM – Central Control Module

CRC – Cyclic Redundancy Check

CSMA/CD – Carrier Sense Multiple Access/Collision Detection

CTM – Central Timing Module

DENATRAN – Departamento Nacional de Trânsito

DLC – Data Lenght Code

DTC – Diagnostic Trouble Code

ECM – Engine Control Module

ECU – Engine Control Unit

EOF – End of Frame

EPROM – Erasable Programmable Ready Only Memory

EUA – Estados Unidos da América

GEM – General Electronic Module

GND – Ground

GPS – Global Position System

IDE – Identifier Extension

IFS – Inter-Frame Space

ISO – International Standard Organization

MISO – Master Input/ Slave Output

MOSI – Master Output/Slave Input

NDA – Non-Destructive Arbitration

OBD – On-Board Diagnostic

PCM – Powertrain Control Module

PID – Parameter Identifier

RAM – Random Access Memory

RM-OSI – *Referencial Model for Open System Interconnection*

RTR – Remote Transmission Request

SAE – Sociedade de Engenheiros de Automóveis

SCK – Serial Clock

SCM – Suspension Control Module

SOF – Start-of-Frame

SPI – Serial Peripheral Interface

TCM – Transmission Control Module

VIN – Vehicle Identification Number

SUMÁRIO

1 INTRODUÇÃO

Desde o surgimento do automóvel, várias tecnologias foram empregadas para melhorar o desempenho e acrescentar funcionalidades a esse meio de transporte. Acompanhando o desenvolvimento da ciência e os recursos tecnológicos cada vez mais acessíveis, o que antes baseava-se numa simples máquina de combustão interna, se tornou um sistema robusto que integra a mecânica automotiva com uma eletrônica sofisticada para o controle do veículo.

O crescimento da produção em massa, em função das linhas de montagem, e a popularização dos automóveis, devido à redução nos preços de mercado, consolidou esse veículo como um dos meios de locomoção mais utilizados no mundo. No Brasil, segundo fontes do Departamento Nacional de Trânsito (DENATRAN), circulam aproximadamente 53 milhões de automóveis (DEPARTAMENTO NACIONAL DE TRÂNSITO, 2018), fazendo-se então necessário o aprimoramento constante de tecnologias que permitam ao usuário um maior gerenciamento do seu veículo.

Acompanhando o desenvolvimento de novas tecnologias e na tentativa de atender um consumidor que busca cada vez mais modernidade, conforto e segurança, os carros passaram a contar com sistemas computacionais gerenciando os módulos veiculares. Sistemas eletrônicos conseguem identificar simultaneamente o estado do motor, dos sistemas de iluminação e combustível, por exemplo. Sistemas de localização, como o GPS (*Global Positioning System*), também fazem parte das tecnologias atualmente presentes nos automóveis do mundo todo.

O avanço do uso de sistemas eletrônicos e computacionais busca trazer confiabilidade aos veículos e aumento da segurança e do conforto dos usuários. Na década de 1970, o desenvolvimento de tecnologias era direcionado aos sistemas mecânicos do carro, sendo o custo com eletrônicos apenas de 5% do total do automóvel (CHARETTE, 2009). Em 2005, os gastos com a eletrônica do veículo representaram 15% do valor do produto, porém, os investimentos nesse setor só tendem a crescer, estimando-se que o custo com esse tipo de tecnologia atingirá 50% do valor do automóvel.

O chamado computador de bordo é o aparelho que melhor exemplifica o sistema de monitoramento integrado a grande parte dos projetos automotivos nos dias de hoje. Apesar de ser um dispositivo já comercializado nos EUA na década de 1980 (O COMPUTADOR, 2013), no Brasil, vem sendo desenvolvido desde a década de 1990. Os primeiros computadores surgiram após a inserção de uma das principais tecnologias presentes nos automóveis atuais, a injeção eletrônica, que permite o monitoramento da quantidade de combustível consumida.

Um dos sistemas precursores patenteado no país foi uma arquitetura para o monitoramento de informações em locomotivas (PINTO; FILHO; AMARAL, 1992). O computador armazenava 10 dados operacionais, além de eventos verificados durantes as viagens, que poderiam ser acessados posteriormente para a análise do funcionamento da locomotiva. O sistema também contava com um display alfanumérico para exibir as informações ao usuário, tais como consumo de energia e a potência elétrica do gerador principal da locomotiva.

Os computadores de bordo dos dias de hoje contam com um sistema de monitoramento eficiente, rápido e preciso, além de oferecerem interfaces amigáveis aos usuários. Apesar de contarem com diversas facilidades que priorizam o conforto, como o gerenciamento de temperatura do ambiente e sistema de som, uma das funcionalidades mais importante desse dispositivo é a detecção de falhas. A sinalização de possíveis problemas nos módulos veiculares é essencial para mantenabilidade do veículo e para a segurança do condutor.

Estima-se que o brasileiro gaste 40% do valor do automóvel com manutenção e despesas por ano (PINTO LIMA, 2013). Caso não seja realizada a manutenção devida, o funcionamento do carro fica comprometido e os gastos futuros serão maiores. A maior parte dos usuários é leiga e não tem conhecimento da mecânica do seu veículo, nem dos prazos para troca de óleo e pneus, por exemplo. O mau funcionamento de veículos é responsável por muitos acidentes de trânsito, comprometendo a vida do motorista e de terceiros.

Entretanto, computadores de bordo ainda não estão efetivamente presentes na maior parte dos veículos que circulam no Brasil. Esse dispositivo é um opcional encontrado em carros mais novos e com valor mais elevado no mercado, os quais representam uma pequena parcela da frota nacional. Em razão dessa carência,

foram desenvolvidos alguns dispositivos capazes de realizar a função dos computadores de bordo, fazendo o monitoramento dos módulos do veículo e apontando possíveis falhas no sistema.

Esses dispositivos funcionam ao serem conectados a uma porta de acesso, presente em todos os carros produzidos no Brasil a partir de 2010 (TELEMETRIA, 2016). Tal entrada permite a comunicação com o sistema chamado de *On-Board Diagnostic*, que em inglês significa diagnóstico de bordo, e possibilita a leitura e transmissão de dados do veículo. O sistema de diagnóstico vigente de acordo com a legislação atual é o OBD-II, segunda versão do padrão originalmente estabelecido.

Para este trabalho, busca-se desenvolver um dispositivo que, ao ser conectado à porta OBD-II, consiga acessar a rede CAN do automóvel e interpretar os dados referentes às variáveis de estado dos módulos veiculares. Através de um sistema embarcado que traduz esse protocolo e processa os dados, o protótipo disponibilizará informações sobre o carro, bem como sua localização. As informações serão enviadas remotamente ao usuário e poderão ser acessadas através de um aplicativo de celular.

1.1 TEMA

Desenvolvimento de um sistema de monitoramento veicular, que, através de hardware embarcado, lê e processa os dados obtidos pela porta OBD-II provenientes da rede CAN e envia as informações para um aplicativo de celular via comunicação remota.

1.2 DELIMITAÇÃO DO TEMA

O protótipo confeccionado será limitado ao uso em carros cujo protocolo seja o CAN 11kbps, atendendo principalmente às normas SAE J1979 (2006), ISO 11898 (2003) e ISO 15031 (2004), que regulamentam os serviços e parâmetros para efetivação do sistema de diagnóstico automotivo.

1.3 DEFINIÇÃO DO PROBLEMA

Apesar dos carros possuírem o sistema de diagnóstico, os dados não podem ser acessados naqueles veículos nos quais não há computador de bordo. Devido ao preço elevado e a escassez de produtos com interface amigável que realizam o monitoramento veicular, foi proposto o desenvolvimento um sistema de monitoramento com comunicação remota que permita ao usuário obter informações sobre as variáveis de estado do seu automóvel, bem como possíveis falhas nos módulos veiculares.

1.4 JUSTIFICATIVA

Existem poucos dispositivos no mercado atualmente que fornecem informações necessárias ao usuário sobre o funcionamento e manutenção do seu veículo. A busca pela segurança e pela redução de gastos estão entre os fatores mais importantes para aqueles que adquirem automóveis. O desenvolvimento de um sistema que permita ao usuário tomar conhecimento sobre dados técnicos do funcionamento do seu veículo, além de indicar a manutenção preventiva do mesmo, traria segurança e redução de gastos.

Segundo pesquisa realizada pelo Instituto Scaringella Trânsito (CZERWONKA, 2016), 30% dos acidentes de trânsito tem causas relacionadas à falta de manutenção dos veículos. A maior parte dos motoristas desconhece o mecanismo do seu carro, limitando-se a checa-lo apenas quando problemas começam a aparecer. De acordo com o Observatório Nacional de Segurança Viária (CZERWONKA, 2016), o cuidado preventivo do veículo, como a checagem de peças e a calibração de pneus, reduz significativamente os custos do motorista.

Esse projeto busca atender a carência de produtos de supervisão veicular no mercado, na intenção de fornecer uma ferramenta confiável e de fácil uso para o usuário que não possui conhecimento prévio sobre o funcionamento e a manutenção preventiva do seu automóvel.

1.5 OBJETIVOS

1.1.1 Objetivo geral

O presente trabalho propõe como objetivo geral o desenvolvimento de um sistema de monitoramento automotivo, baseado no protocolo de rede CAN, que informa remotamente ao usuário sobre o funcionamento e localização do seu veículo.

1.1.2 Objetivos específicos

a) Estudar a Rede CAN, essencial para o desenvolvimento do projeto;
b) Realizar a aquisição de sinais provenientes da porta *On-Board Diagnostic* do carro;
c) Processar os sinais através de um módulo CAN;
d) Gerenciar os dados e os processos do sistema através do microcontrolador ATMEGA328p;
e) Identificar as informações relevantes sobre o funcionamento do automóvel;
f) Detectar possíveis falhas nos módulos veiculares;
g) Identificar a localização do carro através de um módulo GPS;
h) Enviar os dados remotamente via módulo Bluetooth;
i) Desenvolver um aplicativo Android que apresente as informações requisitadas.

2 REVISÃO DA LITERATURA

Para compreender o funcionamento da eletrônica presente nos automóveis e assim ter acesso às informações de estado dos módulos do sistema, é necessário o entendimento das tecnologias empregadas no sensoriamento do veículo. O projeto proposto visa desenvolver um sistema de monitoramento automotivo, através da leitura de informações de estado do veículo.

Os dados serão obtidos pela porta *On-Board Diagnostic* e provêm de sensores gerenciados pelo sistema computacional do veículo, a ECU (*Electronic Control Unit*). Todas essas informações são compartilhadas entre os módulos veiculares através de um sistema de comunicação serial chamado CAN (*Controller Area Network*). Para processar os dados recebidos será utilizado um microcontrolador ATMEGA328P, em conjunto com um módulo CAN, um circuito responsável pela tradução da linguagem de comunicação do carro para a linguagem Serial, compreendida pelo microcontrolador.

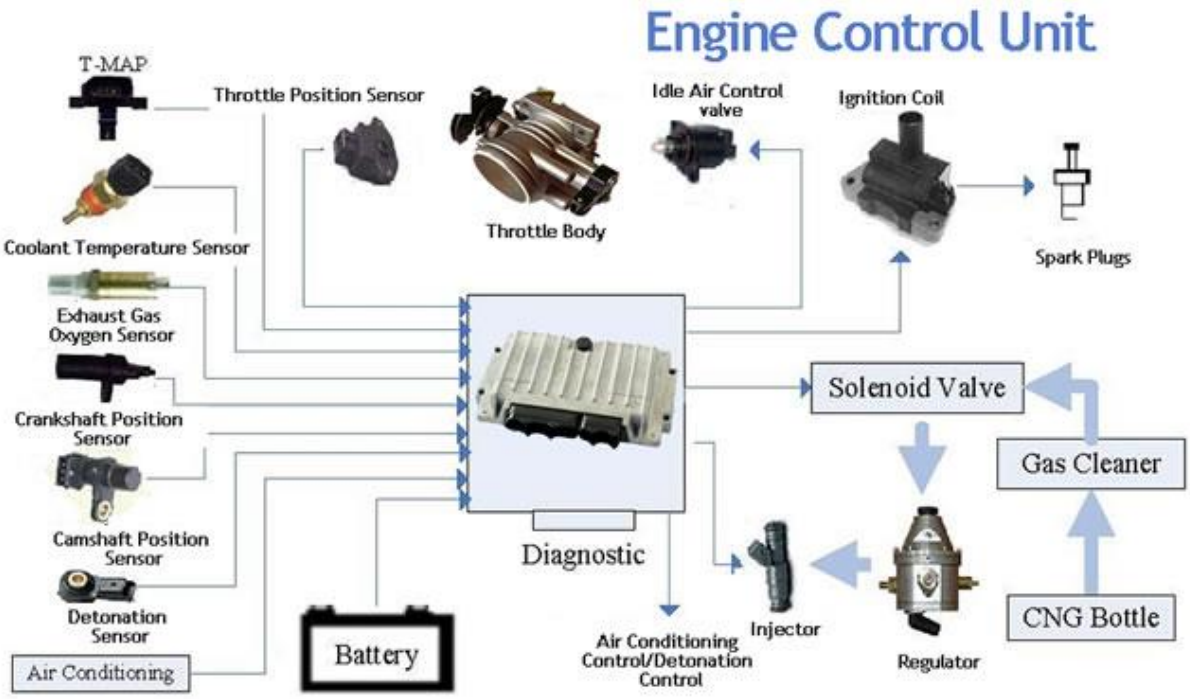
2.1 ECU

As ECUs são as centrais eletrônicas dos carros e são as unidades responsáveis pela comunicação dos módulos automotivos. A ECU é o cérebro do carro e é quem comanda o sistema de injeção eletrônica. Ela recebe os dados provenientes dos sensores presentes no carro, como sensores de pressão e velocidade, para assim comandar os atuadores e mantê-los funcionando com a maior eficiência possível.

A unidade de controle é constituída basicamente de um microprocessador que compara os dados recebidos com parâmetros presentes em uma memória EPROM (*Erasable Programmable Ready Only Memory*) (DIAS, 2012). De acordo com a leitura feita dos sensores, são enviados sinais de comando aos atuadores, adaptando o sistema de injeção a diferentes circunstâncias de funcionamento do veículo. Quando há alguma falha no sistema, A ECU grava o erro em forma de código, em uma memória RAM (*Random Access Memory*), que pode ser acessada posteriormente com um aparelho de diagnóstico automotivo.

Os sensores presentes nas diversas partições do automóvel, como mostrado na Figura 1, emitem sinais analógicos variando de 0 a 5 volts (DIAS, 2012), indicando o estado dos elementos verificados. A ECU, em contrapartida, interpreta unicamente sinais digitais, fazendo-se necessário um conversor analógico-digital para a tradução da informação.

