



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E INFORMÁTICA

Sistema de telemetria para veículo off-road baseado em protocolo CAN

Matheus Torres Rocha

Campina Grande, PB

Julho de 2019

Matheus Torres Rocha

Sistema de telemetria para veículo off-road baseado em protocolo CAN

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Área de Concentração: Instrumentação Eletrônica, Sistemas Embarcados, Internet das Coisas e Sistemas Supervisórios.

Orientador: José Sérgio da Rocha Neto

Campina Grande, PB

Julho de 2019

Matheus Torres Rocha

Sistema de telemetria para veículo off-road baseado em protocolo CAN

Aprovado em ____/____/____

Prof. Jaidilson Jó da Silva

Universidade Federal de Campina Grande

Avaliador

Prof. José Sérgio da Rocha Neto

Universidade Federal de Campina Grande

Orientador

Aos meus pais que nunca desistiram de mim.

Agradecimentos

Agradeço primeiramente a Deus por ser o meu porto seguro e me dar forças para continuar cada dia em meio a tantas dificuldades.

Aos meus pais Dayse e Antônio, à minha irmã Thays, à minha avó Vanilde, à minha namorada Klayne, pelo suporte e incentivo para eu nunca desistir do meu sonho. Tenho a certeza que não teria chegado até aqui sem vocês.

Ao Prof. Péricles por ser o primeiro professor a acreditar no meu potencial e me dando a oportunidade de estudar tudo que a universidade não pode me oferecer, ao Prof. Rafael pela amizade, apoio e ajuda e ao amigo Charles por orientar em diversas situações. Prof. José Sérgio e Prof. Jaidilson por me dar a honra de tê-los como orientador e avaliador, respectivamente, de trabalho de conclusão de curso, direcionando o caminho certo a ser seguido.

À toda equipe ParahyBaja, em especial, Lucas, Douglas e Alberi pela oportunidade de realizar meu projeto de conclusão de curso no seu projeto de extensão.

Aos amigos da universidade, em especial, Yago, Junior, Luquinhas e Ravi pela ajuda, incentivo, leites, brincadeiras e trabalhos desenvolvidos durante todos os anos da minha graduação.

Aos meus amigos da época do colégio, em especial, Rennan, Lucas, Giusepe, Gabriel, Maercio, Jorge, Samuel, Emmanuel e João por sempre estarem ao meu lado nos momentos mais difíceis da minha caminhada, me trazendo felicidade e suporte.

Por fim, agradeço à minha maior fonte de inspiração, minha avó, mais conhecida como Dona Isabel a pessoa mais iluminada que tive a honra de conhecer. Que mesmo não estando presente, fisicamente, tenho certeza que esteve ao meu lado em todos os momentos da minha vida.

*"A única coisa que ninguém pode tomar de você é o **conhecimento**."*

Resumo

O nível de tecnologia presente nos carros é um fator determinante quando se trata de conforto, praticidade e desempenho. Assim, a necessidade da presença de telemetria em carros para competição Baja SAE BRASIL é indiscutível. Este trabalho de conclusão de curso tem como objetivo o estudo para desenvolver, elaborar, testar e implementar um sistema de telemetria embarcado utilizando rede de comunicação via protocolo CAN, microcontroladores ST e Computação em Nuvem (Cloud Computing) de modo a analisar, salvar e tratar velocidade, rotação por minuto e nível de combustível do carro desenvolvido pela equipe ParahyBaja.

Palavras chave: CAN, ParahyBaja, SAE BRASIL, Telemetria.

Abstract

The level of technology present in cars is a determining factor when it comes to comfort, practicality and performance. As a result, the necessity of telemetry in cars for competition Baja SAE BRAZIL is unquestionable. The purpose of this paperwork is the study to develop, test and implement an embedded telemetry system using CAN protocol, ST microcontrollers and Cloud Computing to analyze, save and treat speed, rotation per minute and fuel level of the car developed by ParahyBaja team.

Keywords: CAN, ParahyBaha, SAE BRASIL, Telemetry.

Lista de Figuras

1.1	Diagrama dos sensores e componentes para telemetria.	2
2.1	Diagrama de blocos da conexão genérica de ECU.	5
2.2	Diagrama com as camadas utilizadas pelo protocolo CAN no modelo OSI.	6
2.3	Topologia de barramento de rede CAN de acordo com a norma ISO 11898-2.	7
2.4	Nível de barramento de acordo com a norma ISO 11898-2.	8
2.5	Gráfico da taxa de transmissão versus a distancia máxima entre os nós.	8
2.6	Esquema de uma mensagem de dados.	9
2.7	Esquema de uma mensagem de erro.	10
2.8	Exemplo de arbitração na rede CAN.	11
2.9	Diagrama de blocos de um nó CAN.	11
2.10	Diagrama de blocos de um nó CAN com a arquitetura Stand-Alone.	12
2.11	Diagrama de blocos de um nó CAN com a arquitetura Integrada.	12
2.12	Diagrama de blocos de um nó CAN com a arquitetura <i>Single-Chip</i>	13
2.13	Representação do funcionamento interno do sensor capacitivo.	14
2.14	Exemplificação do campo eletromagnético de um sensor capacitivo.	14
2.15	Representação da estrutura interna do sensor indutivo.	15
3.1	Representação esquemática da vista lateral do Baja ParahyBaja e locais propostos para ECU.	16

3.2	Circuito implementado na utilização do <i>transiver</i> MCP2551.	17
3.3	Representação esquemática da ECU1.	18
3.4	Diagrama geral em maquina de estados da ECU1.	18
3.5	Fotografia do sensor indutivo LJ12A3-4-Z/BX.	19
3.6	Circuito implementado na utilização do sensor indutivo.	19
3.7	Fotografia da bobina acoplada ao eixo do motor.	20
3.8	Curvas da saída do sensor de rpm puro (2) e saída do sensor de rpm tratado (1).	21
3.9	Circuito implementado na utilização do sensor de RPM.	21
3.10	Fotografia do sensor capacitivo LJC18A3-B-Z/BY.	22
3.11	Circuito implementado na utilização do sensor de combustível.	23
3.12	Representação esquemática da ECU2.	24
3.13	Diagrama geral em maquina de estados da ECU2.	24
3.14	Representação esquemática da ECU3.	25
3.15	Exemplo do painel frontal com LCD.	25
3.16	Diagrama do recebimento de dados <i>WEB</i>	26
3.17	Escopo da mensagem <i>JSON</i> dentro do <i>MongoDB</i>	27
3.18	Fotografia do cabeçalho de dados únicos na plataforma <i>WEB</i>	28
3.19	Gráficos de rpm em função do tempo na plataforma <i>WEB</i>	28
3.20	Gráficos de velocidade em função do tempo na plataforma <i>WEB</i>	29
4.1	Fotografia do sistema implementado em <i>protoboard</i>	31
4.2	Fotografia da montagem do motor e sensores em bancada.	32
4.3	Circuito simulador de RPM.	33
4.4	Fotografia da placa final ECU1.	34

4.5	Fotografia das placas finais ECU3 e ECU2.	34
4.6	Fotografia do painel e sensores localizados no Baja.	35
4.7	Plataforma <i>WEB</i>	36

Lista de Abreviaturas e Siglas

CAN - *Controller Area Network*

CI - Circuito Integrado

ECU - *Electronic Controller Unit*

GND - Ground

GPS - *Global Position System*

I2C - *Inter-Integrated Circuit*

RPM - Rotações por Minuto

SAE - Sociedade de Engenheiros da Mobilidade

Conteúdo

1	Introdução	1
1.1	Objetivo	2
1.2	Organização do texto	3
2	Revisão Bibliográfica	4
2.1	Telemetria	4
2.2	Competição Baja SAE BRASIL	5
2.3	Protocolo de Comunicação CAN	6
2.3.1	Camada Física	7
2.3.2	Camada de Enlace	8
2.3.3	Controle de Acesso ao Meio	10
2.3.4	Nó CAN	11
2.4	Sensores	13
2.4.1	Sensor Capacitivo	13
2.4.2	Sensor Indutivo	14
3	Sistema Implementado	16
3.1	ECU1	17

3.1.1	Sensor de Velocidade	18
3.1.2	Sensor de RPM	20
3.1.3	Sensor de Combustível	22
3.2	ECU2	23
3.3	ECU3	24
3.4	Interface Humano-Máquina	26
3.4.1	Recebimento de Dados	26
3.4.2	Tratamento de Dados	26
3.4.3	Armazenamento em Banco de Dados	27
3.4.4	Plataforma <i>WEB</i>	27
4	Implementação	30
4.1	Implementação em Bancada	30
4.2	Confecção das Placas	33
4.3	Implementação no Baja	34
5	Conclusões	37
6	Bibliografia	38

Capítulo 1

Introdução

No decorrer dos avanços tecnológicos, a eletrônica automotiva vem se tornando cada vez mais presente no dia a dia dos indivíduos, tornando-se essencial para o conforto e a segurança do condutor, assim como para o funcionamento e a performance de veículos.

Ao logo do desenvolvimento tecnológico dispositivos (sensores e atuadores) foram desenvolvidos para auxiliar na automação de máquinas, equipamentos e processos. Os sensores, mais especificamente, tem um papel incontestável para o desenvolvimento automotivo. Possibilitando a verificação e obtenção de dados para processos, até então, indetectáveis, como por exemplo velocidade, rotação por minuto, entre outros [3].

Dada a importância da eletrônica automotiva, microcontroladores são projetados e programados para serem embarcados em linhas automotivas chamados de unidade de controle eletrônico (*Electronic Control Unit* - ECU). Suas aplicações são as mais diversas possíveis, possuindo uma capacidade de processamento.

Existem diversas ECU distribuídas pelo carro, por exemplo, no controle de nível de combustível por meio de sensores resistivos, no controle de acionamento de *airbag* através de sensores de vibração e no velocímetro através de sensores indutivos.

Entre as ECU tem-se então a necessidade de uma comunicação precisa e segura, de forma a garantir confiabilidade e qualidade do sistema. Para essa necessidade, o sistema de barramento de dados CAN é utilizado como padrão para a aplicação veicular [4 - 5], estando presente em todos os carros desenvolvidos atualmente.

Entretanto, os veículos *off-road* Baja, mais precisamente os veículos utilizados para competição Baja SAE BRASIL, possuem uma deficiência quando se trata de eletrônica digital automotiva por se tratar de carros que necessitam, a princípio, apenas de botões de emergência e luzes para seu funcionamento.

