

UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Emanuel Damasceno Bocci

**Projeto de um sistema embarcado de aquisição
de dados com implementação e testes de
sistema de telemetria**

São Carlos, SP

2020

Emanuel Damasceno Bocci

**Projeto de um sistema embarcado de aquisição de dados
com implementação e testes de sistema de telemetria**

Monografia apresentada ao departamento de Engenharia Elétrica, da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientação Prof. Dr. Rafael Vidal Aroca

São Carlos, SP

2020

Agradecimentos

Agradeço à Deus por ter me dado a oportunidade de realizar e concluir este projeto e por nunca ter me abandonado mesmo nos momentos mais difíceis, suprindo todas as minhas necessidades.

Aos meus pais João Carlos Bocci e Rejane Damasceno Maciel que me apoiaram em todas as minhas escolhas e foram um suporte durante toda a caminhada até aqui.

Ao Prof. Dr. Rafael Vidal Aroca por todo auxílio na realização de cada etapa deste projeto, sendo essencial para a sua conclusão.

A todos os amigos que me acompanharam em todos esses anos, gerando momentos incríveis dos quais eu me lembrarei por toda a minha vida e com os quais eu espero poder contar por muitos anos mais.

À equipe Baja UFSCar por ter me direcionado em direção ao conhecimento e amadurecimento pessoal e profissional, onde pude ter experiências incríveis.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota”.
(Madre Teresa de Calcutá)

Resumo

Bocci, Emanuel D. Projeto de um sistema embarcado de aquisição de dados com implementação e testes de sistema de telemetria. 94 p. Monografia – Departamento de Engenharia Elétrica (DEE), Centro de Ciências Exatas e de Tecnologia (CCET), Universidade Federal de São Carlos (UFSCar), 2020.

Os projetos de extensão são uma grande porta para o desenvolvimento da ciência e engenharia no ambiente universitário. Com o projeto Baja SAE não é diferente, pois ele possibilita que diversos campos da engenharia sejam postos à prova para a criação de um veículo *off-road* que tenha melhor desempenho possível, necessitando, para isso, de um esforço grande em torno do estudo e aplicação de vários conceitos das engenharias. E nesse contexto, o projeto eletrônico tem papel fundamental, não somente na criação de um sistema de aquisição de dados para a aferição do desempenho final do carro, como também servindo como um suporte para todo o projeto mecânico, fornecendo dados através da instrumentação dos subsistemas para sua otimização e melhoria.

Este trabalho tem por objetivo a apresentação de todo o processo de desenvolvimento do sistema eletrônico de aquisição de dados que se utiliza de um protocolo de comunicação interno e um sistema de telemetria da equipe de Baja da Universidade Federal de São Carlos, através da utilização de alguns tipos de sensores para a leitura de informações do veículo, para então processar e apresentar todas elas para a avaliação do desempenho do carro em testes e durante provas das competições organizadas pela SAE Brasil.

Além dos estudos referentes aos temas importantes do assunto deste trabalho, são abordados também as tecnologias utilizadas em cada etapa, metodologias para o projeto de cada subsistema de medição, a estrutura da comunicação interna com o protocolo de comunicação automotivo CAN, testes com o módulo de radiofrequência LoRa para a implementação de telemetria buscando definir uma transmissão de dados por uma maior distância em comparação com módulos utilizados em estudos anteriores na aplicação definida de mobilidade, e o desenvolvimento da interface que apresenta as informações de forma mais facilitada e intuitiva para a equipe no *box*. O resultado final foi a concepção de um sistema robusto, que inclui *hardware* e *software*, capaz de obter e processar todas as informações necessárias para a avaliação do veículo, enviando em tempo real toda a informação para a equipe, e com potencial para a sua evolução e melhoria, através da adição de novos sensores ou alteração dos já utilizados.

Palavras-chave: Baja, Eletrônica embarcada, Sensoriamento, Telemetria, Aquisição de dados.

Abstract

Extension projects are a great way for science and engineering development in the University. It is not different with the SAE Baja project, since it requires different engineering concepts to be utilized to develop a competitive off-road vehicle prototype, thus requiring great effort in the study and application of concepts from mechanical, electrical, materials and production engineering, among others. In this context, the electronic system has a fundamental role, not only regarding the acquisition of data but also as a support for the mechanical part by providing input data that allows global optimization of the prototype.