Figura 1 – Central Eletrônica do Carro



Fonte: (ECU, 2018)

Apesar de fornecer dados referenciais, como mapas de injeção e curvas características, para o controle dos atuadores, a ECU nem sempre receberá os sinais de entrada previstos. Em caso de má funcionamento dos sensores, ou falha de algum elemento do sistema, a unidade de controle conta com uma medida de segurança, que permite que o veículo mantenha seu funcionamento. Esse modo emergencial aciona uma ação de recuperação do sistema, chamada Recovery, na qual a ECU prioriza um valor pré-estabelecido como parâmetro de leitura, ignorando os sinais provenientes do sensor defeituoso.

A ECU dos automóveis atuais é subdividida em módulos, podendo haver mais de 100 ECUs em um carro de alta performance (SCOTT BROWN, 2017). As unidades de controle usuais são (ONBOARD, 2018):

a) ECM (*Engine Control Module*): Módulo de Controle do Motor;
b) PCM (*Powertrain Control Module*): Módulo de Controle do Trem de Força;
c) TCM (*Transmission Control Module*): Módulo de Controle de Transmissão;
d) ACU (*Airbag Control Unit*): Módulo de Controle do *Airbag*;
e) BCM (*Brake Control Module*): Módulo de Controle do Freio;
f) CCM (*Central Control Module*): módulo de controle central;
g) CTM (*Central Timing Module*): módulo temporizador;
h) GEM (*General Electronic Module*): módulo eletrônico geral;
i) BCM (*Body Control Module*): modulo de controle do chassi;
j) SCM (*Suspension Control Module*): módulo de controle da suspensão;

O PCM seja talvez o mais importante dos módulos e geralmente inclui ambos ECM e TCM. A ECU do motor é chamada de ECM, apesar de algumas literaturas utilizarem o termo ECU (*Engine Control Unit*) também para designar esse módulo. A ECM é considerada o cérebro do sistema de gestão do motor e as variáveis mais importantes para o funcionamento do motor, como a mistura de combustível e o ponto de ignição, são gerenciadas por ele (O QUE, 2018).

Apesar da resistência e qualidade do sistema eletrônico da ECM, o módulo pode apresentar danos. Problemas como vibração e calor excessivo podem atrapalhar o funcionamento da unidade de controle. Ao menor sinal de distúrbio do sistema, códigos de diagnóstico (DTC – *Diagnostic Trouble Code*) podem ser obtidos através da porta OBD do carro.

2.2 *ON-BOARD DIAGNOSTIC*

Embora seja um meio de transporte amplamente utilizado, os automóveis são bastante suscetíveis a falhas. Inicialmente, sua estrutura baseava-se em uma mecânica simples, descrita por um movimento motor gerado através de um processo químico, a combustão. Com o passar dos anos e o desenvolvimento tecnológico,

vários artifícios foram empregados a fim de aumentar as funcionalidades, bem como a segurança do veículo, deixando o sistema mais complexo e aumentando as variáveis de controle.

Visando a detecção de possíveis falhas no veículo, o sistema de diagnóstico OBD (*On Board Diagnostics*) começou a ser desenvolvido na década de 1960, na Califórnia (TELEMETRIA, 2016). O objetivo inicial desse sistema era o controle das taxas de emissão de carbono, porém sua aplicação acabou se ampliando e abrangendo o sensoriamento de diferentes módulos veiculares. Por se tratar de uma questão ambiental a priori, foi a California Air Resources Board (CARB) que estabeleceu, em 1966, que esse sistema deveria ser obrigatório em todos os carros americanos produzidos a partir de então (MACHADO; OLIVEIRA, 2008).

O diagnóstico fundamenta-se na leitura dos dados provenientes da ECU, apontando qualquer mau funcionamento de algum componente do carro. Um computador utilizaria então a porta OBD para acessar e ler as informações de estado processadas pela unidade de controle do motor. Informações como rotação do motor e temperatura podem ser obtidas simultaneamente através de um computador de bordo, por exemplo.

A primeira versão desse sistema, hoje conhecida como OBD-I, não apresentava uma padronização nem em relação aos conectores nem aos protocolos usados. Cada empresa desenvolvia seu produto com seus critérios e normais próprias, o que dificultava o acesso às informações sobre avarias do veículo. Em sua maioria, os elementos pertencentes ao OBD I eram:

a) Sensor de oxigênio
b) Sistema de EGR
c) Sistema de combustível
d) Componentes elétricos
e) Sistemas Eletrônicos
f) Informação de Diagnóstico
g) Códigos de erros

Além da falta de uniformidade, o sistema OBD I se mostrou ineficiente na determinação do elemento defeituoso. Viu-se então a necessidade de se desenvolver um novo conjunto de normas para diagnóstico automotivo. Em 1988, a Sociedade de

Engenheiros de Automóveis (SAE) propôs a regulamentação do *On-Board Diagnostics*, visando padronizar os modelos existentes, culminando no surgimento do sistema atualmente em uso, o OBD II (MACHADO; OLIVEIRA, 2008).

2.2.1 OBD-II

No ano de1996, os Estados Unidos tornaram obrigatória a utilização do OBD II em todos os veículos produzidos em seu território, independente da montadora. A segunda geração do diagnóstico de bordo apresenta também uma lista maior de elementos controlados:

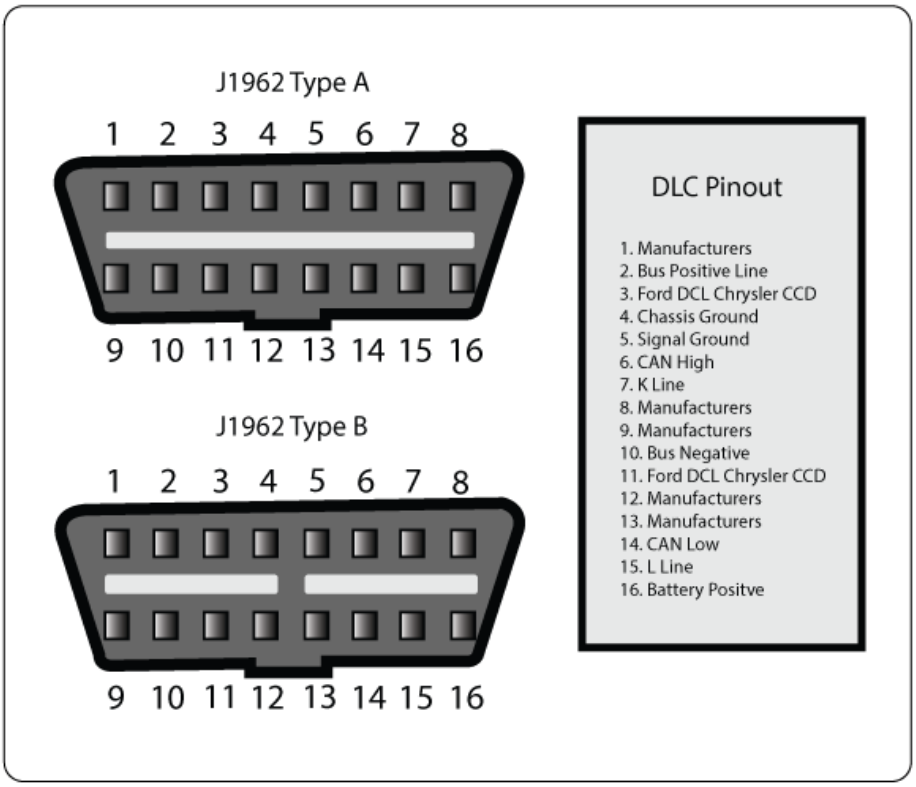
a) Sensor de oxigênio
b) Sistema de EGR
c) Sistema de combustível
d) Componentes elétricos
e) Sistemas Eletrônicos
f) Eficiência do catalisador
g) Aquecimento do catalisador
h) Combustão espontânea
i) Sistema de evaporação
j) Sistema de ar secundário
k) Informações do Diagnóstico
l) Códigos de falhas
m) Parâmetros do motor
n) Memorização de avarias
o) Padronização das ligações

O conector padrão, representado na Figura 2, definido para o novo sistema, apresenta 16 contatos. Entre eles, nove tem funções fixas bem definidas e o restante é determinado a critério do fabricante (LIN et al., 2009). O conector padrão para esse sistema é o J1962 e atende aos requisitos da norma ISO 15031-3 (2004).

1 Fabricante;
2 Barramento Positivo: Comunicação de dados de acordo com o protocolo SAE J1850;
3 Comunicação de dados (positivo) para diagnóstico;

4 Terra do chassi;
5 Terra;
6 Sinal CAN (alta), seguindo o protocolo ISO 12765-4;
7 Linha K, seguindo protocolos ISO 9141-2;
8 Fabricante;
9 Fabricante;
10 Barramento negativo: Comunicação de dados de acordo com o protocolo SAE J1850;
11 Comunicação de dados (negativo) para diagnóstico;
12 Fabricante;
13 Fabricante;
14 Sinal CAN (baixa), seguindo o protocolo ISO 12765-4;
15 Linha L, seguindo protocolos ISO 9141-2;
16 Alimentação de tensão positiva.

Figura 2 - Conector OBD-II



Fonte: (OBD II, 2018)

Através do conector, é possível acessar informações proveniente dos módulos veiculares. O sistema OBD-II estabelece 10 modos de operação, descritos pela norma SAE J1979 (2006), porém nem todos os modelos de veículo suportam todos os modos (OBD MODES, 2018).

2.2.1.1 Modo 01

O Modo 01 retorna leituras atuais dos módulos veiculares. Através dessa configuração, os dados provenientes de diversos sensores da ECU podem ser acessados em tempo real. A requisição das informações é feita através de identificadores de parâmetros, ou PIDs (*Parameter Identifiers*), que são códigos que representam os sensores do carro e têm formato hexadecimal. A tabela 1 apresenta uma lista de alguns PIDs suportados pelo Modo 01.

Tabela 1 – Lista de PIDs

PID	Descrição
05	Temperatura do Motor
0A	Pressão do combustível em kPa
0C	Rotação do motor em RPM
0D	Velocidade do carro em Km/h
0F	Temperatura do ar de admissão em Â°C
11	Posição do acelerador em %
2F	Nível de combustível em %
33	Pressão Barométrica em kPa
42	Tensão do módulo de controle em V
51	Tipo de combustível usado pelo veículo
52	% de etanol
5C	Temperatura do óleo do motor em Â°C
5E	Consumo de Combustível em l/h
67	Temperatura da água do motor em Â°C

Fonte: Autoria própria

2.2.1.2 Modo 02

Através do Modo 02, os dados do sensor requerido são registrados no momento em que a falha ocorre.

2.2.1.3 Modo 03

Fazendo uma requisição para o Modo 03, são mostrados os códigos de falha armazenados no sistema. Para representar as falhas nos elementos automotivos, o OBD II é representado por 5 códigos (CÓDIGOS, 2018). O primeiro código diz respeito ao tipo de sistema analisado e é representado por uma letra.

a) B – *Body System*: Carroceria;
b) C – *Chassis*: Chassi;
c) P – *Powertrain System*: Sistema de transmissão do motor;
d) U – *Network*: Comunicação entre módulos de controle.

Os códigos restantes são designados por dígitos numéricos. O primeiro identifica o tipo de código (0 para genérico e 1 para específico da montadora). O segundo refere-se ao sub-sistema onde a falha se encontra e os demais dígitos descrevem o mau funcionamento, como exemplificado na Figura 3.

Figura 3 – Estrutura do Código de Falha

Fonte: (CÓDIGOS, 2018)

2.2.1.4 Modo 04

O Modo 04 é utilizado para excluir os códigos de falha armazenados, fazendo assim com que a luz de indicação de problema no motor se apague.

2.2.1.5 Modo 05

Por meio do modo 05, são feitos testes de diagnóstico do sensor de oxigênio em veículos que não usam o protocolo CAN.

2.2.1.6 Modo 06

O modo 06 apresenta testes de diagnóstico do sensor de oxigênio e de outros sistemas em veículos que utilizam o protocolo CAN.

2.2.1.7 Modo 07

O Modo 07, quando solicitado, responde com códigos de falha não confirmados. Utilizam os mesmos DTCs do Mode 03.

2.2.1.8 Modo 08

Através do Modo 08, é solicitado o controle dos sistemas de bordo por um equipamento de teste externo.

2.2.2.9 Modo 09

O Modo 09 demanda informações do veículo, como por exemplo o número de identificação (VIN).

2.2.2.10 Modo 10

Em base hexadecimal, o Modo 10 é requisitado através do código 0A. Ele exibe códigos de falha permanentes, ou seja, que não podem ser apagados através do Modo 04. Os DTCs usados são os mesmo do Modo 03 e 04.

O sistema OBD-II permite a comunicação entre os módulos veiculares através de diferentes protocolos. Para o protótipo desenvolvido nesse projeto, optou-se por utilizar a comunicação através do protocolo CAN.

2.3 *CONTROLLER AREA NETWORK*

O compartilhamento de dados por uma rede requer a comunicação entre o elemento que envia e o que recebe a informação. Tal comunicação é normatizada pelos chamados protocolos, que são basicamente conjuntos de regras que visam a padronização e a organização da troca de dados entre sistemas computacionais.

Com a maior complexidade e robustez da eletrônica dos automóveis, houve-se a necessidade de se criar um sistema de transmissão de dados em rede entre os módulos de controles veiculares. Atualmente, os protocolos que regem essa comunicação são divididos em classes, de acordo com a SAE (CERVI, 2005). O conjunto de normais mais utilizado em todo o mundo é o protocolo CAN, que se enquadra na classe C, caracterizada por apresentar uma velocidade de transmissão de dados entre 125kbps e 1Mbps e controle de mecanismos em tempo real. Os fundamentos do CAN são especificados pela norma ISO11898 (2003), que determina as características de redes com alta velocidade de transmissão.

O CAN bus, ou barramento CAN, surgiu em 1986 (CAN, 2018), criado pela empresa alemã Bosch. Inicialmente era voltado para indústria automotiva, mas alcançou outros ramos industrias devido à sua eficiência e confiabilidade. A americana Intel lançou, então, em 1987 o primeiro hardware para implementação do protocolo, o chip controlador 82526.

A rede CAN, assim como as demais redes de comunicação de dados, é descrita de acordo com seus parâmetros de operação. Para entender como a mensagem CAN é transmitida, é necessário primeiro compreender sob quais diretrizes ela é gerada e como é qualificada.