Em uma competição com foco em desenvolvimento tecnológico, a implementação de ECU se torna ne-

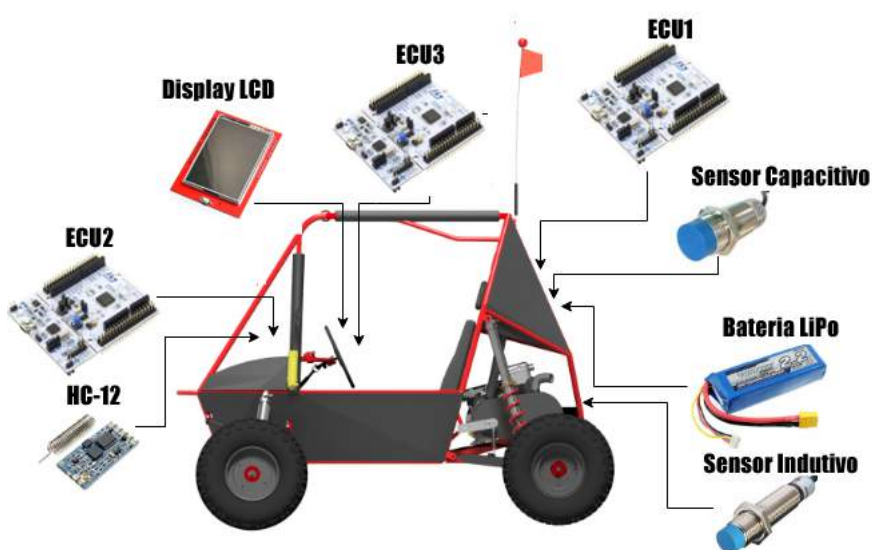
cessária, a fim de validar diversas funcionalidade do sistema, alcançar objetivos, conforto do piloto e melhorar performance do veículo, tornando-se necessária a criação de um sistema de telemetria.

Telemetria consiste na aquisição e envio de dados para um acumulador visando uma posterior análise [3]. Possibilitando uma possível verificação de dados remotamente, por exemplo em painéis automotivos, ou nos dias atuais a verificação *online* em meio a rede de internet.

1.1 Objetivo

Neste trabalho de conclusão de curso tem-se como objetivo projetar um sistema de eletrônica embarcada responsável pela aquisição, transmissão e exibição de dados relevantes para a equipe e para o piloto em tempo real, tais como: rotação do motor, nível de combustível e velocidade. Uma possível solução do sistema está apresentada na Figura 1.1:

Figura 1.1: Diagrama dos sensores e componentes para telemetria.



Fonte: Próprio Autor.

A alocação de sensores, seguida da aquisição dos dados para as ECU responsáveis pela comunicação com o painel e com o computador central, será deste modo, um sistema de telemetria para o carro da equipe ParahyBaja. Os objetivos específicos são:

- Selecionar tipos de sensores e ECU para aplicações;
- Desenvolver circuitos equivalentes para as aplicações;

- Desenvolver a programação do sistema;
- Executar montagem do sistema em placa de ensaio;
- Planejar e executar testes para validação do sistema desenvolvido;
- Desenvolver módulos definitivos para o projeto;
- Executar testes e implementação do sistema desenvolvido no veículo.

1.2 Organização do texto

Neste capítulo foi apresentado a introdução e os meus objetivos para esse trabalho de conclusão de curso. No capítulo 2 será apresentada uma abordagem teórica aos tópicos relevantes do trabalho, para uma melhor compreensão das atividades desenvolvidas. No capítulo 3, o sistema implementado. No capítulo 4, implementações realizadas. Por fim, no capítulo 5 são analisados os resultados com propostas para estudos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo apresenta-se os principais conceitos e fundamentos envolvidos no sistema proposto. São aqui descritos os conceitos de telemetria, competição Baja SAE BRASIL, protocolo de comunicação CAN e sensores utilizados.

2.1 Telemetria

A telemetria surgiu no ano de 1845 com o desenvolvimento do primeiro sistema de transmissão de dados entre o Winter Palace e o quartel da armada russa [3]. Em 1874 foi desenvolvida uma rede de sensores atmosféricos em Mont Blanc, França, com transmissão de dados em tempo real para Paris. Mais tarde, em 1906 foram montadas séries de estações sísmicas na Rússia, com transmissão de dados para o Pulkovo Observatory [3].

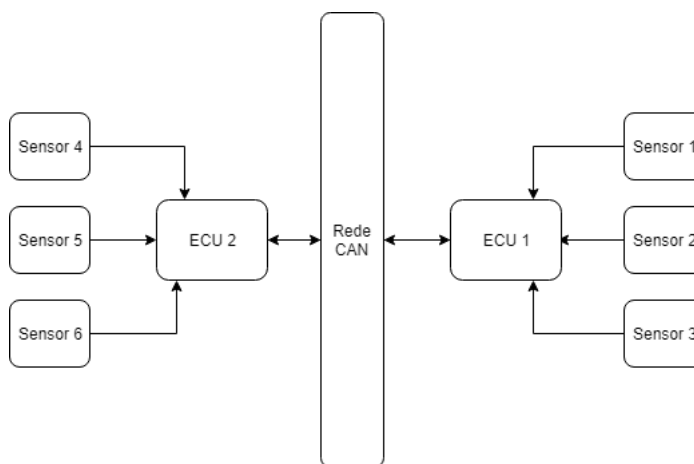
A telemetria automotiva é amplamente utilizada em veículos de uso comercial e máquinas agrícolas [3]. O sistema de telemetria automotiva é responsável pelo transporte e pelo armazenamento de dados do carro, como a velocidade, a rotação do motor, o nível de combustível, a temperatura do motor, etc. Essa aquisição de dados é realizada por meio de sensores interligados, dos mais diversos tipos e maneiras.

A arquitetura interna do *hardware* composto nos veículos automotivos é distribuída em ECU responsáveis, em grande maioria, por uma aquisição de dados direcionado a um único sensor, atuador ou ação do veículo. Seu nível de complexidade e seu numero de ECU, disponíveis e conectadas entre si, é diretamente proporcional ao valor do veículo [1].

As ECUs são conectadas entre si através de um único barramento para troca de informações e concatenação do sistema como um todo. Assim, a diminuição do número de cabos necessários para conexão das mesmas, maior

robustez levando em consideração que erros em uma ECU podem não afetar outras, podem ser observados como pontos positivos para a implementação de ECU, em contrapartida o custo, a exigência de uma implementação de protocolo de comunicação entre ECU são pontos negativos do seu desenvolvimento. Um exemplo de conexão entre ECU é apresentada na Figura 2.1

Figura 2.1: Diagrama de blocos da conexão genérica de ECU.



Fonte: Próprio Autor.

Diante dos avanços tecnológicos os bancos de dados, os tratamento de dados e as amostragem de resultados estão sendo cada vez mais direcionado a sistemas em nuvem. Estes podendo ser acessados de qualquer parte do mundo, possuindo como pré-requisito apenas o acesso a internet.

2.2 Competição Baja SAE BRASIL

O programa Baja SAE foi criado na Universidade da Carolina do Sul, Estados Unidos, sob a direção do Dr. John F. Stevens, tendo sua primeira edição em 1976. O ano de 1991 marcou o início das atividades da SAE BRASIL, que, em 1994, lançava o Programa Baja SAE BRASIL. No ano seguinte, em 1995, seria realizada a primeira competição nacional, na pista Guido Caloi, bairro do Ibirapuera, cidade de São Paulo [10].

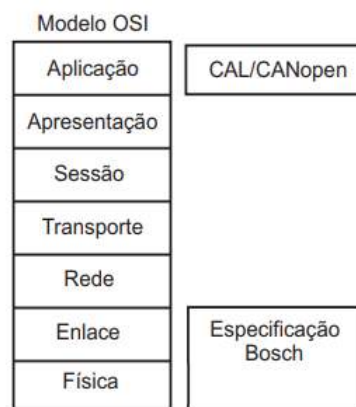
A competição SAE Baja Brasil tem por objetivo de conectar estudantes de diversas universidades para disputar uma competição que tem o intuito de projetar, desenvolver e criar carros tipo baja, capazes de superar diversas condições, como por exemplo, lama, chuva, trepidação, etc. O veículo deve ser capaz de acomodar um condutor desde 1,45 m de altura, com peso de 42 kg até um condutor de 1,90 m de altura, com peso de 109 kg [10].

2.3 Protocolo de Comunicação CAN

Decorrente do alto numero de dispositivos de controles, sensores e atuadores presentes nos automóveis, o numero de cabos para conexões tornou-se um grande problema para a manutenção, criação e detecção de falhas muito complexas nos automóveis. Por essa razão, o protocolo de comunicação CAN (*Controller Area Network*) foi desenvolvido pela empresa Robert Bosch GmbH na década de 80 [5].

Esse protocolo é definido pela resolução ISO 11898 e trás consigo um modelo de referencia tipo OSI (*Open System Interconnection*), possuindo três camadas: a camada física, a camada de enlace de dados e a camada de aplicação. A camada de aplicação pode ser desenvolvida a critério do desenvolvedor como por exemplo, o *CANopen* [4]. Na Figura 2.2 apresenta-se as camadas que são utilizadas dentro do modelo OSI pelo protocolo CAN.

Figura 2.2: Diagrama com as camadas utilizadas pelo protocolo CAN no modelo OSI.



Fonte: Adaptado de [6].

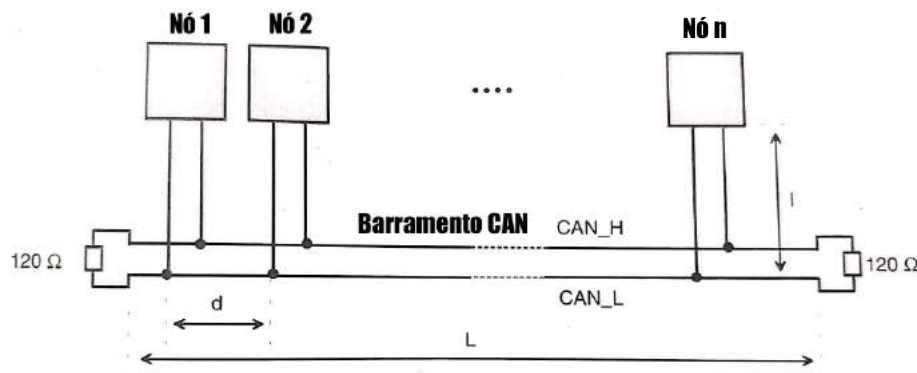
Sua camada de enlace e física não podem ser modificadas e são diretamente implementadas no microchip de controle, já a camada de aplicação, como elencado anteriormente, fica a critério do desenvolvedor.

O protocolo CAN tem como principais vantagens: a robustez, a redução de ruídos eletromagnéticos e a diminuição da complexidade de conexões entre sensores e atuadores [7]. Também, a sua especificação é dividida em dois padrões para a versão CAN 2.0, identificadas como 2.0A e 2.0B, possuindo diferença entre o tamanho do identificador de mensagem com 11 bits e 29 bits respectivamente. Entretanto, os dois padrões são compatíveis entre si [4].

2.3.1 Camada Física

A transmissão de dados do protocolo CAN, é composto por um par de fios trançados possuindo uma resistência de 120 Ohms nas pontas do seu barramento a fim de evitar reflexões do sinal e minimizar os efeitos eletromagnéticos. Pode-se observar a topologia de barramento da rede CAN na Figura 2.3.

Figura 2.3: Topologia de barramento de rede CAN de acordo com a norma ISO 11898-2.



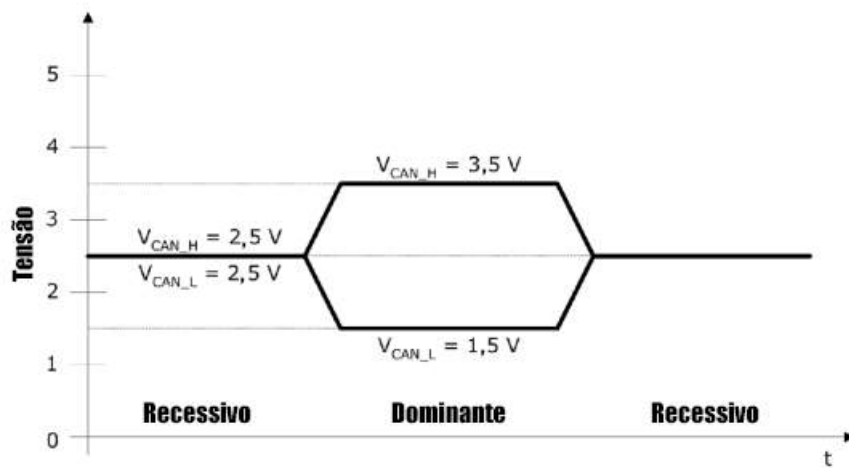
Fonte: Adaptado de [5].

Composto por dois fios, CAN_H (CAN HIGH) e CAN_L (CAN LOW), possui seu nível lógico medido decorrente da diferença de potencial entre os fios.

$$V_{dif} = V_{CAN_H} - V_{CAN_L} \quad (2.1)$$

Assim, quando seu nível lógico possui um valor $V_{dif} < 0,5$ é considerado recessivo e consequentemente um valor 0, quando $V_{dif} > 0,9$ é considerado dominante e possui valor 1. O gráfico da relação entre a diferença de potencial dos fios pode ser vista na Figura 2.4

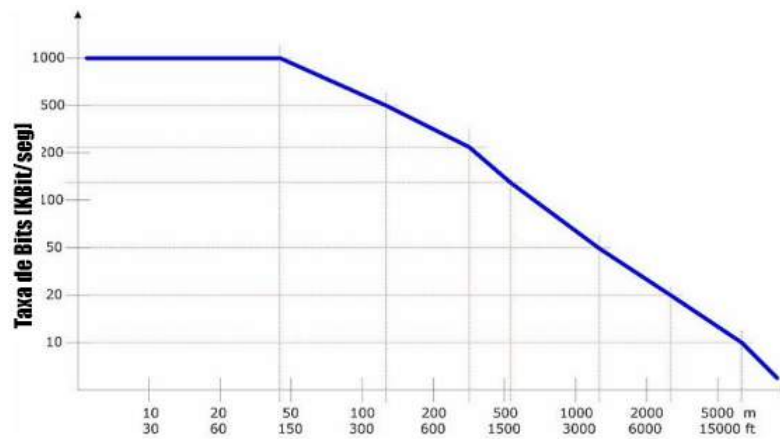
Figura 2.4: Nível de barramento de acordo com a norma ISO 11898-2.



Fonte: Adaptado de [5].

A rede CAN apresenta limitações para seu funcionamento, possuindo uma relação máxima entre a taxa de transferência de bits versus a distancia máxima permitida para o seu funcionamento (Figura 2.5).

Figura 2.5: Gráfico da taxa de transmissão versus a distancia máxima entre os nós.



Fonte: Adaptado de [5].

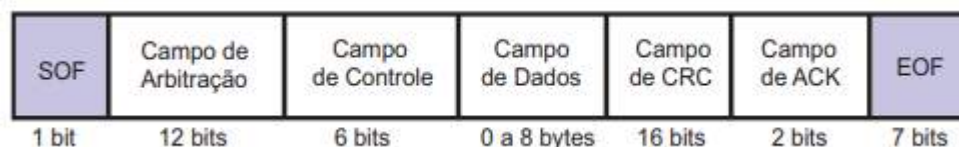
2.3.2 Camada de Enlace

Os dados transmitidos através da rede CAN possuem quatro tipos distintos de mensagem:

Mensagem de Dados

A estrutura da mensagem de dados, apresentada na Figura 2.6, possui espaços especificamente alocados para determinado dado da mensagem, são eles: SOF (*Start of Frame*), arbitração (*Arbitration Field*), controle (*Control Field*), campo de dados (*Data Field*), CRC (*Cyclic Redundancy Check*), ACK (*Acknowledgement*) e EOF (*End of Frame*) [6].

Figura 2.6: Esquema de uma mensagem de dados.



Fonte: [6].

SOF - Responsável por marcar o início da mensagem, apresentado por um único bit dominante.

Arbitração - O campo de arbitração possui dois tamanhos, 11 *bits* para a versão 2.0 A e 29 *bits* para versão 2.0 B, ele é responsável por enviar o nível de prioridade da mensagem, seguindo a regra que quando menor for o valor maior é o seu nível de prioridade. Também, envia em seu último bit (*Remote Transmit Request* - RTR) a responsabilidade de afirmar sua requisição de retransmissão de dados.

Controle - Seu primeiro bit (IDE) é responsável por identificar a versão da rede CAN e, em seus *bits* restantes, possui a tarefa de informar a quantidade de *bytes* que serão transmitidas na mensagem.

Campo de dados - Reservado para transmissão da mensagem, possuindo um tamanho máximo de 8 *bytes*.

CRC - Responsável por conter o *checksum* da mensagem para uma verificação de possíveis erros na transmissão de dados.

ACK - É responsável por reconhecer uma mensagem sem erro, o transmissor verifica através da resposta do receptor, caso essa resposta esteja correta é enviada bit dominante e caso contrario é enviado um bit recessivo.

EOF - Responsável por finalizar a mensagem.

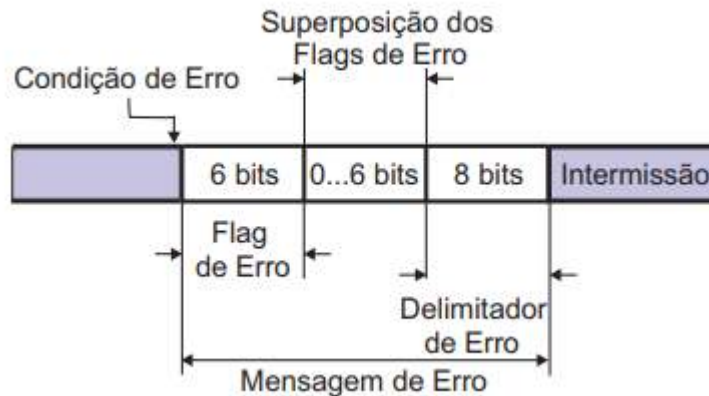
Mensagem de requisição remota de dados

Utilizado para solicitar dados, a estrutura da mensagem de requisição remota de dados possui os mesmos campos da mensagem de dados, entretanto não possui o campo de dados.

Mensagem de erro

A estrutura da mensagem de erro pode ser verificada na Figura 2.7.

Figura 2.7: Esquema de uma mensagem de erro.



Fonte: [6].

Os primeiros 6 *bits* da mensagem correspondem à *flag* de erro que possui dois estados: o ativo, que identifica erro em nós ativos e o passivo, que identifica erro em nós passivos. O delimitador de *flags* de erro é composto por oito *bits* recessivos.

Mensagem de sobrecarga

A mensagem de sobrecarga é igual à estrutura utilizada para a mensagem de erro, mas é utilizada para realizar atrasos maiores na transmissão de pacotes.

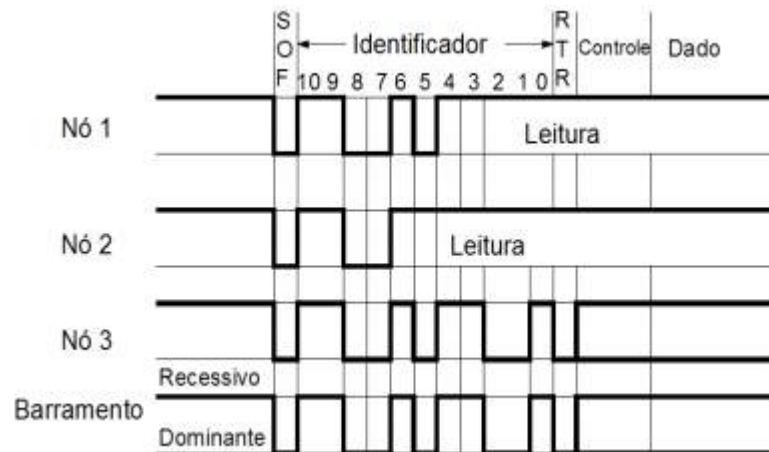
2.3.3 Controle de Acesso ao Meio

Para controlar o acesso ao meio a CAN utiliza o método CSMA/CA (*Carrier-Sense Multiple Access with Collision Avoidance*), o método garante que durante a fase de arbitração, ou seja, durante a disputa pelo acesso ao barramento, apenas o nó que tiver a mensagem com maior prioridade permanecerá transmitindo a mensagem após o final do campo de arbitração [4].

O método funciona da seguinte maneira: quando o nó envia o *bit* dominante, ele continua no processo, e envia o próximo *bit* da sua mensagem de arbitração, caso um nó envie um *bit* recessivo enquanto outro nó envia o *bit*

dominante esse nó ganhará a disputa para leitura de dados. Por exemplo, pode ser observado no bit 5 (Nó 2) e no bit 2 (Nó 1) do identificador, na Figura 2.8, vencendo o nó 3, visto que ele contém a mensagem com maior prioridade [6].

Figura 2.8: Exemplo de arbitragem na rede CAN.

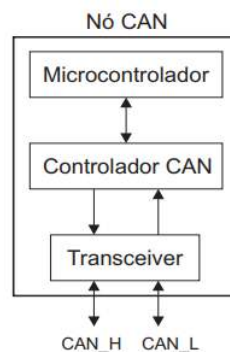


Fonte: [6].

2.3.4 Nó CAN

Para implementar um nó CAN (Figura 2.9) são necessários três componentes básicos: um *transceiver*, um controlador CAN e um microcontrolador [6]. Cada componente possui uma ação primordial para o funcionamento da rede CAN, o microcontrolador é responsável por configurar os periféricos (sensores e atuadores) instalados em seu sistema, o controlador CAN é responsável por disponibilizar as funções para uma possível implementação e o *transceiver* é responsável por traduzir os dados que chegam através do barramento CAN.

Figura 2.9: Diagrama de blocos de um nó CAN.



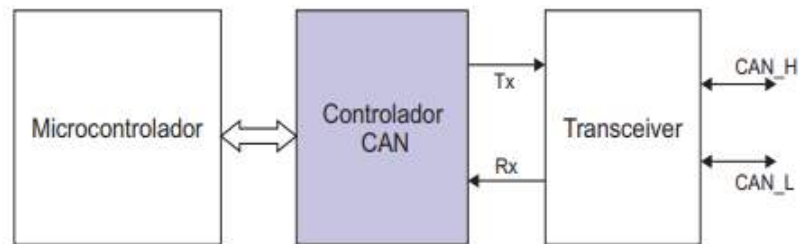
Fonte: [6].

Existem diversos tipos de microcontroladores no mercado, ao que se trata de protocolo CAN, eles podem ser classificados em três tipos de microcontroladores [7]:

- *Stand-Alone*

Na arquitetura *Stand-Alone* (Figura 2.10) é separada do controlador CAN e do *transceiver*, necessitando acoplar um controlador e um *transceiver* em sua implementação.

Figura 2.10: Diagrama de blocos de um nó CAN com a arquitetura Stand-Alone.