2.3.1 Redes Industriais

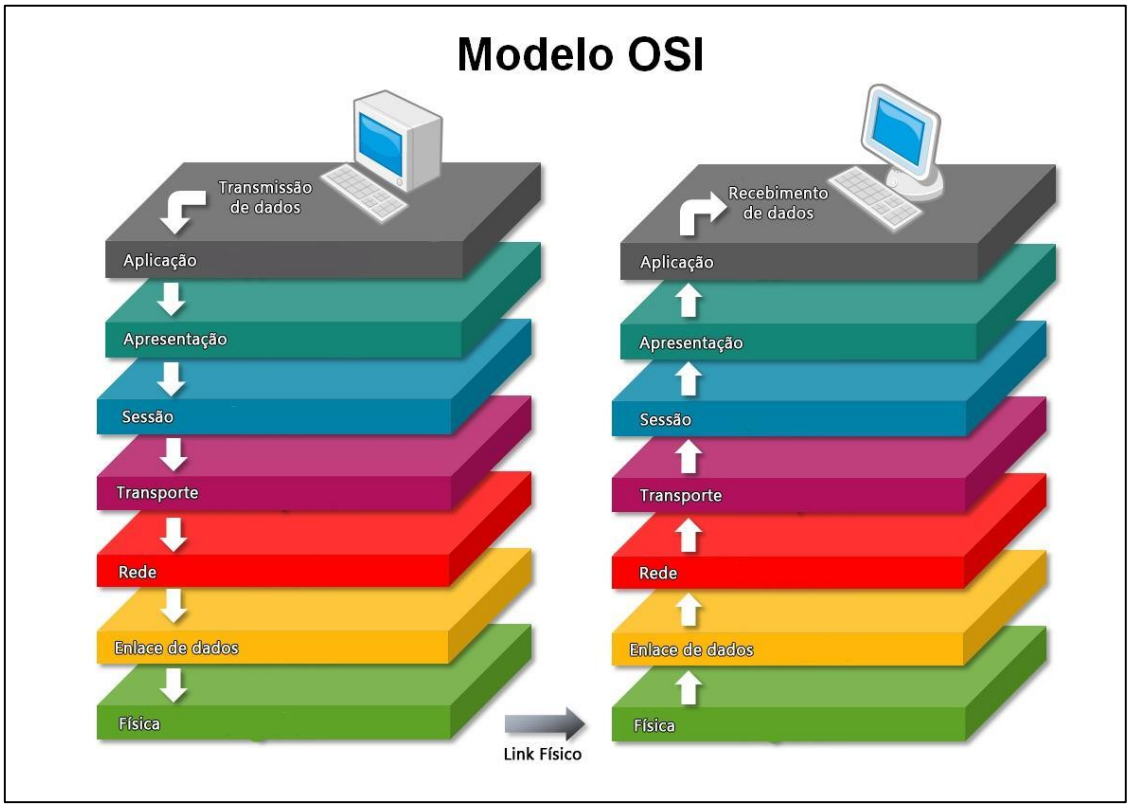
A automação na indústria teve seu crescimento diretamente atrelado à busca por precisão e velocidade no processamento de informações. Para

automatização de qualquer processo é necessária a interligação de todos os componentes do sistema. As redes industrias, portanto, são as responsáveis pela comunicação entre os diferentes elementos de um processo de produção, baseando-se em protocolos que regem essa troca de informações.

Visando uma maior interação de operações entre as redes de comunicação, foi criado a arquitetura RM-OSI (*Referencial Model for Open System Interconnection*) em 1984 pela ISO (PINHEIRO, 2008). Esse modelo passou a servir então como referência na estruturação de redes para permitir a compatibilidade na conexão de sistemas diferentes.

As redes de comunicação são estruturadas em níveis hierárquicos, organização essa gerenciada pelo modelo OSI, que estabelece a divisão da comunicação de dados em 7 camadas (PINHEIRO, 2008). O modelo possibilita a padronização dos componentes da rede e permite a comunicação entre tipos distintos de hardware e software.

Figura 4 – Camadas do Modelo OSI



Fonte: (MODELO, 2013)

A camada mais inferior do Modelo OSI chama-se Camada Física e sua função é garantir uma conexão eficaz entre os elementos do sistema. Ela define as características mecânicas e elétricas da transmissão, bem como os procedimentos de funcionamento. O segundo nível é o Enlace de Dados que comanda: o endereçamento físico, o controle de erros e o controle de fluxo. É também o responsável pelo gerenciamento do protocolo CRC (*Cyclic Redundancy Check*), que será discutido mais adiante;

A terceira camada é a camada de Rede, cuja função é estabelecer a conexão entre os computadores, roteando as informações da origem até a máquina de destino. Acima há a camada de Transporte, que garante a confiabilidade do processo através da transparência na transferência de dados. Realiza uma comunicação ponto-a-ponto assegurando a entrega de informação.

O nível 5 do Modelo OSI administra o diálogo sessão e sincroniza as operações dos computadores da rede e é chamada camada de Sessão. A camada de apresentação, designação da sexta camada, é responsável por formatar e converter linguagens, códigos e sintaxes de maneira que os elementos conectados à rede entendam a informação comunicada. Esse sexto nível realiza também a criptografia e compressão de dados. A última camada do modelo OSI é a camada de Aplicação, que serve como uma interface direta com o usuário e administra o modo como a informação é transmitida. É responsável pela transferência de arquivos e emulação de terminais.

2.3.2 Rede CAN

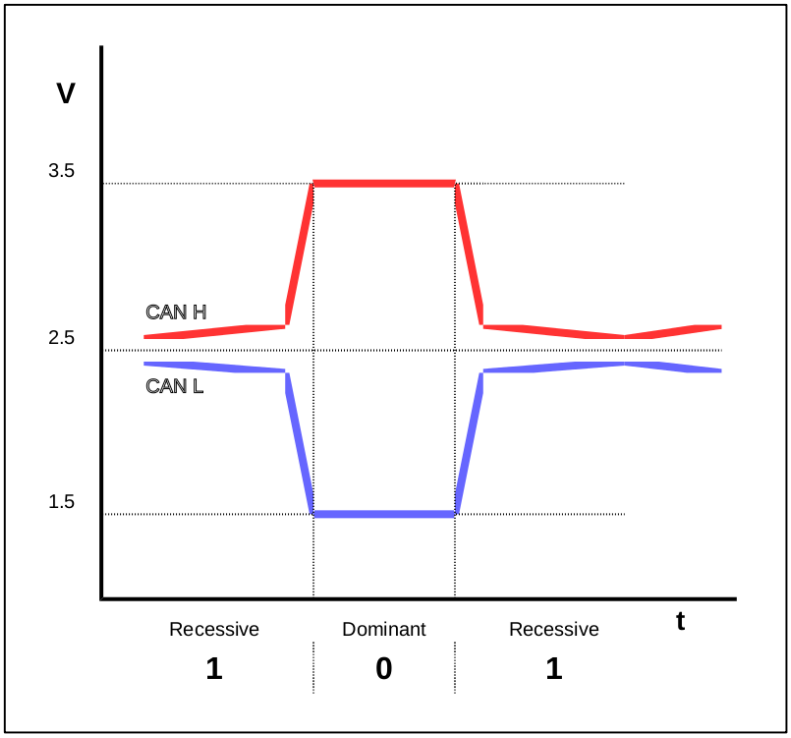
O protocolo CAN abrange apenas as duas camadas inferiores do modelo ISO, a camada Física e o Enlace de Dados. As principais responsabilidades da primeira camada se referem a codificação de bits e a temporização, além de garantir a sincronização desses processos (HUBERT, 2001). A camada física define também as características das conexões dos elementos da rede. O barramento CAN pode ser constituído de três maneiras, de acordo com a quantidade de fios usada, podendo a rede ser baseada em 1, 2 e 4 fios.

Para barramentos CAN de um fio, os dados são transmitidos por essa única via, chamada linha CAN. Sistemas com 2 e 4 fios utilizam sinais CAN_H (CAN High)

e CAN_L (CAN Low), sendo os dados representados pela análise da diferença de potencial entre esses dois sinais. Essa configuração é chamada Par Trançado Diferencial e tem como objetivo diminuir os efeitos de interferências eletromagnéticas na rede. No caso particular dos 4 fios, em adição aos sinais de dados, um fio para alimentação e outro com o aterramento (GND) também fazem parte do barramento.

Na rede CAN, os dados são representados pelo bit recessivo, 1, e pelo bit dominante, como observado na Figura 5. A tensão nominal para o bit recessivo é 2,5V, igualmente para CAN High e CAN Low, portanto a diferença entre as tensões no barramento é 0V. Para o bit dominante, a tensão elétrica nominal é 3,5V para CAN High e 1,5 para CAN Low, causando uma diferença de potencial de 2V.

Figura 5 – Sinal CAN



Fonte: (CAN, 2016)

A norma ISO 11898 (2003) qualifica a CAN como uma rede de prioridade de passagem de dados. Utilizando-se do conceito CSMA/CD with NDA (Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration), o protocolo determina que os módulos analisem o estado dos outros módulos, para detectar qual

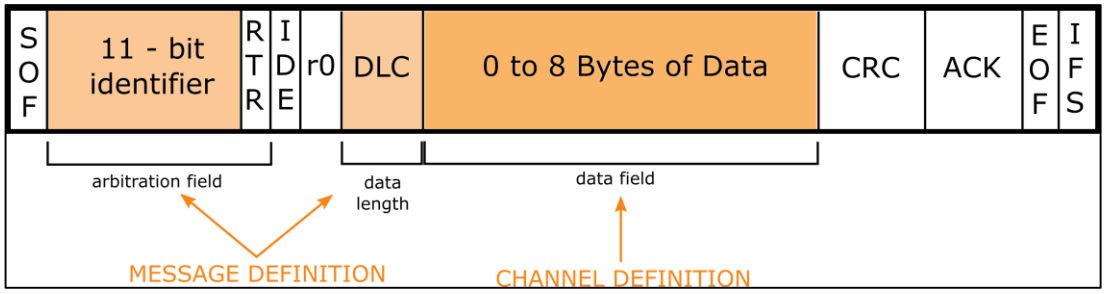
deles está enviando dados com maior prioridade. A transmissão de informações de menor prioridade será cessada em benefício do envio da mensagem de maior prioridade. Portanto, o elemento que está transmitindo uma mensagem com um identificador de menor valor será priorizado. Este processo de acesso ao meio é chamado de Arbitragem.

A rede CAN é classificada como um protocolo de comunicação serial, o que significa que os dados são transmitidos em série, um bit por vez, formando-se um byte após oito pulsos de *clock*. Outra característica importante é o sincronismo, ou seja, a informação é enviada de maneira contínua sem intervalos entre bits. O emissor e o receptor estão sempre sincronizados, pois os dados são transmitidos em intervalos de tempo conhecidos e regulares.

O protocolo CAN permite dois formatos distintos: o CAN 2.0A, que é o formato padrão, e o CAN 2.0B. A diferença principal entre ambas configurações diz respeito ao número de bits, que traduzirão a mensagem a ser transmitida. Em um formato de identificador de 11 bits, segundo o protocolo CAN 2.0A, há a possibilidade de 2048 mensagens, enquanto o modo CAN 2.0B permite a comunicação de 537 milhões de mensagens. Esse quadro de bits está relacionando com especificações da camada de enlace de dados.

Nessa literatura, será descrito apenas o formato padrão CAN 2.0A, empregado nesse projeto por ser o mais amplamente utilizado. Conforme a Figura 6, a estrutura da mensagem CAN possui alguns elementos discriminados a seguir:

Figura 6 – Formato da mensagem CAN



Fonte: (CAN, 2018)

a) SOF (*Start-of-Frame*): é o bit que identifica o início de uma mensagem e é responsável pela sincronização dos elementos conectados ao barramento;

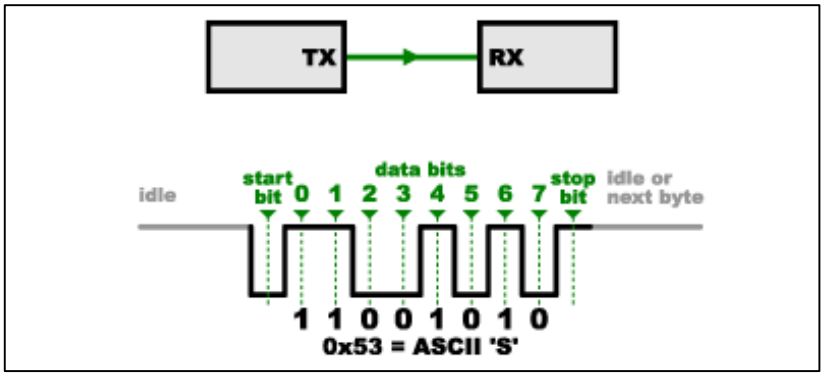
b) Identificador: 11 bits que definem a prioridade da mensagem;
c) RTR (*Remote Transmission Request*): é o bit que faz a requisição de transmissão remota;
d) IDE (*Identifier Extension*): quando esse bit for dominante, o identificador da mensagem é de 11 bits, configurando o formato padrão CAN 2.0A;
e) R0: bit reservado para futuras variações da rede CAN;
f) DLC (*Data Lenght Code*): são 4 bits que indicam o número de bytes do campo de dados;
g) Dados: até 8 bytes (64 bits) que carregam toda a informação transmitida;
h) CRC (*Cyclic Redundancy Check*): Campo composto por 16 bits dedicados à detecção de erros. 15 bits são usados para a implementação do código de detecção de falhas e 1 bit recessivo é usado como delimitador do campo.
i) ACK (*Acknowledge*): Esse campo de confirmação consiste em 1 bit que indica se a mensagem foi recebida corretamente pelo nó de destino e 1 bit recessivo delimitador, resultando em 2 bits de dados;
j) EOF (*End of Frame*): 7 bits indicando o fim da mensagem no barramento CAN;
k) IFS (*Inter-Frame Space*): Esse campo é composto por 7 bits que marcam um espaço entre duas mensagens.

Essas definições tornam a rede CAN uma rede de baixo custo (apenas dois fios), segura (sistema de prioridade e detecção de erros) e veloz (máximo de dados 8 bytes, portanto consegue atingir velocidade de até 1Mbps).

2.4 MÓDULO CAN

Para que o sinal vindo do barramento CAN seja interpretado pelo microcontrolador ATMEGA328p, responsável por gerenciar o processo de monitoramento de dados, faz-se necessária a implementação de um módulo de tradução do protocolo CAN para o protocolo SPI (Serial Peripheral Interface), compreendido pelo microcontrolador. O módulo proposto nessa literatura é composto por um transceptor, o circuito integrado MCP2551, e por um controlador CAN, o circuito integrado MCP2515, ambos da empresa MICROCHIP®. Os dois circuitos se comunicam via serial, na qual os bits de dados são enviados sequencialmente, utilizando as linhas TX para transmissão e RX para o recebimento de dados, como mostrado na Figura 7.

Figura 7 – Comunicação Serial



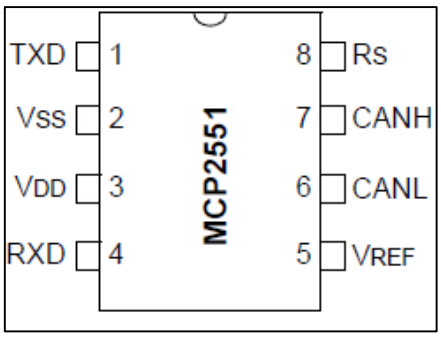
Fonte: (SERIAL, 2018)

2.4.1 MCP2551

O transceptor MCP2551 é o responsável pela interação entre o barramento físico e o controlador CAN MCP2515. Ele atende a norma ISO-11898, referente as requisições da camada física e opera em velocidade de até 1Mbps (MCP2551, 2018).

A função desse circuito é transformar a diferença das tensões de entrada, CAN High e CAN Low, em sinais digitais, nível lógico 0 e 1. Essa saída, representada pelo TXD, será enviada para o controlador através do pino 1, conforme indicado na Figura 8. O pino 4, recebe os bits de resposta vindos do controlador MCP2515, representados pelo sinal RXD. Para cada bit recessivo no barramento CAN, é enviado um bit 1 pelo pino TXD. Em contrapartida, um bit 0 é enviado a cada bit dominante decodificado pelo transceptor. As informações são enviadas bit a bit para o controlador MCP2515, que interpreta a sequência de acordo com o protocolo CAN.

Figura 8 – Sequência dos pinos no circuito integrado MCP2551



Fonte: (MCP2551, 2018)

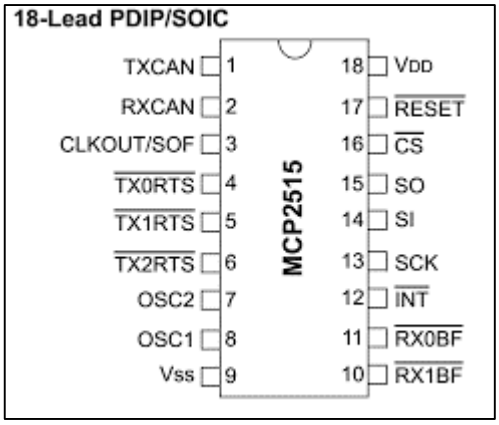
A norma ISO-11898 especifica também que seja colocado um resistor de 120Ω no final de cada barramento CAN, ou seja, entre os pinos 6 e 7. O circuito MCP2551 é alimentado por uma tensão de 5V, no pino 3, enquanto o pino 2 deve ser ligado ao GND. O pino 8, RS, é conectado a um resistor que se conecta ao GND, de modo a controlar a variação das transições de sinal (AN228, 2018). O pino 5 não é utilizado nesse módulo.

2.4.2 MCP2515

A função do controlador MCP2515 é decodificar os sinais digitais enviados pelo transceptor, baseado na rede CAN. Além de receber e enviar mensagens no formato definido pelo protocolo *Controller Area Network*, ele se comunica com demais controladores via *Serial Peripheral Interface* (SPI). Esse circuito possibilita a comunicação SPI de alta frequência, chegando a até 10MHz (MCP2515, 2018).

Conforme observado na Figura 9, a alimentação desse circuito integrado é dada pelo pino 18 e deve ser de 5V, enquanto o GND é conectado ao pino 9. Os pinos 7 e 8 são os pinos de saída e entrada, respectivamente, do oscilador, que é o elemento responsável por manter a frequência do circuito. Para essa abordagem do módulo CAN, foi escolhido um cristal oscilador de 16MHz. O pino 17 tem a função de reset do hardware e deve ser conectado a um resistor, de 10kΩ nessa configuração, conectado a alimentação do circuito.

Figura 9 – Sequência dos pinos no circuito integrado MCP2515



Fonte: (MCP2515, 2018)

Os pinos 1 e 2 fazem o envio e a recepção dos dados transmitidos pelo transceptor MCP2551. Os pinos 12, 13, 14, 15 e 16 são direcionados para a função de comunicação SPI, enquanto os demais pinos não são utilizados nesse módulo.

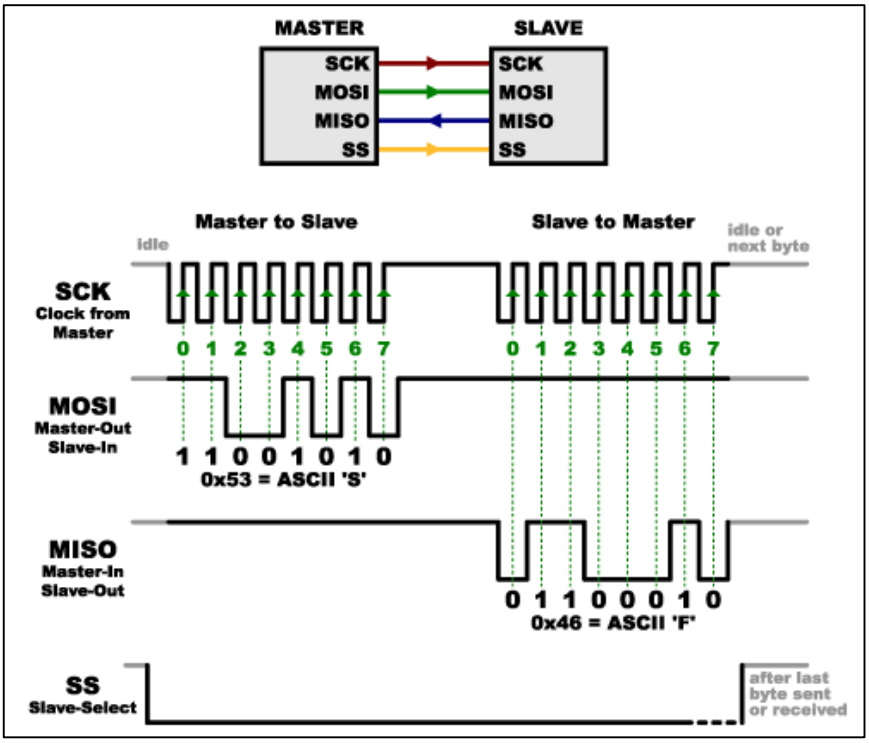
2.5 SERIAL PERIPHERAL INTERFACE

A comunicação entre computadores de um sistema pode se dar de forma serial ou paralela. Quando serial, pode ser classificada como síncrona ou assíncrona. Na comunicação assíncrona, os dados são transmitidos de forma irregular, podendo ser enviados em qualquer ordem, ocasionando uma baixa eficiência na transmissão. A vantagem da comunicação síncrona de dados está no fato de que a informação é transmitida e recebida num instante de tempo definido. O transmissor e receptor, devem estar sincronizados, ou seja, quando um bloco de dados é enviado, o receptor é bloqueado e o transmissor só pode enviar outro bloco quando o primeiro for recebido pelo receptor.

Na comunicação serial síncrona é estabelecida a definição de Mestre-Escravo, onde o Mestre (Master) é o elemento gerador do sinal de sincronismo (SACCO, 2014). Os demais dispositivos conectados ao sistema são chamados de escravos (Slaves). Uma das transmissões síncronas mais utilizadas, em virtude da sua simplicidade, é a *Serial Peripheral Interface*, conhecida como SPI. A comunicação SPI se caracteriza por ser *full-duplex* (SACCO, 2014), o que significa que simultaneamente o mestre envia um bit de dado e recebe um bit do escravo.

Quando implementada, a rede necessita de um mínimo de 4 fios, sendo 3 conexões + 1 para cada escravo conectado ao mestre. O pino SCK (Serial Clock), também conhecido por CLK, gera o *clock*, o sinal que temporiza a comunicação. A linha MOSI (Master Output/Slave Input) tem a função de trafegar os dados que saem do mestre em direção aos escravos. A terceira conexão é denominada de MISO (Master Input/Slave Output) e tem a tarefa contrária à da MOSI, trafegando os dados que saem do escravo e devem chegar ao mestre. O outro fio que liga o mestre ao escravo, ou escravos da rede, é o SS (Slave Select). O mestre pode ter várias conexões SS (SS1, SS2, etc.), que dependem diretamente do número de escravos. Para seleção do dispositivo com o qual se deseja comunicar, o pino SS respectivo deve estar em nível logico baixo. Estas ligações podem ser observadas na Figura 10.

Figura 10 – Comunicação SPI



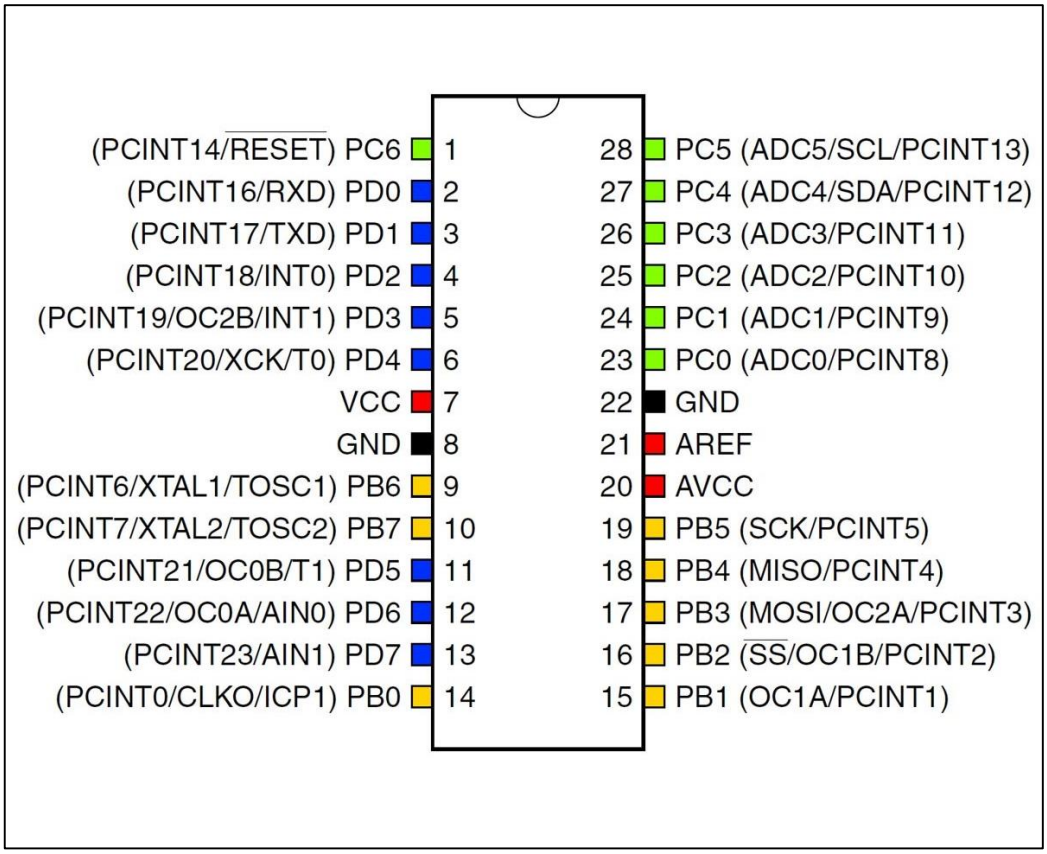
Fonte: (SERIAL, 2018)

2.6 ATMEGA328P

O microcontrolador ATMEGA328p, fabricado pela empresa Atmel® (ATMEL, 2016), tem o papel de gerenciar os processos de todo o sistema. Ele foi escolhido por ser um sistema simples e barato, conseguindo atender a todos os requisitos do projeto. Este circuito integrado é o mesmo microcontrolador usado na plataforma de prototipagem Arduino UNO®, que é uma plataforma modulada, configurável por firmware e com um microprocessador de código aberto.

O Arduino conta com um ambiente de desenvolvimento *open-source*, oferecendo ao usuário acesso às mais diversas bibliotecas escritas para diferentes funcionalidades. Sendo assim, o microcontrolador ATMEGA328p, utilizado nesse projeto, usufrui das mesmas vantagens dessa plataforma de prototipagem. Os pinos desse microcontrolador usados na comunicação SPI são os mesmos designados pelo fabricante. As saídas 19, 20, 21 e 22 do microcontrolador são respectivamente as linhas SCK, MISO, MOSI e SS que se comunicam com o controlador MCP2515.

Figura 11 – Sequência dos pinos no circuito integrado ATMEGA328p



Fonte: (ATMEGA328P-PU, 2018)

2.7 *GLOBAL POSITION SYSTEM*

Uma das atribuições do sistema de monitoramento proposto nesse trabalho é fornecer a localização geográfica do veículo ao usuário. Para tal, fez-se uso de um receptor GPS, fabricado pela empresa GlobalSat. Ele se conecta ao microcontrolador por comunicação serial através de um pino receptor e um pino transmissor, que se conectam ao pino transmissor e ao pino receptor, respectivamente, do ATMEGA328p. Segundo o manual do usuário desse dispositivo (GLOBALSAT, 2018), para seu funcionamento, ele precisa ser alimentado por uma tensão de 5V, além de ter dois pinos conectados ao GND do sistema.

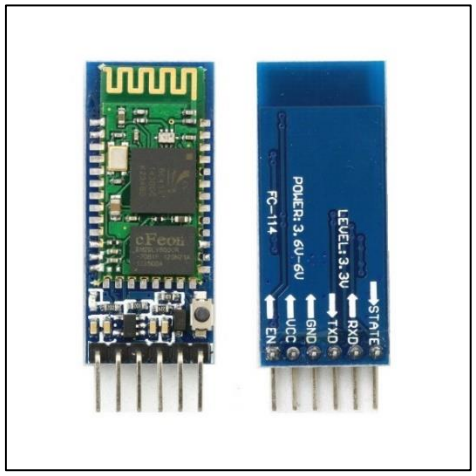
Figura 12 – Receptor GPS

Fonte: (GPS, 2018)

2.8 *BLUETOOTH*

A comunicação remota entre o protótipo e o dispositivo na qual o usuário deseja ler as informações do veículo pode se dar de diversas maneiras, como por exemplo através de um módulo GPRS ou um módulo 3G. Para esse projeto, optou-se pelo uso de um módulo Bluetooth, tornando possível a comunicação com o celular do usuário, sem a necessidade de haver uma rede de telefonia ou sinal de internet disponíveis. Esse dispositivo foi escolhido devido ao seu preço e simplicidade de utilização. Ele se conecta com o protótipo através de comunicação serial e é alimentado pela mesma tensão dos demais componentes do sistema (5V).