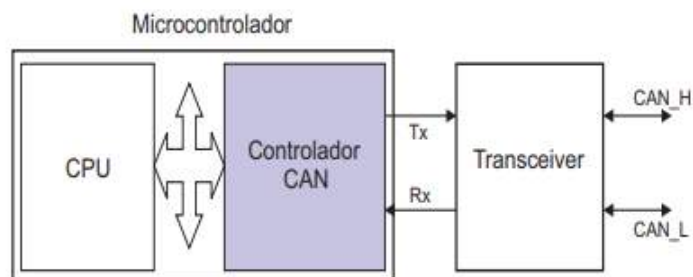


Fonte: [6].

- Integrada;

Na arquitetura integrada (Figura 2.11) temos o microcontrolador e o controlador integrados em um único chip e apenas o *transceiver* separado da implementação.

Figura 2.11: Diagrama de blocos de um nó CAN com a arquitetura Integrada.

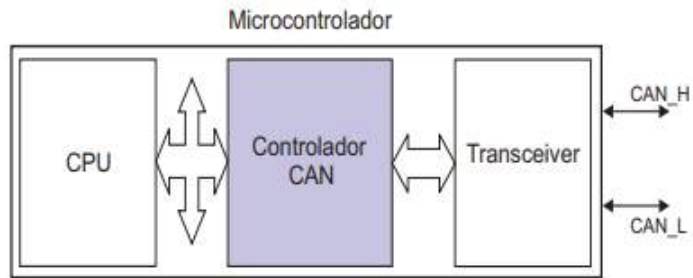


Fonte: [6].

- *Single-Chip*;

Na arquitetura *Single-Chip* (Figura 2.12) temos o microcontrolador, controlador e *transceiver* unidos em um único chip.

Figura 2.12: Diagrama de blocos de um nó CAN com a arquitetura *Single-Chip*.



Fonte: [6].

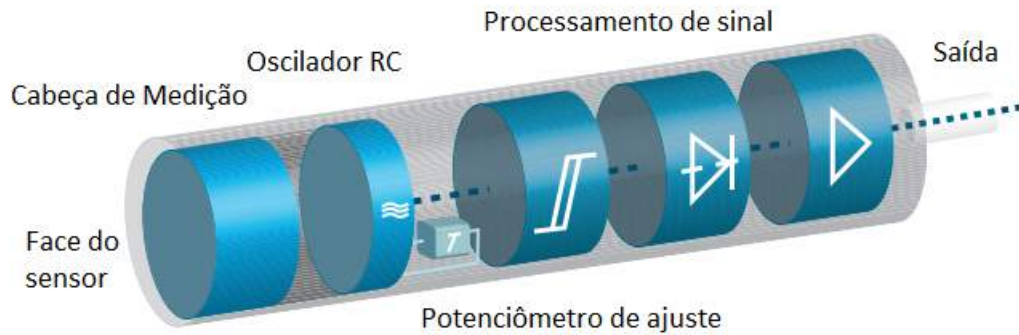
2.4 Sensores

Existem diversos tipos de sensores, como por exemplo os sensores capacitivos e os sensores indutivos.

2.4.1 Sensor Capacitivo

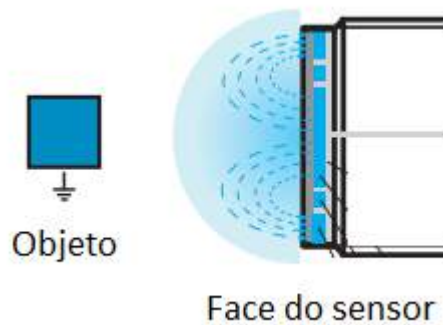
O sensor capacitivo (Figura 2.13) em seu princípio operacional toma como parâmetro a capacitância. Uma chave de proximidade capacitiva funciona com um circuito oscilante de alta frequência que cria um campo eletromagnético na superfície ativa do sensor por meio de um capacitor. Quando uma substância sólida ou fluida se aproxima deste campo ocorre um desequilíbrio de capacitância e leva a uma variação da amplificação no circuito oscilante. Se esta amplificação exceder um valor limite, um sinal de comutação é gerado. Uma exemplificação de suas ondas eletromagnéticas podem ser vistas na Figura 2.14 [8].

Figura 2.13: Representação do funcionamento interno do sensor capacitivo.



Fonte: Adaptado de [8].

Figura 2.14: Exemplificação do campo eletromagnético de um sensor capacitivo.



Fonte: Adaptado de [8].

O sensor pode ser pré-fabricado em diversas situações de funcionamento, lógica positiva (PNP) ou lógica negativa (NPN), ou seja, no caso PNP, o sensor indicará um nível lógico alto ao detectar uma variação no seu campo, já no caso NPN, o sensor indicará um nível lógico baixo para as mesmas condições.

2.4.2 Sensor Indutivo

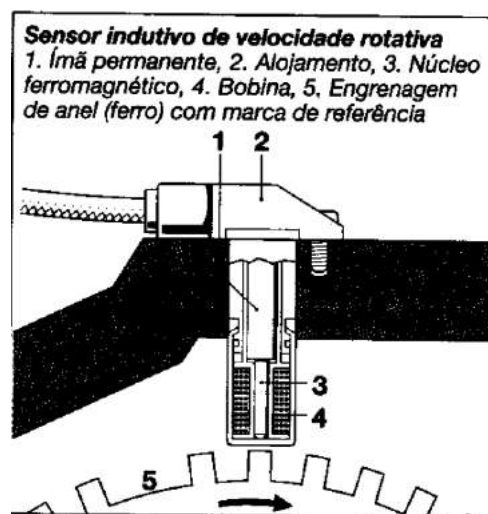
Assim como o sensor capacitivo, o funcionamento do sensor indutivo (Figura 2.15) em seu princípio operacional toma como parâmetro a indutância.

Seu funcionamento consiste em um ímã de barra, com um pino ferromagnético, sustentando uma bobina de

indução com dois terminais e quando uma engrenagem de anel ferromagnético (ou um rotor de projeto semelhante) passa por este sensor, gera uma tensão na bobina, tensão esta diretamente proporcional a variação periódica do fluxo magnético [9].

Esse tipo de sensor, possui o mesmo funcionamento de lógica do sensor capacitivo, ou seja, ele pode ser fabricado como PNP ou como NPN.

Figura 2.15: Representação da estrutura interna do sensor indutivo.



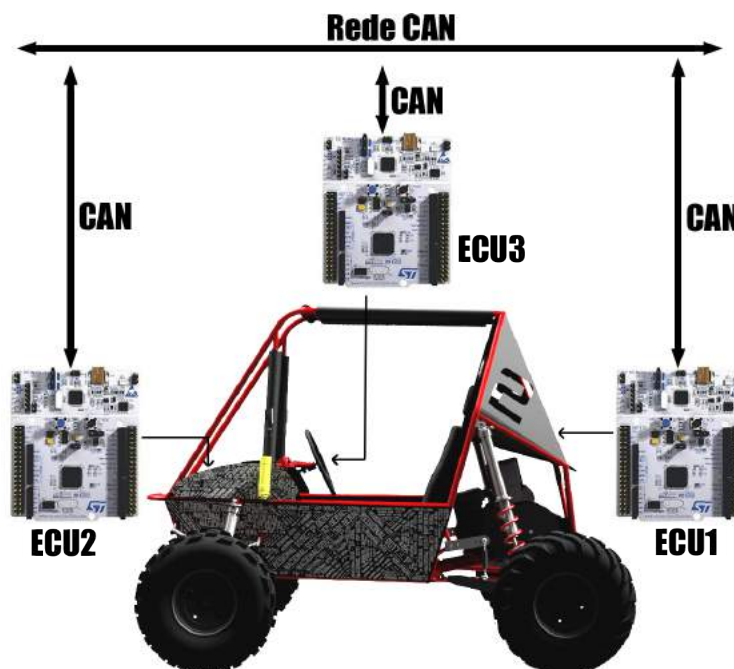
Fonte: [9].

Capítulo 3

Sistema Implementado

Primeiramente, foi feito um estudo sobre sistemas em veículos automotivos, baseado em carros confeccionados por grandes empresas, optou-se por um sistema em rede CAN, contendo três ECU interligadas entre si, possuindo três tipos de sensores: sensor de velocidade, sensor de rotações por minuto e sensor de combustível. Também, o sistema (Figura 3.1) possui o diferencial de estar diretamente interligado ao um sistema online e um banco de dados.

Figura 3.1: Representação esquemática da vista lateral do Baja ParahyBaja e locais propostos para ECU.

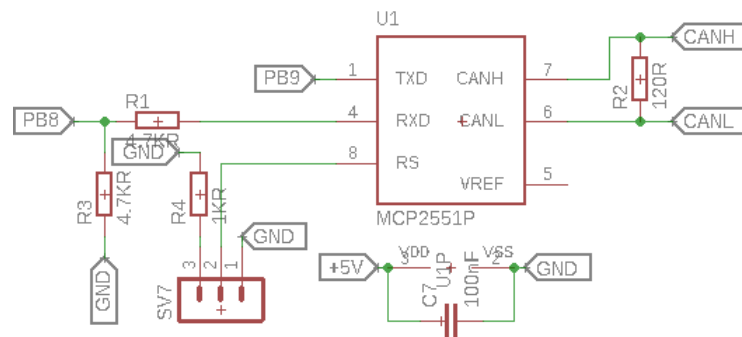


Fonte: Próprio Autor.

As ECUs utilizadas tem como base o microcontrolador NUCLEO-F303RE, fabricado pela empresa *STMicroelectronics* responsável, também, por fabricar sistemas para utilização em veículos automotivos por grande empresas como: BMW, Mercedes, etc. Entretanto, o microcontrolador utilizado para esse projeto, trata-se de um microcontrolador com arquitetura integrada, necessitando, assim, de um *transiver*.

O *transiver* utilizado é o MCP2551, segundo MCP2551 foi montado o circuito (Figura 3.2).

Figura 3.2: Circuito implementado na utilização do *transiver* MCP2551.



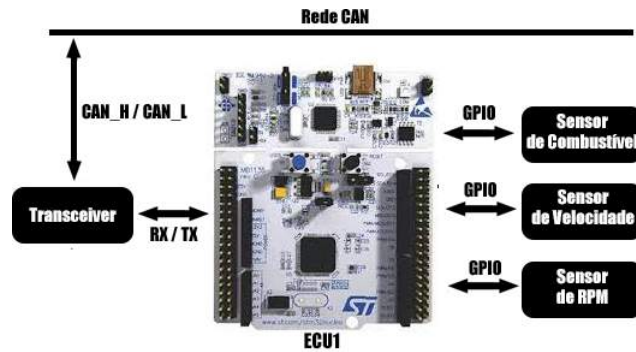
Fonte: Próprio Autor.

Pode-se verificar que o circuito trata as mensagens CAN_H e CAN_L com o objetivo de enviar dados através de seu RX e TX para o controlador interligados com os pinos PB8 e PB9, respectivamente, do microcontrolador.

3.1 ECU1

A ECU1 (Figura 3.3), é responsável por obter e trata as informações dos sensores implementados no carro e repassar essas informações ao longo do barramento CAN.