Figura 13 – Módulo Bluetooth



Fonte: (HC-05, 2018)

3 MATERIAIS E MÉTODOS

Para que os dados provenientes da porta OBD-II possam ser monitorados e processados, faz-se necessário o desenvolvimento de um módulo de decodificação das mensagens CAN. Para o gerenciamento dos processos do sistema de monitoramento foi utilizado o microcontrolador ATMEGA328p, que além de comandar a transmissão das informações da rede CAN, também controla o funcionamento do GPS e do módulo *bluetooth*.

3.1 MATERIAIS

Nesse tópico são descritos os materiais utilizados em todas as fases desse projeto, seja na concepção, testes ou na construção do protótipo final.

3.1.1 Emulador OBD-II

Para o desenvolvimento do protótipo, fez-se uso de um emulador OBD-II com simulador de barramento CAN, apresentado na Figura 14. Esse emulador propiciou que os testes iniciais fossem feitos em laboratório, sem a necessidade de um veículo. O produto Freematics OBD-II Emulator MK2, fabricado e comercializado pela empresa australiana *Freematics* (FREEMATICS, 2018), fornece sinais que replicam dados provenientes da porta OBD-II de um carro real.

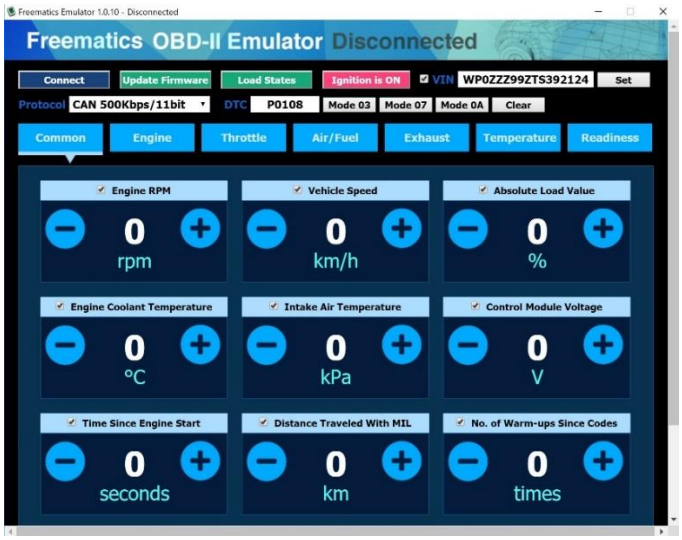
Figura 14 – Emulador OBD-II Freematics

Fonte: (FREEMATICS, 2018)

O emulador ainda conta com uma plataforma virtual (GUI), apresentada na Figura 15, para simulação das variáveis de estado do carro, onde o usuário pode

definir os valores de alguns parâmetros, como velocidade e temperatura do motor, além inserir códigos de falha.

Figura 15 – GUI emulador

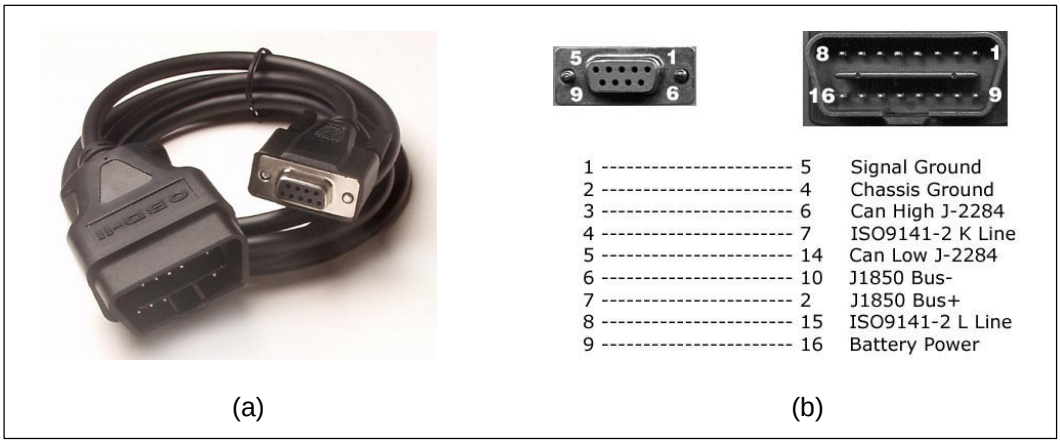


Fonte: Autoria própria

3.1.2 Cabo OBD-II/DB9

A conexão entre a porta OBD-II e o hardware embarcado é feita através de um cabo conversor, ilustrado na Figura 16a, contendo um conector de 16 pinos padrão J1962 e um conector 9 pinos padrão DB9. Os pinos correspondentes de um conector para o outro podem ser observados na Figura 16b. Os 5 pinos utilizados no hardware desenvolvido são: Signal Ground, Chassis Ground, Battery Power, CAN High e CAN Low.

Figura 16 – Cabo OBD-II/DB9 (a) cabo, (b) pinos correspondentes



Fonte: (OBD2, 2018)

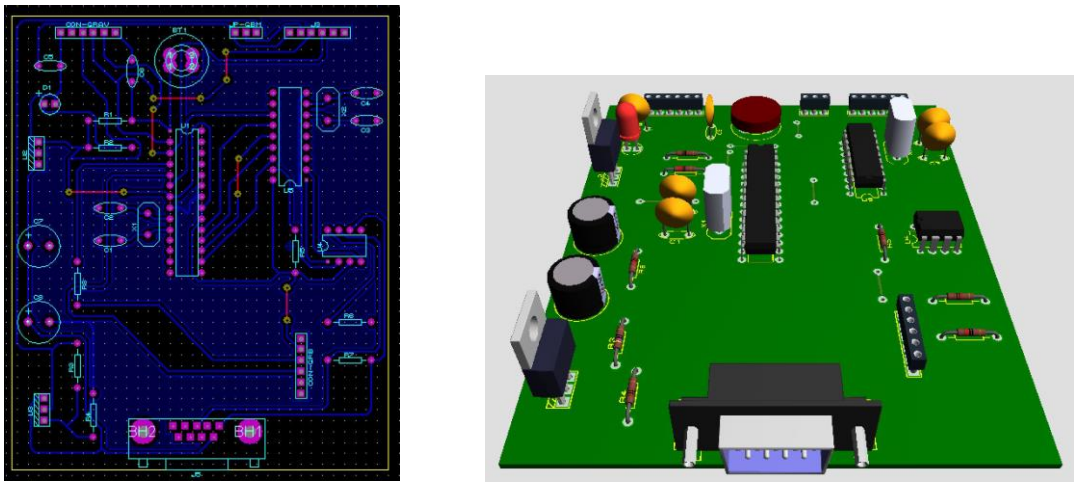
3.1.3 Desenvolvimento do Hardware

Foi desenvolvido um sistema embarcado, composto por uma placa dedicada com a finalidade de transmitir e processar os dados do veículo, bem como alimentar todos os circuitos componentes do sistema. Para confecção da placa, primeiramente foi definido todo o circuito eletrônico necessário para implementar os requisitos do protótipo. O software Proteus foi utilizado para fazer o diagrama do circuito e projetar o layout de PCB, ou seja, desenhar a placa de circuito impresso, que foi dividido em módulos, de acordo com a funcionalidade. Os diagramas de circuito podem ser visualizados no Apêndice A e estão divididos da seguinte maneira:

a) Alimentação: A fonte de energia de todo o circuito integrado da placa será a própria bateria do carro. A porta OBD-II, além de permitir o acesso aos dados da ECU, também é fonte de tensão para qualquer dispositivo conectado a ela. Os componentes da placa necessitam de uma tensão de 5V para um funcionamento correto, porém, a bateria padrão dos automóveis fornece 12V. Através de um regulador de tensão, a voltagem é então diminuída para a tensão requerida pelos componentes eletrônicos. Foi inserido também um regulador de tensão ajustável na intenção de prover 4V para o circuito, caso algum componente precisasse ser alimentado com uma tensão menor.
b) Circuito do módulo CAN: como explanado no tópico anterior, o módulo de aquisição da mensagem CAN é composto por um transceptor MCP2550 e um controlador MCP2515. A entrada desse circuito são os sinais CAN High e CAN Low, provenientes da porta OBD-II e que chegam ao circuito através dos pinos 3 e 5 do conector DB9.
c) Circuito do Microcontrolador: O microcontrolador ATMEGA328p é responsável por administrar todos os processos do sistema. Seu circuito conta com um sistema oscilador, além de um botão de reset, conectado ao pino 1. Os pinos 0 e 1 são designados para a conexão com o módulo bluetooth, enquanto os pinos 2 e 3 fazem a ligação com o módulo GPS.
d) Conectores: A placa ainda conta com conectores, que fornecem a ligação para o receptor GPS e as conexões com os pinos TX e RX do microcontrolador. Através desses pinos, é feita a comunicação serial com o módulo de comunicação remota, além de possibilitarem a comunicação com o conversor USB/Serial, responsável por fazer o upload do programa para o ATMEGA328p.

Feitos os diagramas esquemáticos, o software gera um layout da placa interligando os componentes eletrônicos através de trilhas de circuito, com pode ser observado na Figura 17.

Figura 17 – Placa de circuito impresso (a) Layout, (b) Visualização 3D

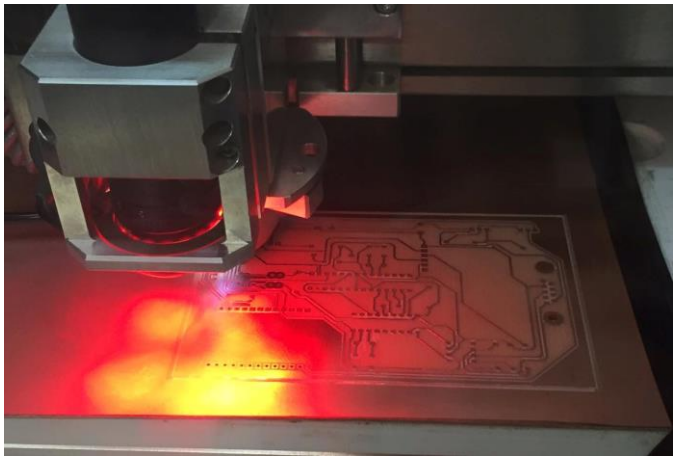


(a) (b)

Fonte: Autoria Própria

A placa foi confeccionada por uma impressora no laboratório de Projeto Integrador do IFSC e pode ser observada na Figura 18.

Figura 18 – Confecção da placa

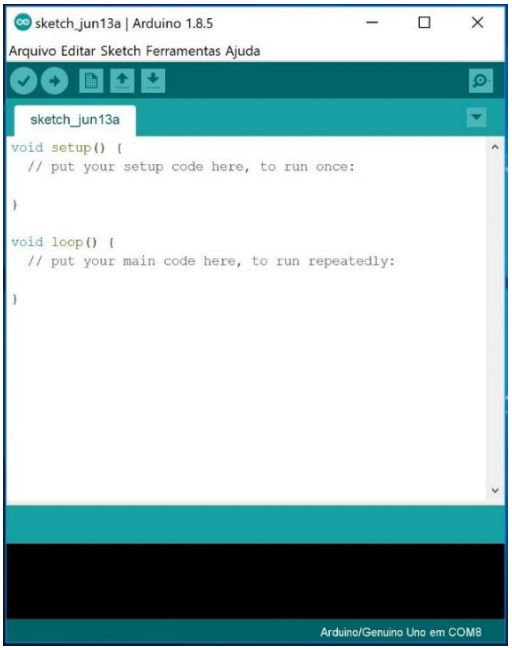


Fonte: Autoria Própria

3.1.4 Software

O microcontrolador usado no protótipo é o mesmo computador que comanda a plataforma de prototipagem Arduino. Pelo fato de ser um software aberto e que contém bibliotecas de código acessíveis, optou-se por utilizar o Ambiente de Desenvolvimento (Integrated Development Environment, em inglês) do Arduino para programar o código de gerenciamento dos processos mais superficiais do sistema, como o comando do GPS e do módulo Bluetooth. A Figura 19 apresenta uma visão geral da plataforma.

Figura 19 – Arduino IDE



Fonte: Autoria Própria

3.2 MÉTODOS

Essa etapa do trabalho apresenta as ferramentas e técnicas utilizadas no projeto, descrevendo o desenvolvimento do sistema de aquisição de dados e o funcionamento do aplicativo criado.

3.2.1 Programação do Sistema de Aquisição de Dados

O ambiente de desenvolvimento do Arduino foi usado para construir e carregar o código de gerenciamento das tarefas do sistema de monitoramento. Utilizando bibliotecas já existentes, para implementação do módulo GPS e para

comunicação serial dos dispositivos presentes, essa plataforma aberta foi escolhida devido à simplicidade e acessibilidade das bibliotecas necessárias para o projeto.

Entretanto, o código principal de leitura e compreensão da mensagem CAN foi desenvolvido no Software CodeBlocks, onde fez-se uso de bibliotecas referentes a comunicação SPI. Nesse código, são definidos os pinos do microcontrolador utilizados na comunicação com o módulo, bem como a configuração dos dados recebidos, de forma a decodificar a mensagem transmitida sob as premissas do protocolo Controller Area Network.

Primeiramente é definido o formato dessa mensagem, através de uma *struct*, um registro de variáveis de diferentes formatos. Seguindo o protocolo CAN, a mensagem deve conter uma identificação de 11 bits, portanto utiliza-se uma variável *unsigned int* (inteiro sem sinal) de 16 bits para guardar esse valor. Na sequência, o campo da mensagem deve conter um bit RTR e o tamanho da mensagem (*lenght*), guardado em 4 bits. Como mostrado na Figura 20, os 8 próximos bytes se referem ao campo de dados e irão guardar a mensagem transmitida. Os demais bits têm configuração padrão já definida e não precisam ser especificados nessa estrutura, que é necessária pois esses são valores variáveis.

Figura 20 – Estrutura da Mensagem CAN

```
46
47  typedef struct
48  {
49      uint16_t id;
50      struct {
51          int8_t rtr : 1;
52          uint8_t length : 4;
53      } header;
54      uint8_t data[8];
55  } tCAN;
56
```

Fonte: Autoria Própria

Seguindo o padrão da *struct* criada, uma mensagem com a solicitação de dados para a rede CAN é enviada. Como mostra a Figura 21, o primeiro byte da mensagem avisa ao sistema quantos bytes devem ser lidos na sequência. Para o caso de requisição de códigos de falha, modo 03, apenas o byte seguinte deve ser lido, sendo esse o dado que contém a informação do modo de operação.

Figura 21 – Mensagem Modo 03

```
311     message2.id = PID_REQUEST;
312     message2.header.rtr = 0;
313     message2.header.length = 8;
314     message2.data[0] = 0x01;      //deve ler o proximo byte
315     message2.data[1] = 0x03;      //modo
316
317     message2.data[2] = 0x00;
318     message2.data[3] = 0x00;
319     message2.data[4] = 0x00;
320     message2.data[5] = 0x00;
321     message2.data[6] = 0x00;
322     message2.data[7] = 0x00;
323
```

Fonte: Autoria Própria

A mensagem de resposta da rede CAN terá o mesmo formato da mensagem enviada, porém os dados serão diferentes. O primeiro byte (message2.data[0]) constará o número de bytes seguintes que representam a mensagem completa. A segunda sequência de bits é o modo de operação, que deve ser o mesmo da requisição, confirmando assim a comunicação da mensagem.

O terceiro byte informa o número de DTCs existentes. Como restam apenas 5 bytes de informação, e cada DTC é composto por 2 bytes, no máximo dois códigos de falha podem ser enviados por mensagem (single frame). Portanto, os bytes message2.data[3] e message2.data[4] traduzem o primeiro código de erro, caso existente.

De acordo com a estrutura do DTC (CÓDIGOS, 2018), o primeiro dígito do código refere-se ao tipo de sistema onde a falha se localiza e é identificado por uma letra, que pode ser P, C, B ou U. Essa informação, no entanto, está codificada nos primeiros dois bits do byte mensagem2.data[3]. Para acessar esse dado é feito o *bitmasking* (Figura 22), com a função de isolar os bits da mensagem.

Figura 22 – *Bitmasking*

```
dado1 = B11000000 & message2.data[3];
dado2 = B00110000 & message2.data[3];
dado3 = B00001111 & message2.data[3];
dado4 = B11110000 & message2.data[4];
dado5 = B00001111 & message2.data[4];

a=dado1>>6;

switch(a)
    {
        case 0:
        letra='P';
        break;

        case 1:
        letra='C';
        break;

        case 2:
        letra='B';
        break;

        case 3:
        letra='U';
        break;
    }
```

Fonte: Autoria Própria

Para esse sistema de monitoramento, foram explorados o modo 03, para detecção de falhas, e o modo 01, para aquisição de dados simultâneos do estado dos módulos veiculares. No caso do modo 01, a mensagem enviada a ECU (Figura 23) difere levemente da mensagem do modo 03. O primeiro byte deve conter o valor 2, em formato hexadecimal, pois os próximos dois bytes devem ser lidos pelo computador do veículo. O segundo byte refere-se ao modo e o terceiro byte deve conter o PID, informação obrigatória para o modo 01, pois nele consta para qual sensor deve ser solicitada a informação.

Figura 23 – Mensagem Modo 01

```
108     float engine_data;
109     int timeout = 0;
110     char message_ok = 0;
111     message.id = PID_REQUEST;
112     message.header.rtr = 0;
113     message.header.length = 8;    //8 bytes
114     message.data[0] = 0x02;      //deve ler os proximos dois bytes
115     message.data[1] = 0x01;      //modo
116     message.data[2] = pid;       //pid
117     message.data[3] = 0x00;
118     message.data[4] = 0x00;
119     message.data[5] = 0x00;
120     message.data[6] = 0x00;
121     message.data[7] = 0x00;
```

Fonte: Autoria Própria

A mensagem de resposta para o Modo 01 segue também a estrutura definida previamente. O campo de dados comporta até 8 bytes de informações. O primeiro byte, no entanto, representa a quantidade de bytes de informação efetiva, ou seja, quantos bytes foram utilizados para descrever o parâmetro requisitado. O segundo refere-se ao modo da mensagem, enquanto o terceiro byte traz a informação do PID em formato hexadecimal. Portanto, sobram apenas cinco bytes para os dados do sistema. O último byte, entretanto, normalmente não é utilizado, o que resulta em 4 bytes para descrever a informação pedida, que são chamados de A, B, C e D. O byte A é o primeiro a ser enviado, e caso a mensagem seja maior que um byte, os demais bytes são preenchidos sucessivamente até completar a mensagem. Caso a identificação da mensagem recebida pelo sistema de monitoramento seja do tipo resposta (*reply*) e o PID seja o mesmo requisitado, os dados lidos serão interpretados de acordo com o sensor solicitado, como mostra a Figura 24.

Figura 24 – Código PID

```
if((message.id == PID_REPLY) && (message.data[2] == pid))
{

    switch(message.data[2])
    {
        case ENGINE_RPM:            //   ((A*256)+B)/4     [RPM]
        engine_data =  ((message.data[3]*256) + message.data[4])/4;
        sprintf(buffer,"%d rpm ",(int) engine_data);

        break;

        case ENGINE_COOLANT_TEMP:   //  A-40
        engine_data =  message.data[3] - 40;
        sprintf(buffer,"%d degC",(int) engine_data);

        break;

        case VEHICLE_SPEED:         // A
        engine_data =  message.data[3];
        sprintf(buffer,"%d km ",(int) engine_data);

        break;

        case MAF_SENSOR:            // ((256*A)+B) / 100
        engine_data =  ((message.data[3]*256) + message.data[4])/100;
        sprintf(buffer,"%d g/s",(int) engine_data);

        break;
```

Fonte: Autoria Própria

A interpretação dos dados lidos em valores efetivos de acordo com o PID solicitado é determinada de acordo com a norma SAE J1979 (2006). Através dos valores descritos pela Tabela 1, é enviado o PID em hexadecimal solicitando o sensor cuja leitura é desejada. A informação é calculada de acordo com os bytes recebidos e transcrita em valores inteiros para o usuário. O cálculo de alguns parâmetros pode ser observado na Tabela 2.

Tabela 2 – Cálculo dos valores de PIDs (Parâmetros de Identificação).

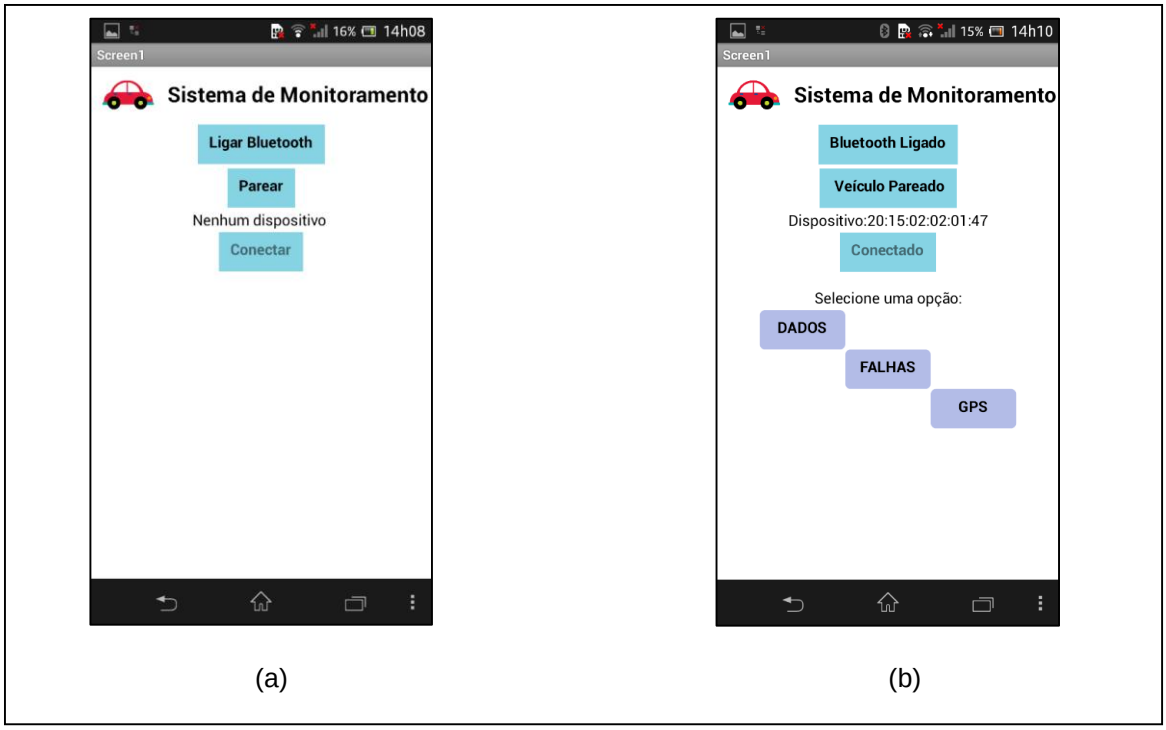
PID	DESCRIÇÃO	UNIDADE	CÁLCULO
0C	RPM	rpm	$\frac{256 * A + B}{4}$
0D	Velocidade	km/h	A
0A	Pressão Combustível	kPa	3*A
2F	Nível Combustível	%	$\frac{100}{255} * A$
5	Temperatura Motor	°C	A-40
11	Posição Acelerador	%	$\frac{100}{255} * A$
33	Pressão Barométrica	kPa	A
5E	Consumo Combustível	L/h	$\frac{256 * A + B}{20}$

Fonte: Autoria Própria

3.2.2 Aplicativo Android

Na intenção de maximizar e facilitar a experiência do usuário com o sistema de monitoramento, foi desenvolvido um aplicativo para sistema Android, no qual é possível fazer a conexão com o protótipo e requisitar as informações clicando nas opções disponíveis. Ele apresenta funções de conectividade do dispositivo *bluetooth* e as opções de serviço são: requisição de dados, detecção de falhas e informe da localização geográfica do carro. A tela inicial do aplicativo e a tela com as opções de serviço são ilustradas na Figura 25.

Figura 25 – Telas Iniciais do aplicativo (a) Tela de conexão, (b) Tela de opções



Fonte: Autoria Própria

As funções de conectividade do celular com o sistema de monitoramento podem ser divididas em:

a) Ligar *Bluetooth*: Para a interação com o protótipo, o primeiro requisito é estar com o sistema *bluetooth* do celular ligado, portanto é mostrado ao usuário a opção de ligar essa conexão através do próprio aplicativo, excluindo a necessidade de sair do aplicativo para configurar a rede *bluetooth*;
b) Parear Dispositivo: Através da opção de pareamento, é possível sintonizar o módulo de conexão remota da placa sem precisar fazê-lo previamente ou fechar o aplicativo para isso. Após ter um dispositivo pareado ao celular, a opção “Conectar” é habilitada;
c) Conectar: O botão “Conectar” possibilita ao usuário escolher qual dos dispositivos pareados ao celular ele deseja conectar ao aplicativo. Somente após efetuar a conexão, é mostrado ao usuário as opções de requisição disponibilizadas;

Os serviços que o aplicativo fornece ao usuário estão ilustrados no Apêndice B e são descritos a seguir:

a) Dados: Quando a opção dados é clicada, outra tela é atualizada no aplicativo. Nela o usuário tem a opção de escolher quais parâmetros do veículo ele deseja ler, como mostrado na Figura Sendo a primeira versão de testes, foram definidas 4 solicitações como exemplo, mas essa opção pode ser configurada para apresentar qualquer PID desejado;
b) Falhas/DTCs: Essa é talvez a opção mais importante do sistema, pois permite ao usuário saber quando há algum problema no seu veículo e, principalmente, em que sistema ou componente está a falha;
c) GPS: Quando clicada, a opção GPS fornece ao usuário a localização precisa do veículo, em formato de coordenadas. Nessa tela, quando solicitado, o usuário consegue visualizar a latitude e a longitude no qual o carro se encontra.

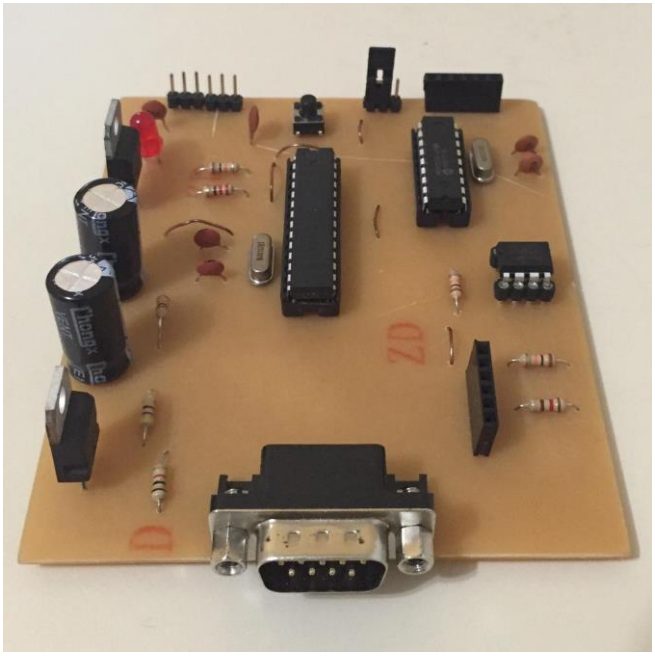
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Após a descrição do desenvolvimento do protótipo no capítulo anterior, esse tópico abordará o resultado e o funcionamento do mesmo. Tanto a parte de hardware quanto a de software atingiram os objetivos desejados na fase de concepção do projeto. O hardware embarcado foi testado primeiramente com a ajuda do emulador *Freematics* e, posteriormente, em dois modelos de carros.

Com o objetivo de propor uma interface amigável e de fácil acesso para o usuário, foi desenvolvido um aplicativo, que através do módulo de conexão remota recebe os dados provenientes da porta OBD-II e os apresenta no celular no qual foi instalado.

A placa desenvolvida com recursos do IFSC apresentou resultados satisfatórios, mantendo a eficiência mesmo após longos períodos de teste. O resultado final pode ser observado na Figura 26.