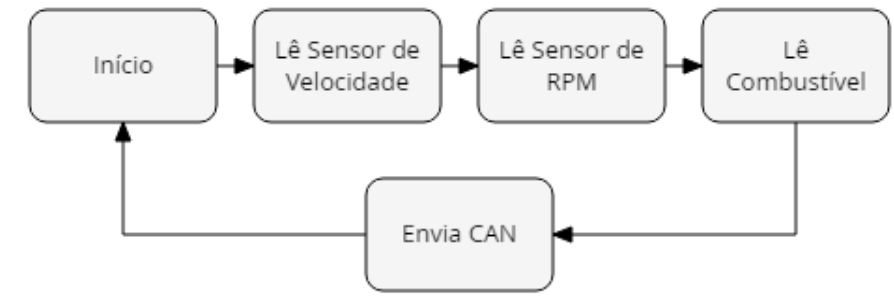
Figura 3.3: Representação esquemática da ECU1.



Fonte: Próprio Autor.

Ao ser chamada a interrupção temporal implementada no microcontrolador, toda a informação adquirida dos sensores são tratadas em inteiros de oito *bits* e enviada. O funcionamento geral da ECU1 pode ser vista decorrente do diagrama em maquina de estados da Figura 3.4.

Figura 3.4: Diagrama geral em maquina de estados da ECU1.



Fonte: Próprio Autor.

3.1.1 Sensor de Velocidade

Para o sensor de velocidade, utilizou-se o sensor indutivo (Figura 3.5 - LJ12A3-4-Z/BX) decorrente de seu baixo custo de mercado, da sua precisão e do seu grau de proteção ser IP65. Segundo ETT, o sensor possui polaridade NPN, com um nível de tensão de operação entre 6 e 36 volts em tensão contínua e uma corrente de saída de 300 mA [12].

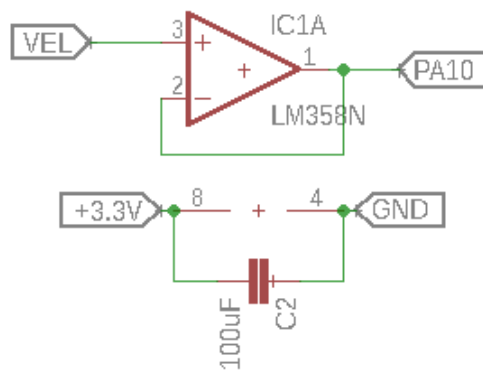
Figura 3.5: Fotografia do sensor indutivo LJ12A3-4-Z/BX.



Fonte: Próprio Autor.

Esse sinal elétrico é recebido e tratado utilizando um circuito eletrônico (Figura 3.6).

Figura 3.6: Circuito implementado na utilização do sensor indutivo.



Fonte: Próprio Autor.

Onde a entrada VEL é responsável por receber o sinal puro do sensor indutivo e o pino PA10 é responsável por enviar o sinal tratado para o microcontrolador. O circuito referente ao tratamento de dados do sensor de velocidade é composto por um amplificador operacional implementado com a função de *buffer* com o intuito de isolar o sensor [11]. Também, alimentando o mesmo com uma tensão de +3,3 V e terra é possível regular a seu nível lógico para 3,3 V, o nível lógico máximo exigido pelo microcontrolador.

O sensor é posicionado bem próximo ao eixo de rotação do carro, similar a Figura 2.15, ao longo do eixo são anexados quatro ímãs igualmente espaçados a fim de aumentar a precisão do sensor.

Em seu algoritmo de implementação, à cada volta do pneu o microcontrolador receberá quatro pulsos em sua entrada digital, entrada na qual está programada com uma interrupção externa, incrementando valor unitário a um contador específico. Ao longo de um período de tempo, já pré-determinado, esse contador é verificado se seu valor é divisível por quatro, caso não seja, recebe um decremento até ser divisível por quatro e, então, é utilizado para o cálculo da velocidade. Logo em seguida, o contador zerado, retomando, assim, o fluxo de contagem.

O veículo automotivo possui uma velocidade variada ao longo de sua trajetória, variado sua aceleração. Entretanto, utilizando-se da equação do movimento retilíneo uniformemente variado (MRUV), sabendo-se que a interrupção temporal é muito rápida e a aceleração no intervalo de tempo para o motor do carro é irrisória, podemos considerar seu movimento como um movimento retilíneo e uniforme (MRU) no pequeno intervalo de tempo e aplicar a equação MRU a fim de se obter a velocidade [11].

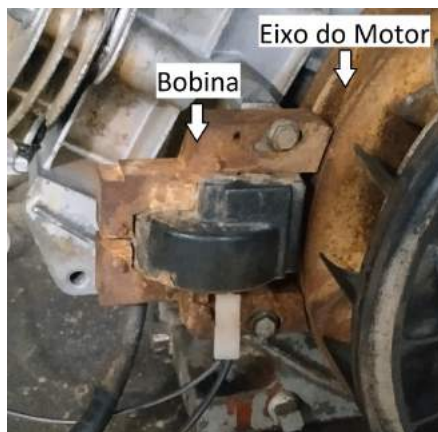
$$V_m = 3,6 \frac{P_v}{T_v} C_p \quad (3.1)$$

Em sua equação, à velocidade (V_m), primeiramente, é calculada com o contador referente a interrupção digital do sensor de velocidade (P_v) dividido pela quantidade de ímãs no eixo (P_e), posteriormente o resultado é dividido por o tempo da interrupção temporal em segundos (T_v), multiplicado por comprimento do pneu (C_p) e, finalmente, multiplicado por 3,6 para transformar de metros por segundo em quilômetros por hora.

3.1.2 Sensor de RPM

O sensor de RPM utilizado ao sistema, trata-se de uma bobina acoplada ao eixo interno do motor (Figura 3.7).

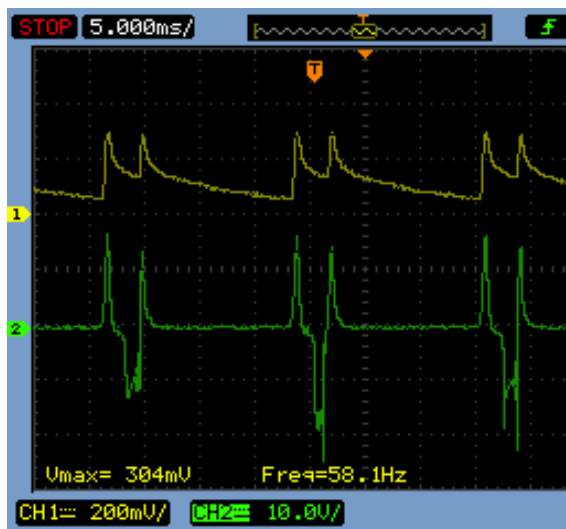
Figura 3.7: Fotografia da bobina acoplada ao eixo do motor.



Fonte: Próprio Autor.

O eixo possui um ímã em seu eixo, ao passar próximo a bobina, gera um campo eletromagnético fazendo a bobina conduzir eletricidade e projeta um espectro (Figura 3.8) cujo à frequência é modificada ao depender, especificamente, da velocidade do eixo, ou seja, velocidade do motor.

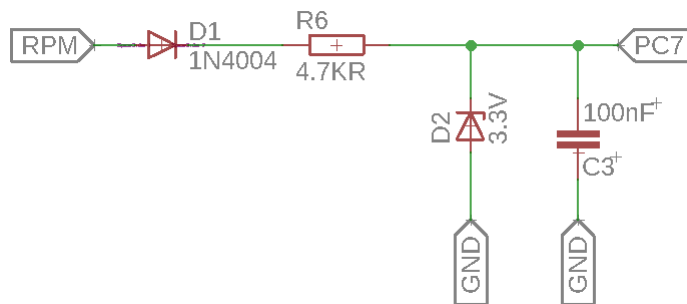
Figura 3.8: Curvas da saída do sensor de rpm puro (2) e saída do sensor de rpm tratado (1).



Fonte: Próprio Autor.

A curva é tratado através de um circuito elétrico (Figura 3.9) para tratar o sinal para uma leitura adequada do microcontrolador.

Figura 3.9: Circuito implementado na utilização do sensor de RPM.



Fonte: Próprio Autor.

Primeiramente, seu sinal é retificado através do diodo (D1), logo após é regulado a tensão a um nível de aproximadamente 3,3 V decorrente do diodo zener (D2). Também, para evitar ruídos, foi-se necessário implementar um capacitor (C3).

O seu algoritmo, funciona com interrupção externa da variação de mudança do seu nível lógico para uma possível decida e subida, como visto na Figura 3.8, terá quatro pulsos para cada volta do eixo do motor mudando apenas a sua equação e a cada interrupção um contador é incrementado. Após isso, o RPM é calculado, por meio da equação (3.2), toda vez que a interrupção temporizada é chamada.

$$RPM = \frac{60}{T_{rpm}} P_{rpm} \quad (3.2)$$

Onde a rotação do motor (RPM) é calculado mediante do valor de pulsos por volta (P_v), tempo da interrupção (T_{rpm}) e contador (P_{rpm}).

3.1.3 Sensor de Combustível

A medição de combustível é baseada em sensores capacitivos (Figura 3.10 - LJC18A3-B-Z/BY), mais especificamente, dois sensores capacitivos. O sensor foi escolhido visando, segundo HWE, o mesmo tipo de especificação do sensor indutivo implementado ao sensor de velocidade, são elas: grau de proteção ser IP65, precisão e custo de mercado [13]. Entretanto, diferentemente do sensor indutivo, o sensor capacitivo implementado ao sistema possui polaridade PNP, nível de tensão contínua de operação entre 6 a 36 V.

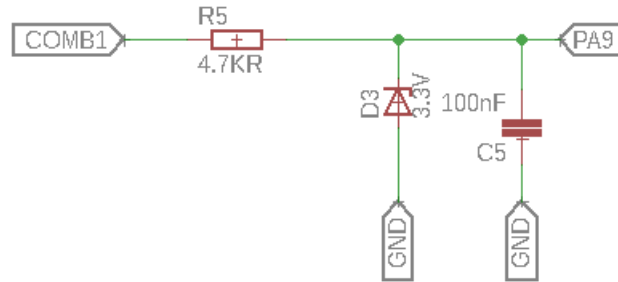
Figura 3.10: Fotografia do sensor capacitivo LJC18A3-B-Z/BY.



Fonte: Próprio Autor.

O circuito eletrônico é condicionado através de um circuito da Figura 3.11. No qual, a entrada do sensor (COMB1) é condicionada através de um diodo zener, assim como o sensor de RPM, o seu nível lógico é retificado para se adaptar ao microcontrolador.

Figura 3.11: Circuito implementado na utilização do sensor de combustível.



Fonte: Próprio Autor.

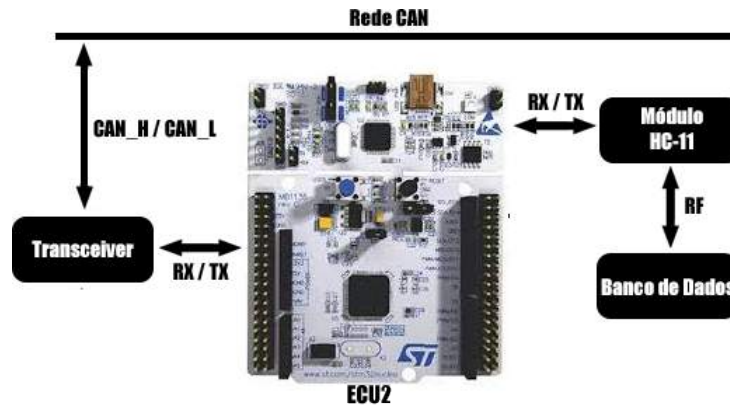
Sua lógica é mais simplificada, quando comparada ao sensores anteriores, a cada interrupção temporal a própria função verifica se o sensor está ativado ou não. O lógica possui três estados possíveis.