Figura 26 – Placa de circuito finalizada

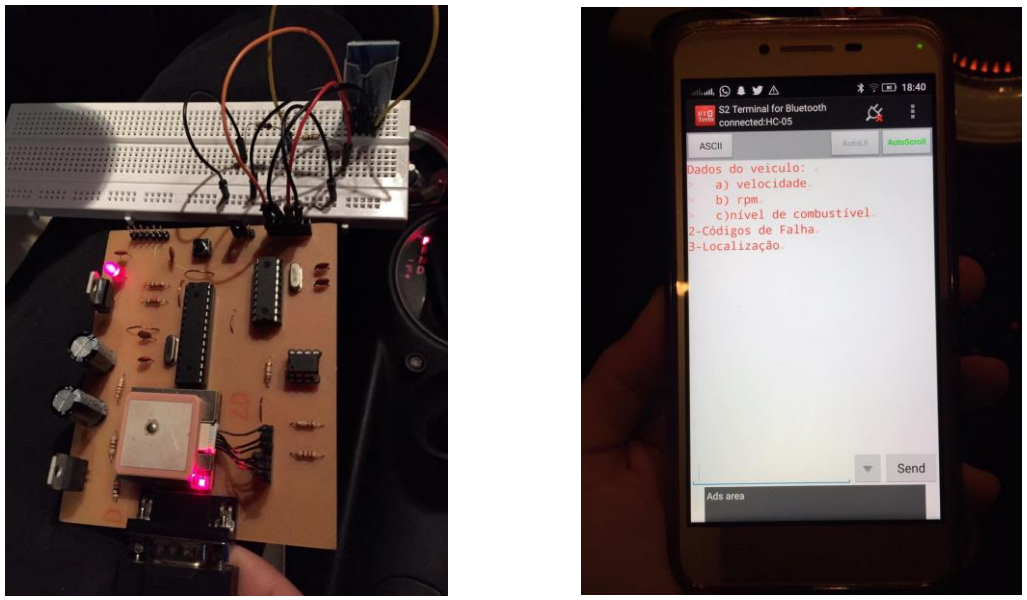


Fonte: Autoria Própria

4.1 TESTES INICIAIS

O teste inicial com veículo foi feito anteriormente ao desenvolvimento do aplicativo, portanto, foi utilizado um terminal Bluetooth instalado no celular para que o usuário fizesse a solicitação das informações à central eletrônica do carro modelo Mini Cooper. Na Figura 27, observa-se o protótipo em funcionamento e a necessidade de se usar uma *protoboard* para fixação do módulo *bluetooth*. Optou-se por deixar esse componente isolado da placa, para que a comunicação remota pudesse ser implementada de maneiras diferentes no futuro. O módulo é alimentado pela placa, que apresenta um conector com saída de sinais GND e 5V. Além dessas duas ligações, ele se conecta com o sistema através dos pinos RX e TX do ATMEGA328p. Na *protoboard*, foram colocados dois resistores, resultando em um divisor de tensão, pois a entrada RX do módulo *bluetooth* aceita apenas 3.3V.

Figura 27 – Sistema de Monitoramento (a) protótipo (b) terminal bluetooth

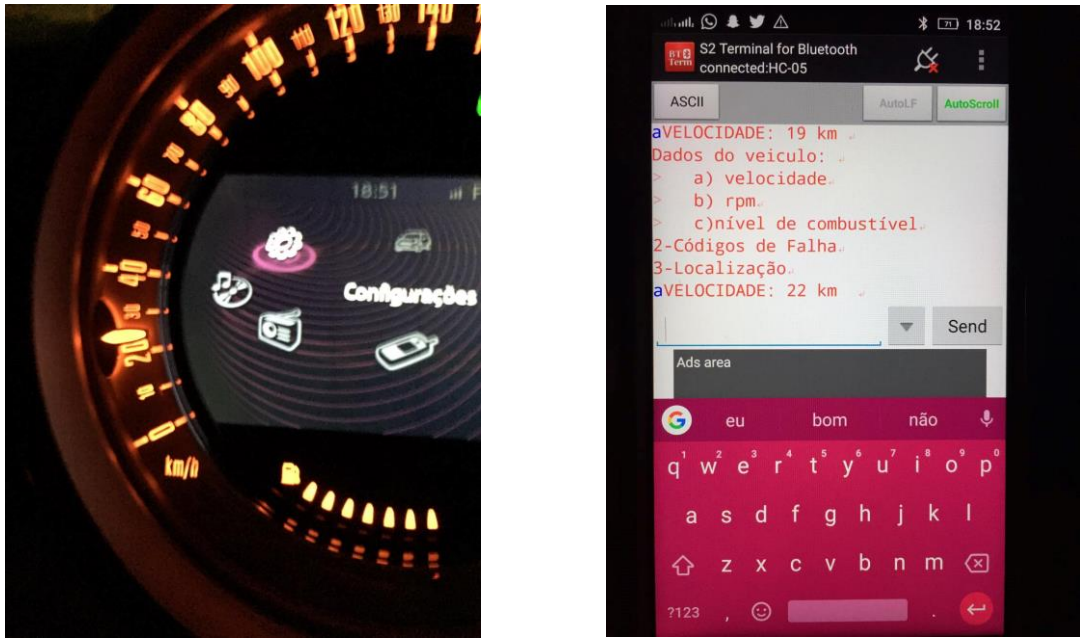


(a) (b)

Fonte: Autoria Própria

Como mostrado a seguir na Figura 28, quando requisitada a velocidade do veículo, o terminal apresenta com precisão o valor do parâmetro em tempo real. O primeiro teste demonstrou a fidelidade das informações recebidas, em tempo real, em relação ao estado do carro.

Figura 28 – Teste Inicial (a) Velocímetro, (b) Dado recebido



(a) (b)

Fonte: Autoria Própria

Todos os parâmetros requisitados através do terminal foram respondidos com precisão pelo sistema de monitoramento. Como o veículo em questão estava em perfeito estado, não foi possível detectar nenhum código de falha, portanto, os testes de verificação de DTCs foram realizados com o emulador e também obtiveram resultados confiáveis.

4.2 RESULTADOS

O sistema de monitoramento atendeu com eficiência e precisão a todos os requisitos esperados, concretizando todos os objetivos pré-definidos. Sendo um sistema embarcado e contendo todos os sistemas eletrônicos na mesma placa, o protótipo pode ser instalado e transportado facilmente. O aplicativo contribui para a melhor utilização do sistema, por ser uma ferramenta de acesso simples e rápido, dispensando o usuário de qualquer conhecimento prévio sobre a mecânica do carro ou a eletrônica do protótipo. A Figura 29 mostra o teste final com o aplicativo, realizado no carro Ford Fiesta, demonstrando a fidelidade da leitura da variável RPM pelo sistema ao valor real mostrado no painel do carro.

Figura 29 – Teste Final

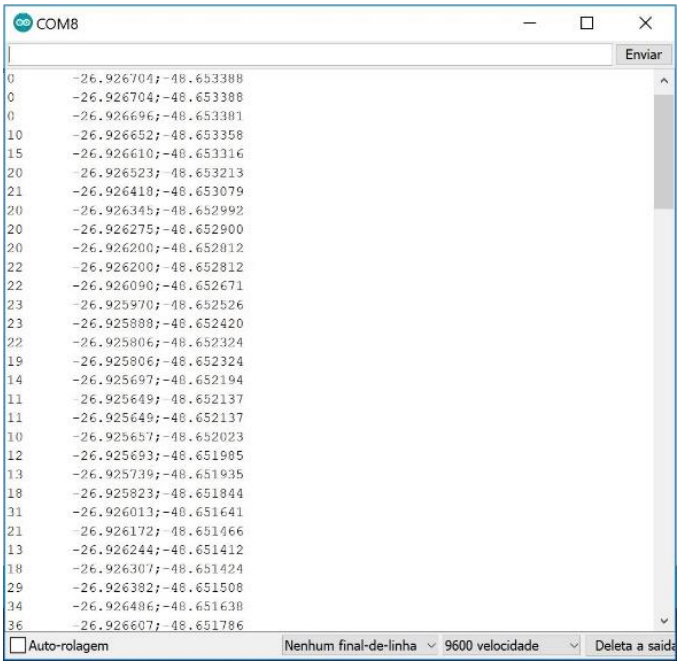
Fonte: Autoria Própria

Ambos os carros testados, bem como o emulador *Freematics*, serviram para corroborar a eficácia do sistema de monitoramento proposto nesse documento. Realizando a aquisição de dados simultaneamente e os enviando com rapidez e precisão de valores, o hardware desenvolvido conseguiu efetuar o diagnóstico dos módulos veiculares e o aplicativo ilustrou de maneira clara essas informações de para o usuário.

4.2.1 Aplicações

Uma maneira de exemplificar os resultados atingidos nesse projeto, e demonstrar a importância para futuras aplicações, foi gerar um gráfico que ilustrasse a variação de algum parâmetro automotivo ao longo de um determinado período. Primeiramente, foi feito um código que obtivesse um número específico de valores lidos via porta OBD-II. A aquisição foi feita a cada 1 segundo e visualizada através do monitor serial do Arduino no computador, para facilitar a leitura dos dados. Essa ação registrou a velocidade (coluna da esquerda) e a localização do veículo (coluna da direita) durante o percurso de uma volta em um quarteirão, mostrado na Figura 30.

Figura 30 – Aquisição de dados simultâneos

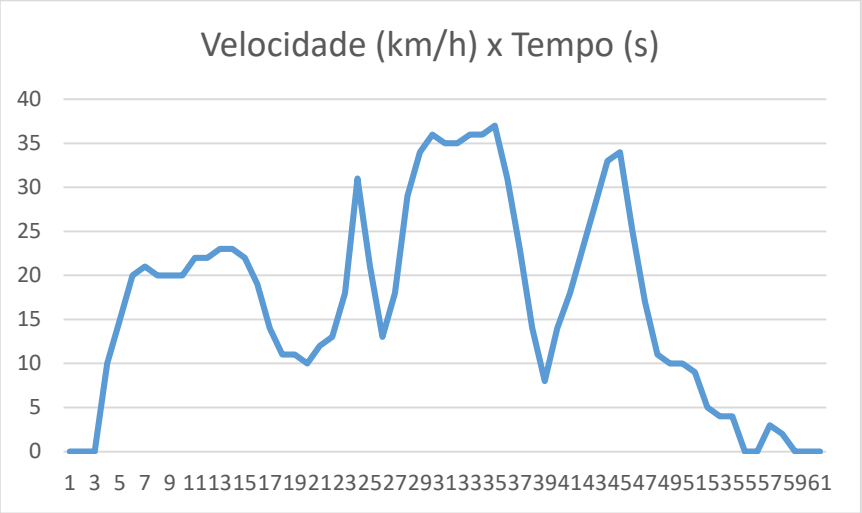


Fonte: print screen do monitor serial do Arduino

Como resultado da aquisição de dados, foi possível gerar um gráfico no software Excel apresentando a variação da velocidade ao longo do percurso (Figura 31). Tal aplicação pode ser estendida para a criação de um sistema *datalogger*, onde todos os valores coletados, referentes a diversos parâmetros automotivos, podem ser guardados em memória e acessados posteriormente, caracterizando-se assim um sistema de análise de trajetória percorrida, bem como da eficiência do veículo monitorado.

Um sistema *datalogger* contribuiria para que o motorista conseguisse visualizar com clareza e precisão o desempenho do seu veículo, por exemplo, ao analisar o consumo de combustível durante um percurso. Através do monitoramento continuo de outras variáveis, como aceleração e velocidade, o motorista também seria capaz de melhorar sua condução, resultando no aumento da segurança durante sua viagem.

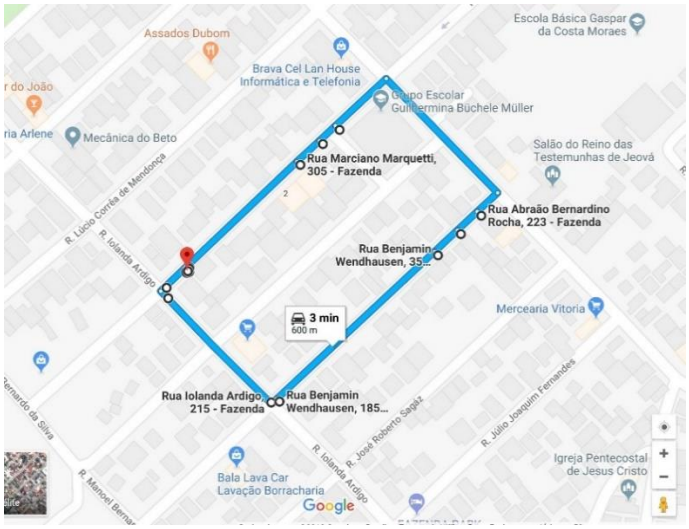
Figura 31 – Gráfico Velocidade versus Tempo



Fonte: Autoria Própria

Ao obter-se também as coordenadas geográficas do trajeto em questão, é possível associar esses dados ao sistema de mapas do Google, obtendo-se assim, uma ilustração precisa do percurso analisado. A volta no quarteirão, gerada no Google Maps através dos dados obtidos pelo sistema de monitoramento, pode ser observada na Figura 32.

Figura 32 – Mapa da trajetória percorrida pelo veículo de teste



Fonte: Print Screen da página Google Maps

5 CONCLUSÃO

A pesquisa sobre a rede de comunicação CAN, ponto inicial desse trabalho e imprescindível para a consolidação do projeto, resultou no entendimento da interação entre os módulos veiculares e as normas a que esse protocolo está sujeito. Com isso, foi possível iniciar o escopo do projeto para que atendesse a todas as normas de regulamentação. Conhecendo as diretrizes para transmissão de mensagens via Rede CAN e através da compreensão da estrutura da porta OBD-II, foi possível desenvolver um hardware embarcado que se conectasse ao sistema do veículo através de um cabo conversor ligado à porta *On-Board Diagnostics.*

Os testes foram realizados em dois veículos diferentes, de modo a certificar a eficácia do protótipo. Através da obtenção dos parâmetros dos módulos do carro, foi possível monitorar diversas variáveis de estado, como velocidade, rotação, temperatura do motor e nível de combustível. Essa supervisão pode ser direcionada para se compreender o desempenho do veículo e aprimorar a condução do motorista. Levando-se em conta que a manutenção preventiva não é um hábito recorrente dos condutores, dispor de um dispositivo que faça a leitura precisa dos módulos veiculares e detecte eventuais falhas contribui para um maior controle do veículo e segurança do motorista

Um dos objetivos propostos para esse projeto foi a implementação de uma comunicação remota. O *bluetooth* foi a transmissão sem fio escolhida para atender a esse requisito, entretanto, ele limita o usuário a estar dentro ou perto do seu carro, pois essa rede sem fio funciona dentro de um limite de proximidade. Uma mudança vantajosa no projeto seria a implementação futura de um módulo de comunicação remota para longas distâncias, permitindo assim ao usuário ter acesso às informações do veículo estando em lugares longínquos.

Para esse sistema de monitoramento, também foi criada uma interface simples e amigável para leitura dos dados, através do desenvolvimento de um aplicativo para *Android*. O programa para celulares se mostrou uma ferramenta útil principalmente para usuários leigos, por não requerer nenhum conhecimento prévio sobre a eletrônica do hardware embarcado ou o software do sistema.

Tendo em vista que todos os objetivos de projeto previamente estipulados foram atendidos, o sistema apresentado nesse trabalho se mostrou uma ferramenta simples e eficaz que contribui para a mantenabilidade do veículo e oferece ao motorista a possibilidade de monitorar o funcionamento do carro. Dispondo de informações adicionais às indicadas no painel do seu veículo, o condutor tem acesso à parâmetros mais específicos e técnicos, provenientes dos diversos sensores presentes nos módulos veiculares. Consequentemente, além de analisar a trajetória percorrida e todas as variáveis de funcionamento do veículo, o motorista provém de uma ferramenta que contribui para o aumento da sua segurança durante a viagem realizada.