- Primeiro caso: Dois sensores estão ativados, a lógica identifica que o tanque de combustível está cheio.
- Segundo caso: O sensor inferior está ativado e o sensor superior desativado, identifica que o tanque de combustível está com metade do combustível máximo.
- Terceiro caso: Ambos os sensores estão desativos, a lógica identifica que o tanque está na reserva e precisa ser abastecido urgentemente.

3.2 ECU2

A ECU2 (Figura 3.12), é responsável por receber e tratar os dados enviados via rede CAN pela ECU1, também, é responsável por enviar-los, através de um módulo de radio frequência (HC-11) para o computador a fim de salvar os dados no banco de dados.

Figura 3.12: Representação esquemática da ECU2.

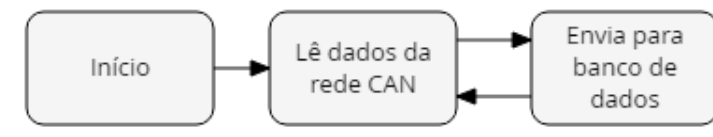


Fonte: Próprio Autor.

A transmissão de rádio frequência utiliza-se do protocolo de comunicação RS-232 [16], disponível no módulo implementado. Sua pre-configuração é feita para existir uma sincronia e uma transmissão de dados perfeita em seu par.

A implementação da ECU2 é feita da seguinte maneira: Os dados que chegam através da rede CAN são verificados e analisados se estão corretos, caso estejam, os dados são enviados via comunicação RS-232 para o módulo, que por sua vez, envia os dados para o seu respectivo par conectado ao um banco de dados [2]. Seu diagrama de maquina de estados geral pode ser visto na Figura 3.13.

Figura 3.13: Diagrama geral em maquina de estados da ECU2.

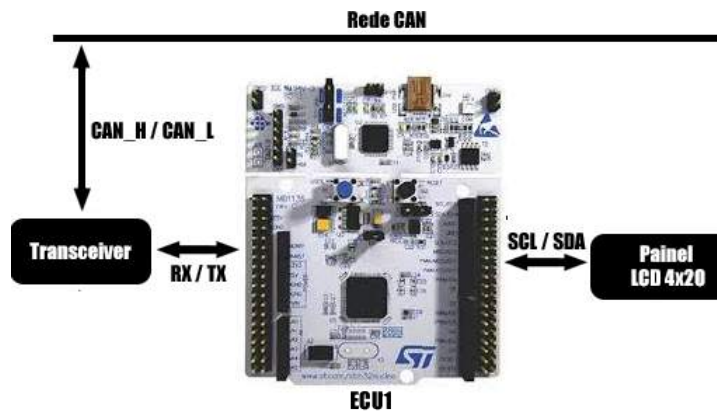


Fonte: Próprio Autor.

3.3 ECU3

A ECU3 (Figura 3.14), é responsável, também, por receber e tratar os dados enviados via rede CAN e exibi-los em um *display* de cristal líquido (LCD 4x20 [14]).

Figura 3.14: Representação esquemática da ECU3.



Fonte: Próprio Autor.

A transmissão de dados para o *display* utiliza-se do protocolo de comunicação I2C [15], assim, faz-se necessária apenas dois barramentos (SCL e SDA) para a sua transmissão.

Seu algoritmo de implementação funciona, primeiramente, verificando os dados recebidos, caso os dados recebidos estejam corretos, os mesmos são enviados para o LCD já pre-configurado para alocá-los nos seus devidos lugares. Esse painel (Figura 3.15) é instalado na parte frontal do volante, dando visibilidade ao motorista.

Figura 3.15: Exemplo do painel frontal com LCD.



Fonte: Próprio Autor.

3.4 Interface Humano-Máquina

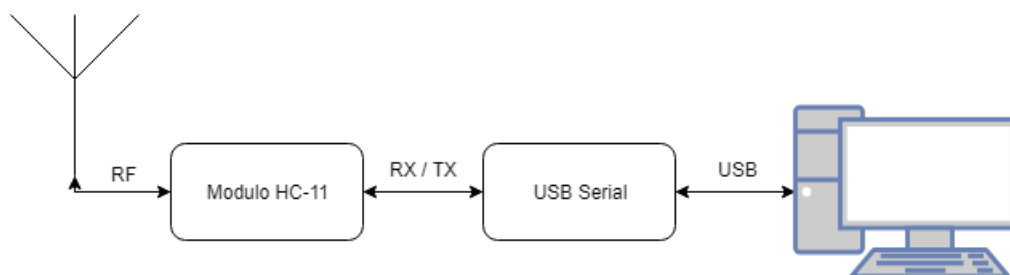
O painel foi desenvolvido em uma plataforma web, pensando justamente na praticidade de acesso e também pela grande compatibilidade, por funcionar em qualquer dispositivo que navegue na internet.

O funcionamento de tudo se dá pelo recebimento de dados que são enviados pelo microcontrolador por meio do modulo HC-11, visto que estes dados recebidos são tratados e armazenados em banco de dados, para que assim sejam exibidos na plataforma Web para amostrar dados como velocidade, RPM, nível de combustível, entre outras análises necessárias.

3.4.1 Recebimento de Dados

Os dados são recebidos por meio da utilização do módulo HC-11, em que a leitura serial é feita por meio de código desenvolvido em linguagem *python*, visto que a mesma é capaz de receber os dados com a mesma velocidade de transmissão (*baundrate*) com que os dados são enviados, criando assim, uma leitura em tempo real dos dados que devem ser aferidos e tratados. O diagrama do recebimento de dados está na Figura 3.16.

Figura 3.16: Diagrama do recebimento de dados *WEB*.



Fonte: Próprio Autor.

3.4.2 Tratamento de Dados

Os dados são recebidos, formando dez conjuntos de 8 bits para valor aferido nos sensores, estes valores, por sua vez, são convertidos separadamente em binários e concatenados após a conversão, gerando assim, dois binário de 32 bits e dois binários de 8 bits correspondendo aos valores de velocidade, RPM e combustível respectivamente. Uma vez que formado o binário de 32 bits, este valor é convertido para decimal, afim de retornar ao valor inicial aferido pelos sensores.

3.4.3 Armazenamento em Banco de Dados

Para banco de dados, foi escolhido o *MongoDB* um dos bancos não relacional (*Not Only SQL - NOSQL*). O mesmo se trata de um banco de dados orientado a documentos no formato *JSON*, ou seja, diferente de um banco de dados relacional, ele não possui como restrição a necessidade de ter as tabelas e colunas criadas previamente, aumentando assim, a sua escalabilidade e permitindo que um documento represente toda a informação necessária, com todos os dados que o sistema necessita, no formato de um *JSON* [17].

Assim como em um *JSON* utilizado em comunicações HTTP entre aplicações, no documento do *MongoDB* podem existir valores simples, como números, *strings* e datas, assim como também podem existir listas de valores e listas de objetos. Os documentos são agrupados em coleção, um conjunto de coleções formam um banco de dados. O *MongoDB* permite que seu banco de dados seja replicado para outros servidores, aumentando assim a disponibilidade de suas informações, sendo esse recurso conhecido por *replica set*. Dessa forma, cada servidor terá uma cópia dos dados.

Para o projeto, para o escopo da palavra *JSON* à ser utilizada no sistema fazemos o uso dos seguintes dados: RPM, velocidade, combustível, tempo e o índice de cada informação (*index*), conforme a Figura 3.17.

Figura 3.17: Escopo da mensagem *JSON* dentro do *MongoDB*.

```
_id: ObjectId("5caf7d41c027f221a8d0ac4f")
RPM: 4775
Velocidade: 42
Tempo: 52
Distancia: 66
index: 1
```

Fonte: Próprio Autor.

3.4.4 Plataforma WEB

Pensando em deixar mais prática a visualização das estatísticas e valores em tempo real do veículo, foi desenvolvido uma plataforma *WEB* para que fosse compatível com uma grande quantidade de usuários e que não possuísse grandes limitações. Deste modo, foi desenvolvida uma aplicação que, junto dos procedimentos citados anteriormente, é capaz de amostrar todos os dados recebidos e ter, também, uma análise ao longo do tempo sobre o desempenho do veículo capaz de cronometrar os tempos de voltas que o baja percorreu.

Toda a parte de parte de retaguarda (*back-end*) foi construída em linguagem de programação *python* por meio da plataforma *Django*, pelo fato de que todo o recebimento de dados é tido em *Python*, conseguimos fazer o uso de

Chart.js, *JQuery*, *HTML5* e *Bootstrap* com a interação dos dados aferidos inicialmente pelos sensores instalados no veículo.

O ciclo de dados da plataforma é dado, primeiramente, por receber os dados via HC-11 e os transmite pelo módulo USB-Serial, após isso, realiza uma leitura serial e trata os dados em *Python*, os dados são armazenados em banco de dados não relacional que por sua vez é responsável por apresentar os dados que serão amostrados com gráficos e valores únicos na plataforma *WEB*.

A plataforma *WEB* pode ser dividida em três cabeçalhos: Os dados únicos (Figura 3.18), os dados de rpm em função do tempo (Figura 3.19) e os dados de velocidade em função do tempo (Figura 3.20).

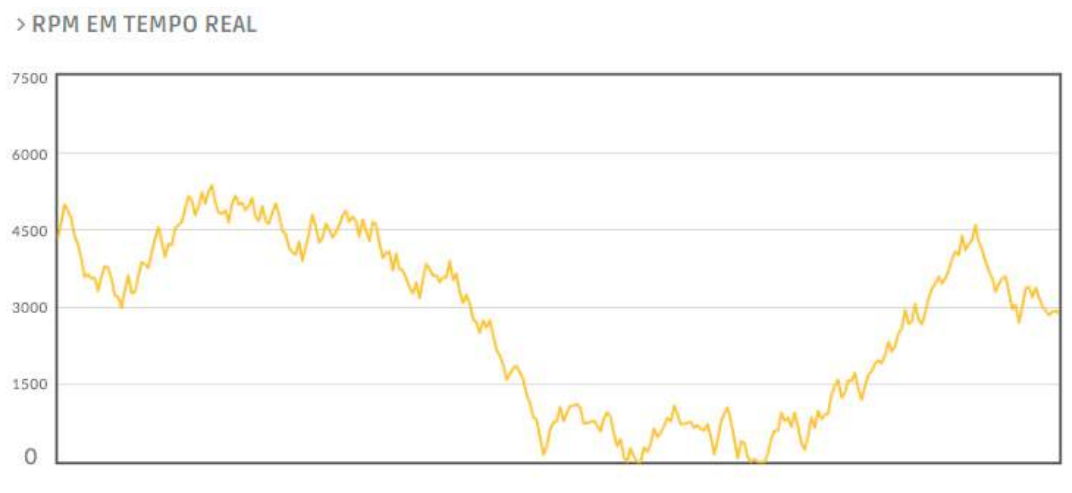
Figura 3.18: Fotografia do cabeçalho de dados únicos na plataforma *WEB*.



Fonte: Próprio Autor.