Como sugestão para trabalhos futuros, além da mudança de módulo de comunicação remota, propõem-se o desenvolvimento de um sistema *datalogger*, que permita o armazenamento de dados em memória de hardware. Dessa maneira, o motorista pode acessar posteriormente esses dados e analisar a variação nos diversos parâmetros medidos no seu veículo. Esse sistema de armazenamento de dados poderia também atender ao serviço de localização, ao coletar as coordenadas geográficas de um percurso. Uma vez que essas informações estejam disponíveis no sistema, seria possível fazer um *link* com o Google Maps e trançar o trajeto percorrido em um mapa.

REFERÊNCIAS

AN228: A CAN Physical Layer Discussion. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00228a.pdf>. Acesso em: 03 jun. 2018.

ATMEGA328P-PU ATMEL 8 BIT 32K AVR MICROCONTROLLER. Disponível em: <https://protostack.com.au/shop/microcontrollers/atmega328p-pu-atmel-8-bit-32k-avr-microcontroller/>. Acesso em: 09 jun. 2018.

ATMEL: ATmega328/P - Datasheet. 2016. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>. Acesso em: 03 jun. 2018.

CAN Bus. 2016. Disponível em: <https://www.mikroe.com/blog/can-bus>. Acesso em: 09 jun. 2018.

CAN lower- and higher-layer protocols. Disponível em: <https://www.can-cia.org/can-knowledge>. Acesso em: 10 abr. 2018.

CAN message types. Disponível em: <https://dewesoft.pro/online/course/automotive-buses-can-measurement/page/can-message-types>. Acesso em: 09 jun. 2018.

CERVI, Murilo. Rede de iluminação semicondutora para aplicação automotiva. 2005. 106 p. Dissertação (Mestrado em Engenharia Elétrica)- Universidade Federal de Santa Maria, Santa Maria, RS, 2005. Disponível em: <http://repositorio.ufsm.br/bitstream/handle/1/8511/MURILOCERVI.pdf?sequence=1>. Acesso em: 10 abr. 2018.

CHARETTE, Robert. This Car Runs on Code. 2009. Disponível em: <https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code>. Acesso em: 01 mar. 2018.

CÓDIGOS de Erro OBD-II Motor/Cambio – P0001 a P3493. Disponível em: <https://www.doutorcarro.com.br/tabela-de-codigos-de-erro-dtcs-obd2-significados/>. Acesso em: 10 abr. 2018.

CZERWONKA, Mariana. Falta de manutenção triplica risco de acidentes, 2016. Disponível em: <http://portaldotransito.com.br/noticias/falta-de-manutencao-triplica-risco-de-acidentes/>. Acesso em: 01 mar. 2018.

DEPARTAMENTO NACIONAL DE TRÂNSITO. Ministério da Cultura. Frota de Veículos - 2017. Disponível em: <http://www.denatran.gov.br/index.php/estatistica/610-frota-2017>. Acesso em: 01 mar. 2018.

DIAS, Anderson. Injeção Eletrônica: UCE(ECU) – Unidade de comando eletrônico.. 2012. Disponível em: <http://www.carrosinfoco.com.br/carros/2012/07/injecao-eletronica-uceecu-unidade-de-comando-eletronico/>. Acesso em: 10 abr. 2018.

ECU (Engine Control Unit) Cars,ECM,Parts,Functioning. Disponível em: <http://aermech.com/ecu-engine-control-unit-carsecmpartsfunctioning/>. Acesso em: 10 abr. 2018.

FREEMATICS OBD-II Emulator MK2. Disponível em: <https://freematics.com/pages/products/freematics-obd-emulator-mk2/>. Acesso em: 09 jun. 2018.

GLOBALSAT GPS Module: Hardware Data Sheet Product No : EM-506. Disponível em: <https://cdn.sparkfun.com/datasheets/GPS/EM506_um.pdf>. Acesso em: 09 jun. 2018.

GPS Receiver - EM-506 (48 Channel). Disponível em: <https://www.sparkfun.com/products/12751>. Acesso em: 09 jun. 2018.

HC-05 Bluetooth serial pass-through master-slave module. Disponível em: <https://artofcircuits.com/product/hc-05-bluetooth-serial-pass-through-master-slave-module>. Acesso em: 09 jun. 2018.

HUBERT, Marco Kasdorf. O protocolo CAN como solução para aplicações distribuídas, baseadas em objetos, entre PCs e microcontroladores. 2001. 58 p. Monografia (Bacharelado em Informática)- Universidade Federal de Pelotas, Pelotas, RS, 2001. Disponível em: <http://www.eletronica.org/arq_artigos/mono-can.pdf>. Acesso em: 07 maio 2018.

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

INTERNATIONAL STANDARDIZATION ORGANIZATION. ISO 15031-3: Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics. Geneva: Iso, 2004.

LIN, Jyong et al. A Study on Remote On-Line Diagnostic System for Vehicles by Integrating the Technology of OBD, GPS, and 3G. International Journal of Humanities and Social Sciences, [S.l.], v. 3, n. 8, p. 1-7, ago. 2009. Disponível em: <https://waset.org/publications/13356/a-study-on-remote-on-line-diagnostic-system-for-vehicles-by-integrating-the-technology-of-obd-gps-and-3g>. Acesso em: 10 abr. 2018.

MCP2551: High-Speed CAN Transceiver. Disponível em: <http://ww1.microchip.com/downloads/en/devicedoc/21667d.pdf>. Acesso em: 03 jun. 2018.

MCP2515: Stand-Alone CAN controller with SPI™ Interface. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/21801d.pdf>. Acesso em: 03 jun. 2018.

MACHADO, António Sérgio Leite; OLIVEIRA, Bruno Rafael Resende. O Sistema OBD (On-Board Diagnosis). 2008. 8 p. Artigo (Mestrado em Automação e Sistemas)- Instituto Superior de Engenharia do Porto, [S.l.], 2008. Disponível em: <http://ave.dee.isep.ipp.pt/~mjf/act_lect/SIAUT/Trabalhos%202007-08/Trabalhos/SIAUT_OBD.pdf>. Acesso em: 10 abr. 2018.

MODELO OSI. 2013. Disponível em: <http://mitodasredes.blogspot.com/2013/08/modelo-osi.html>. Acesso em: 09 jun. 2018.

O COMPUTADOR de bordo evolui. ESTADÃO, [S.l.], 03 dez. 2013. Jornal do Carro, p. 1. Disponível em: <http://jornaldocarro.estadao.com.br/carros/o-computador-de-bordo-evolui/>. Acesso em: 01 mar. 2018.

O QUE é o módulo ECM em um veículo com injeção eletrônica?. Disponível em: <https://www.carrodegaragem.com/que-modulo-ecm-veiculo-com-injecao-eletronica/>. Acesso em: 10 abr. 2018.

OBD II Data Link Connector. Disponível em: <https://www.freeasestudyguides.com/electrical-troubleshooting-data-link-connector.html>. Acesso em: 10 abr. 2018.

OBD MODES AND PIDS. Disponível em: <https://www.outilsobdfacile.com/obd-mode-pid.php>. Acesso em: 10 abr. 2018.

OBD2 Cable. Disponível em: <https://obd2allinone.com/products/obd2cable.asp>. Acesso em: 13 jun. 2018.

ONBOARD Diagnostics (OBD & non OBD protocols). Disponível em: <https://www.munic.io/documentations/>. Acesso em: 10 abr. 2018.

PINHEIRO, José Mauricio Santos. OSI: Um Modelo de Referência. 2008. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_osi_um_modelo_de_referencia.php>. Acesso em: 07 maio 2018.

PINTO, Benedito G. Miglio; FILHO, Antônio Bento; AMARAL, Paulo Faria Santos. Computador de Bordo para Locomotivas. MU 7001432-9 U2. 11 de fev. de 1992. [S.l.], p. 13 Disponível em: <https://gru.inpi.gov.br/>. Acesso em: 01 mar. 2018.

PINTO LIMA, Bianca. Brasileiro gasta por ano 40% do valor do carro com manutenção e despesas, 2013. Disponível em: http://economia.estadao.com.br/noticias/geral,brasileiro-gasta-por-ano-40-do-valor-do-carro-com-manutencao-e-despesas,145264e. Acesso em: 01 mar. 2018.

SACCO, Francesco. Comunicação SPI – Parte 1. 2014. Disponível em: <https://www.embarcados.com.br/spi-parte-1/>. Acesso em: 09 jun. 2018.

SCOTT BROWN, David. Your Car, Your Computer: ECUs and the Controller Area Network. 2017. Disponível em: <https://www.techopedia.com/your-car-your-computer-ecus-and-the-controller-area-network/2/32218>. Acesso em: 10 abr. 2018.

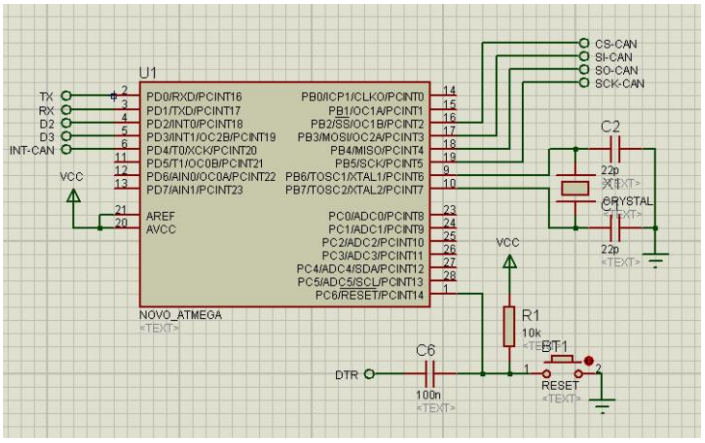
SERIAL Peripheral Interface (SPI). Disponível em: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>. Acesso em: 09 jun. 2018.

SOCIETY OF AUTOMOTIVE AND AEROSPACIAL ENGINEERS. SAE J1979: E/E Diagnostic Test Modes. Detroit: Sae, 2006.

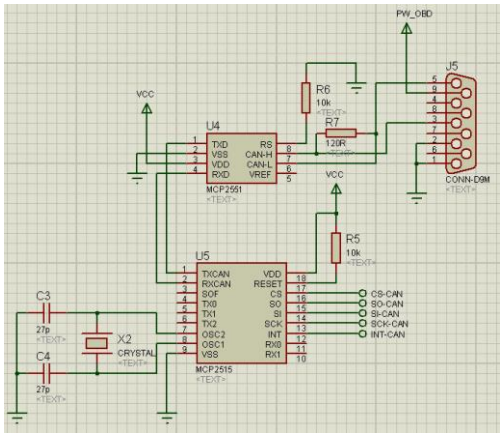
TELEMETRIA AUTOMOTIVA: A HISTÓRIA DO SISTEMA OBD E SUA EVOLUÇÃO. 2016. Disponível em: <http://carrorama.net/a-historia-do-obd-e-sua-evolucao/>. Acesso em: 01 mar. 2018.

APÊNDICE A

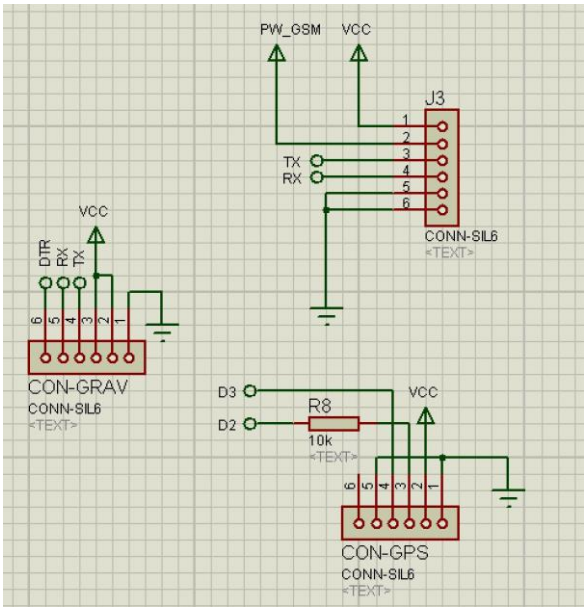
Figura 33 – Diagramas de Circuito



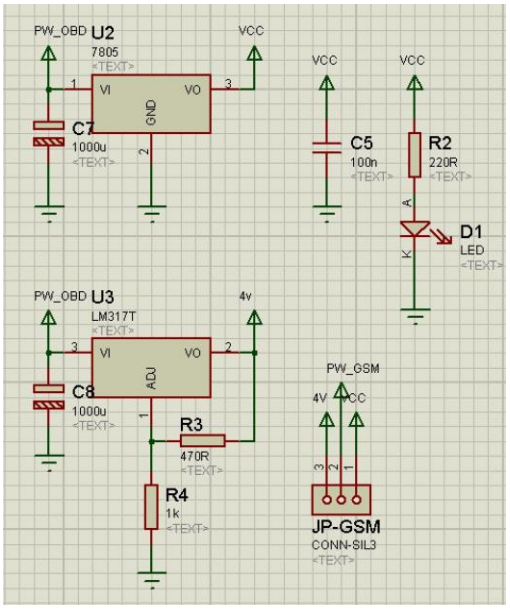
(a) Circuito ATMEGA328p



(b) Circuito CAN



(c) Circuito Conectores

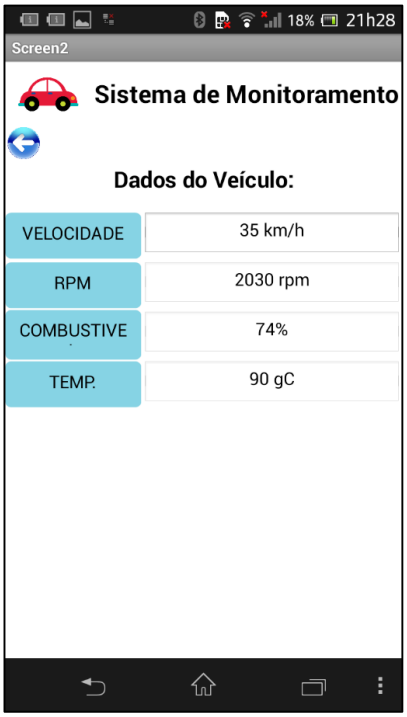


(d) Circuito de Alimentação

Fonte: Autoria Própria

APÊNDICE B

Figura 34 – Telas do Aplicativo



(a) Tela de Dados

(b) Tela de Código de Falhas

(c) Tela do GPS

Fonte: Autoria Própria