Em sua primeira parte são apresentados dados únicos, contendo a porcentagem de combustível restante no tanque, a velocidade em tempo real, o RPM em tempo real e o clima em tempo real.

Figura 3.19: Gráficos de rpm em função do tempo na plataforma *WEB*.



Fonte: Próprio Autor.

A segunda parte é responsável por amostrar o gráfico de RPM versus o seu tempo no momento que foi aferido, sendo atualizado a cada período de tempo que novos dados são recebidos e tratados.

Figura 3.20: Gráficos de velocidade em função do tempo na plataforma *WEB*.



Fonte: Próprio Autor.

A terceira, última, parte é composta por um gráfico que é responsável por plotar os valores de velocidade em função do tempo, sendo atualizado, também, a cada período de tempo que novos dados são recebidos e tratados.

Capítulo 4

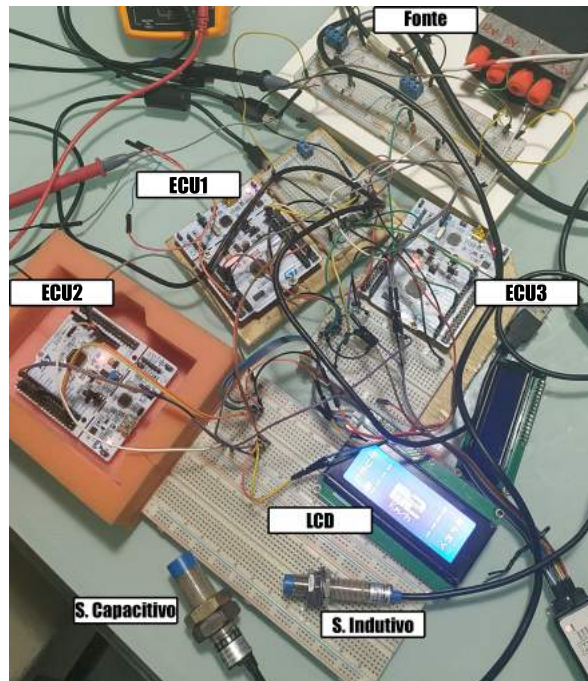
Implementação

Para implementação do sistema, etapas foram seguidas. Primeiramente foi realizada a implementação em bancada, depois foram desenvolvidas as placas e por fim foi feita a implementação no baja.

4.1 Implementação em Bancada

Primeiramente, o código em c para o funcionamento do sistema foi escrito e o circuito referente ao sistema foi montado em *protoboard*. O resultado da montagem pode ser visto na Figura 4.1.

Figura 4.1: Fotografia do sistema implementado em *protoboard*.

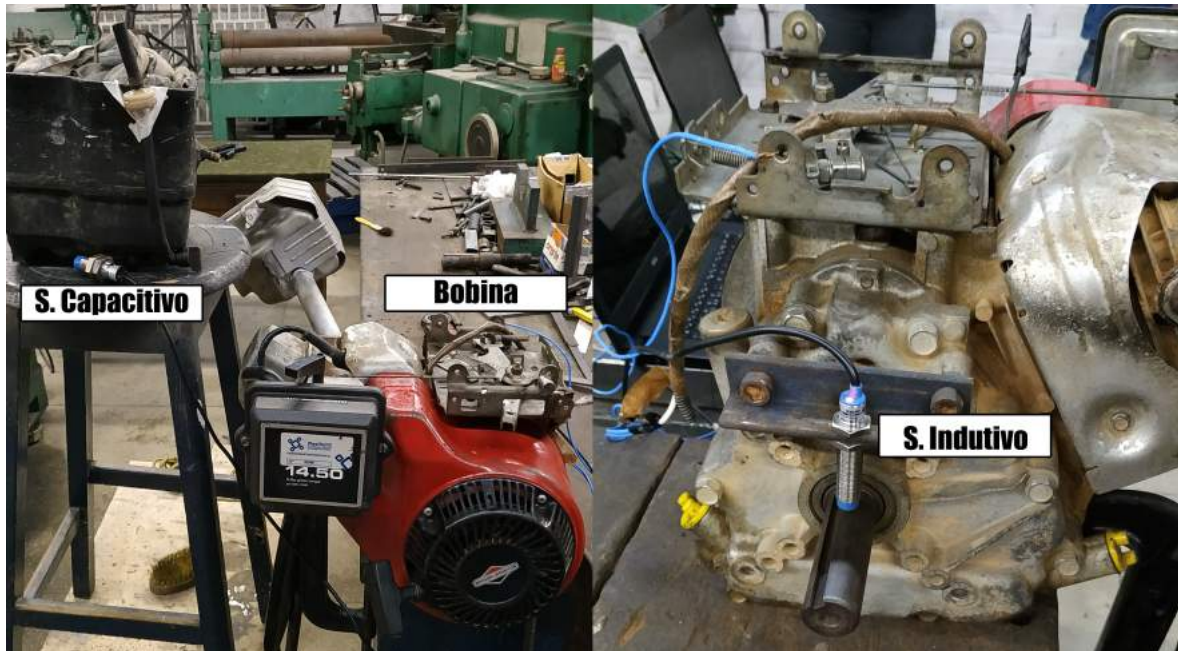


Fonte: Próprio Autor.

Onde temos sensor indutivo, sensor capacitivo, ECU1, ECU2, ECU3, todo o circuito eletrônico e uma fonte de bancada para ligar todo o sistema.

Após a montagem, foi-se para a fase de uma possível validação do sistema, para isso foi montado um motor em bancada e instalado os sensores em seu corpo. Fixou-se o sensor de velocidade ao motor, o sensor de combustível ao tanque de combustível e anexou um conector a saída da bobina interno do motor a fim de se obter o sinal do RPM. A Figura 4.2 apresenta a montagem referenciada.

Figura 4.2: Fotografia da montagem do motor e sensores em bancada.

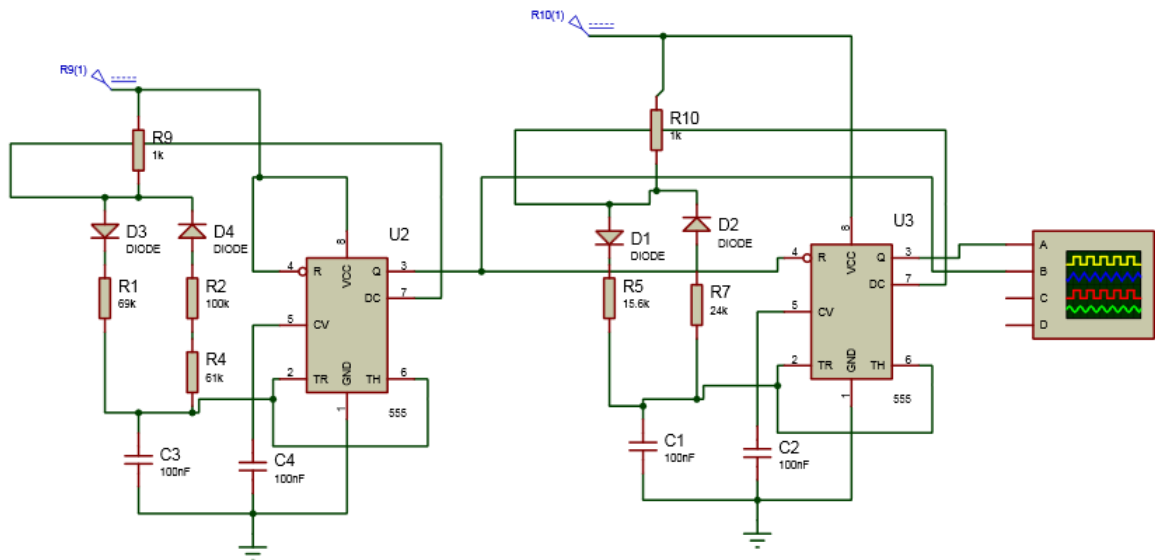


Fonte: Próprio Autor.

Para fins de verificar seu funcionamento, foi conectado aos barramentos da rede CAN um analisador lógico e seguindo o manual do motor foi possível verificar que o sistema estava em pleno funcionamento.

Além disso, foi projetado um circuito simulador (Figura 4.3) para simular o mesmo espectro de saída da bobina do sensor de RPM. Sabendo que o sinal tratado e digitalizado da tensão da vela se trata de dois pulsos agrupados. Para a digitalização se tomou por parâmetro 2,5V que é um valor frequente ao microcontrolador utilizado (STM 32).

Figura 4.3: Circuito simulador de RPM.



Fonte: Próprio Autor.

Foram utilizados dois CIs com um formato de associação em série no qual o que se encontra no início tem sua saída conectada ao pino *reset* do seguinte e a temporização é feita de modo a o primeiro CI se manter ativo durante o tempo de ocorrerem os dois pulsos e o segundo temporizado para gerar uma onda quadrada com dois períodos no tempo ativo do primeiro.

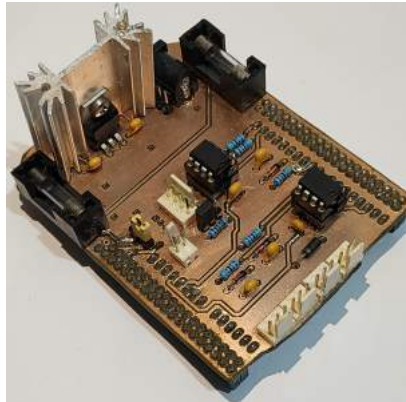
O circuito simulador foi montado e em função da imprecisão dos componentes utilizados o circuito montado gerou um sinal condizente com uma rotação de 2886 RPM. Ao conecta-lo ao sistema, foi verificado, mais uma vez, a eficiência do mesmo.

4.2 Confeção das Placas

Ao realizar os ensaios em bancada, o próximo passo foi a confecção das placas em um circuito impresso. Para isso, todo o circuito foi projetado e desenvolvido no *software* Eagle, sendo possível fazer toda a placa em seus mínimos detalhes.

Para a placa de circuito impresso referente a ECU1, além de todo o circuito já apresentado, foi introduzido componentes para regular a tensão e medidas de segurança segundo o regulamento SAE BRASIL. A MCU1 impressa e com todos os componentes soldados em seus respectivos lugares, apresenta-se na Figura 4.4

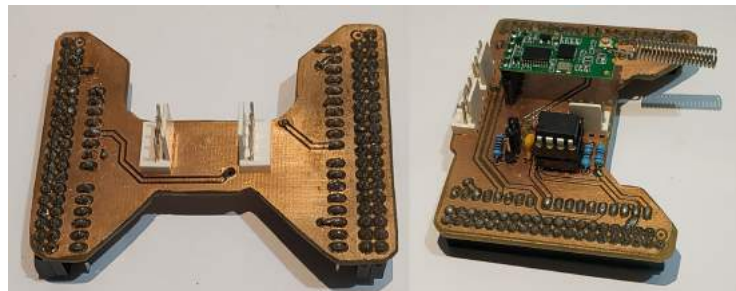
Figura 4.4: Fotografia da placa final ECU1.



Fonte: Próprio Autor.

Já a placas confeccionadas referente à ECU3 e ECU2 são apresentadas na Figura 4.5.

Figura 4.5: Fotografia das placas finais ECU3 e ECU2.



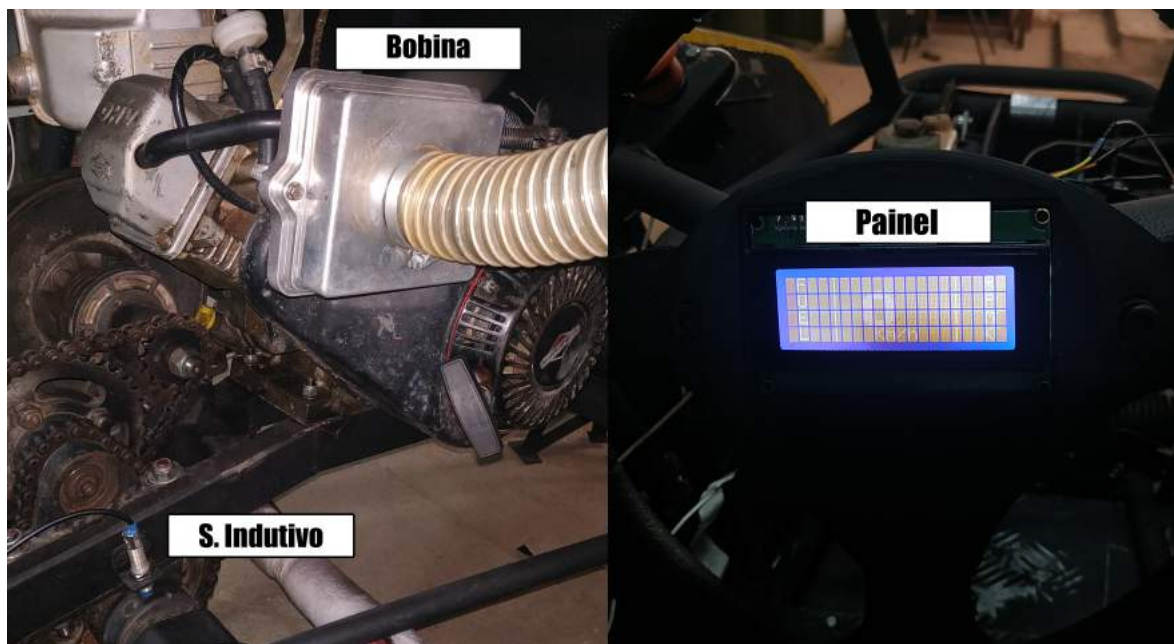
Fonte: Próprio Autor.

A ECU3, o circuito referente ao *transceiver* está localizada na parte inferior da placa, estando em sua parte superior apenas as conexões com a rede CAN e o *diplay*. A ECU2, todo o circuito já apresentado no capítulo anterior está localizado na parte superior da placa juntamente com o módulo de radio frequência (HC-11).

4.3 Implementação no Baja

O painel e todas ECU foram devidamente instaladas no carro, e então realizado o experimento no Baja. Também, foram instalados os sensores, assim como nos testes de bancada, em seus respectivos lugares. A localização do sistema na estrutura do baja apresenta-se na Figura 4.6

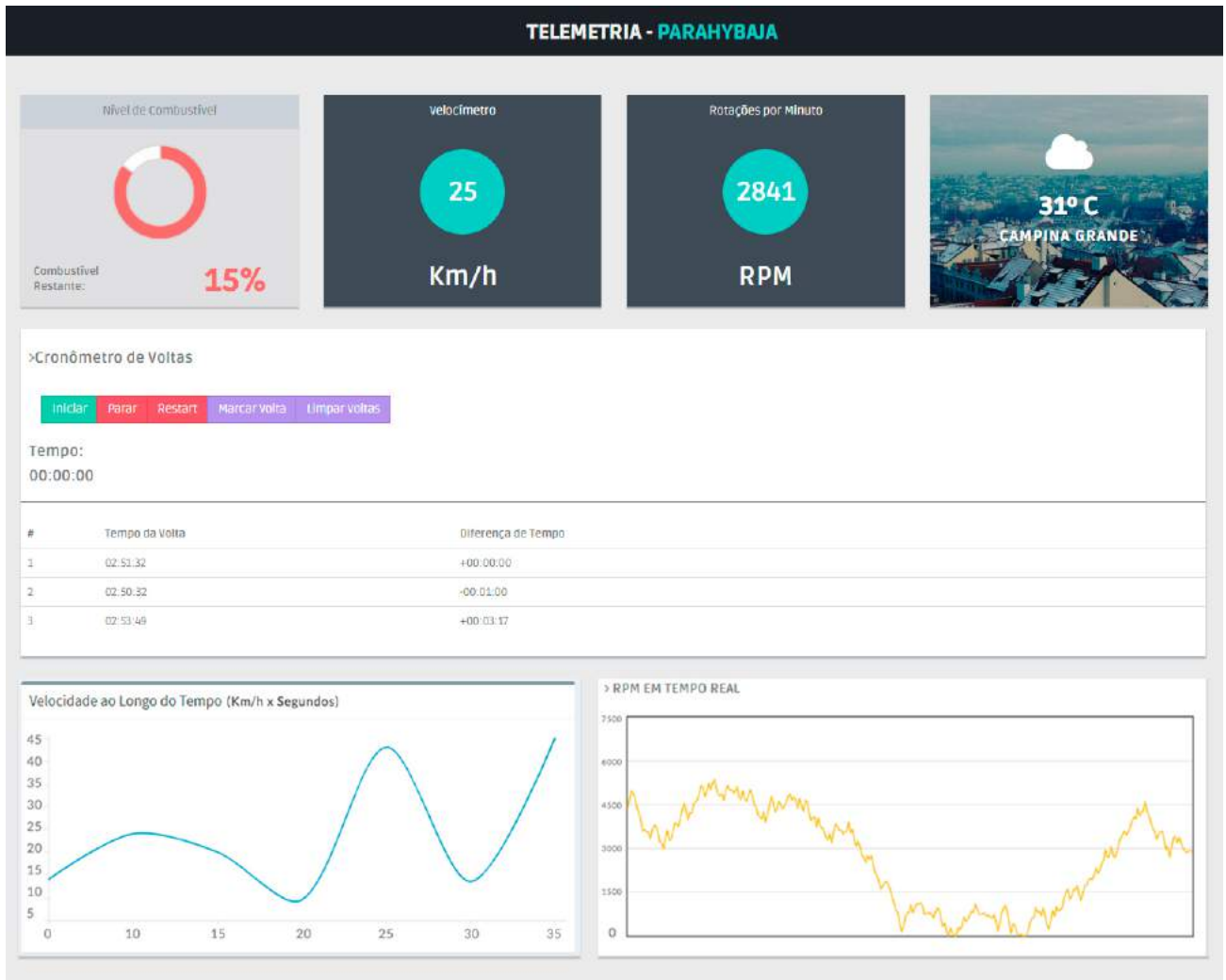
Figura 4.6: Fotografia do painel e sensores localizados no Baja.



Fonte: Próprio Autor.

Através da transmissão em tempo real dos dados via o módulo HC-11 tornou-se possível analisar os dados via plataforma *web* (Figura 4.7).

Figura 4.7: Plataforma WEB.



Fonte: Próprio Autor.

Assim como foi detalhado na seção anterior, a interface homem-máquina apresentou os valores adequados para as medições no exato momento da implementação.

Capítulo 5

Conclusões

Com o sistema de rede CAN implementado, o projeto se tornou robusto, confiável e eficaz. Sensores de velocidade e de rpm, anteriormente ineficientes devido a implementação errônea, foram validados e implementados de maneira apropriada e a quantidade de fios devido a rede CAN diminuiu.

Considerando, também, requisitos e restrições do regulamento da competição objetivos foram traçados de maneira objetiva e clara. Portanto, o objetivo deste projeto de conclusão de curso não é aprofundar um conhecimento específico mas sim aplicar conceitos e conhecimentos pré-existentes.

O desenvolvimento possibilita uma visão mais clara da construção do projeto ao longo do tempo que se foi projetado e desenvolvido. Conclui-se, portanto, que o real objetivo do projeto foi alcançado.

Por fim, devido a equipe possuir uma constante rotatividade de membros efetivos, o registro do projeto possibilita que o trabalho seja continuado para uma futura otimização do mesmo. Assim, é sugerido como trabalhos futuros a introdução de mais sensores, como: GPS, sensor de óleo, sensores de temperatura e sensor de vibração dos quais não foram possíveis adicionar ao trabalho de conclusão de curso, devido ao pequeno tempo de desenvolvimento do mesmo.

Capítulo 6

Bibliografia

- [1] ALMEIDA, G. A. de *O Protocolo CAN Bus nas Aplicações Off-Road: Uma Análise Comparativa entre os Padrões Existentes.* Society of Automotive Engineers, 2001.
- [2] WANG, K. C. *'Embedded and Real-Time Operating Systems'*, Springer, 2017.
- [3] TEXEIRA, F. C. R.; OLIVEIRA, M. C. de; HELENO, A. L. "Telemetria Automotiva Via Internet Móvel," *Revista Ciência e Tecnologia*, vol. 16, 2013.
- [4] PFEIFFER, O.; AYRE, A.; KEYDEL, C. *"Embedded Networking with CAN and CANopen"*, RTC Books, 2003.
- [5] ETSCHBERGER, K. *"Controller Area Network: Basics, Protocols, Chips and Applications"*, IXXAT Automation GmbH, 2001.
- [6] SÁ, J. da S. *"Projeto e Análise de um Sistema de Aquisição de Dados Distribuído para Aplicações em Tempo Real"*, Tese (Mestrado) - Universidade Federal de Campina Grande, 2007.
- [7] ESRO, M.; BASARI, A. A.; S, S. K.; A. SADHIQIN, A. M. I.; SYARIFF, Z. *"Controller Area Network (CAN) Application in Security System "*, *World Academy of Science, Engineering and Technology* 35, 2009.
- [8] BAUMER, *"Capacitive sensors: Versatile, contactless, durable"*, Edition 2012/2013, 2012.
- [9] BOSCH, R. *"Manual de Tecnologia Automotiva"*, Editora Edgard Blucher, Tradução da 25ª edição alemã, 2004.
- [10] SAE BRASIL *"Regulamento Baja SAE Brasil"*, 2019.
- [11] SEDRA, A. S., SMITH, K. C. *"Microeletrônica"*, Editora Edgard Blucher, Quarta Edição, Makron Books, 2000.
- [12] ETT CO., LTD. *"Datasheet: LJ12A3-4-Z/BX"*, Disponível em: http://www.ett.co.th/productSensor/LJ12A3-4-Z_BX/LJ12A3-4-Z_B\%20SENSOR.pdf Acesso: 18/06/2019.

- [13] HWE "Datasheet: LJC18A3-B-Z/BY", Disponível: <http://files.microjpm.webnode.com/200003722-858c88684b/Conductive\%20Proximity\%20Switch\%20Series.pdf> Acesso: 18/06/2019.
- [14] SYSTRONIX "Datasheet: LCD 4x20", Disponível: http://www.systronix.com/access/Systronix_20x4_lcd_brief_data.pdf Acesso: 20/06/2019.
- [15] PARET, D.; FENGER, C. "The I2c Bus: From Theory to Practice" John Wiley Sons, 1997
- [16] AXELSON, J. "Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems" Lakeview Research, 2nd, 2007
- [17] CROCKFORD, D. "The application/json Media Type for JavaScript Object Notation (JSON)" The Internet Society, 2006.