This project presents the development of the data acquisition system, which uses an internal communication protocol and telemetry system from the Baja SAE team of the Federal University of São Carlos, through some sensors that collect information from the vehicle, which is processed and then shown to the team so it is possible to evaluate its performance. To achieve this, the research was made to choose adequate components for the system, aiming for high data reliability and robustness.

Moreover, this project also addresses the technologies used in each step, the methods used in each measurement system, the structure of the internal communication with the CAN automotive protocol, tests with the telemetry module LoRa to create a data transmission over a greater distance in comparison with other modules already used in previous studies for the proposed application, and the development of an interface that shows information in an easier and more intuitive way for the team. The final result was the creation of a robust electronic system that includes hardware and software, is able to collect, process and send real time information to the team, and has potential for improvement by adding new sensors or changing the ones already in use.

Keywords: Baja, Embedded electronic, Sensing, Telemetry, Data acquisition.

Lista de ilustrações

Figura 1 – Comparativo entre protocolos de comunicação por custo de nós e velocidade. Adaptado de (ARTAL; CARABALLO; DUFO, 2014).	22
Figura 2 – Diagrama do projeto eletrônico da equipe UFVBaja (DIAS, 2010).	22
Figura 3 – Protótipo M8 da equipe Baja UFSCar durante a competição Nacional de 2020.	27
Figura 4 – Parte da arquitetura de um sistema embarcado veicular (NAVET; SIMONOT-LION, 2017).	28
Figura 5 – Unidade de controle eletrônico Bosch para monitoramento do motor (BOSCH, 2020).	29
Figura 6 – Funções realizadas por ECUs em um carro (ENGINEERING, 2018).	30
Figura 7 – Ilustração da atuação do sistema eletrônico de controle de estabilidade (PAULO, 2011).	31
Figura 8 – Crescimento na utilização dos sistemas embarcados em carros (SOLUTIONS, 2018).	31
Figura 9 – Camadas definidas pelo modelo OSI para protocolos de comunicação (DLTEC, 2011).	32
Figura 10 – Interface de um sistema de aquisição de dados de um carro de corridas (SERGERS, 2014).	34
Figura 11 – Sensor indutivo, em detalhe uma ilustração de seu esquema de funcionamento.	37
Figura 12 – Montagem do sensor indutivo em uma das rodas do carro.	37
Figura 13 – Imagem do sensor de efeito Hall, indicando sua pinagem e sua representação em esquemáticos eletrônicos (MICROSYSTEMS, 2016).	38
Figura 14 – Imagem do sensor LM35, indicando sua pinagem e sua representação em esquemáticos eletrônicos (INSTRUMENTS, 2017).	39
Figura 15 – Módulo GPS utilizado no projeto.	40
Figura 16 – Rota gerada durante os testes com o módulo GPS.	42
Figura 17 – Célula de carga (a), posicionamento (b) e esquema de ligação dos strain gauges que a compõem (c). Adaptado de (WEIGHING, 2010).	43
Figura 18 – Circuito utilizado para o projeto com a célula de carga (BROWN, 1998).	44
Figura 19 – Gráfico gerado da relação entre a tensão elétrica medida e massas utilizadas.	44
Figura 20 – Unidade inercial utilizada nos testes (ARDUINO, 2019).	45
Figura 21 – Módulo construído para a leitura do nível de combustível.	45
Figura 22 – Circuito divisor de tensão para leitura da carga da bateria.	46
Figura 23 – Diferentes modelos de PIC disponíveis no mercado (MICROCHIP, 2020).	47

Figura 24 – Modelo gravador programador de PIC (AUTOCORE, 2020).	47
Figura 25 – Algumas das diferentes placas Arduino (THOMSEN, 2014).	48
Figura 26 – Esquemático de portas do microcontrolador ATmega328 (CAMPOS, 2015).	49
Figura 27 – Um dos modelos de ARM disponíveis. Adaptado de (MARQUES; CARRIJO; JÚNIOR, 2019).	49
Figura 28 – Placa de desenvolvimento STM32 F103C8T6 (MARQUES; CARRIJO; JÚNIOR, 2019).	50
Figura 29 – <i>Display</i> utilizado para apresentar informações ao piloto (DWIN, 2012).	51
Figura 30 – <i>Layout</i> das informações mostradas para o piloto.	52
Figura 31 – Padrão dos arquivos com os dados gravados no cartão de memória. . .	52
Figura 32 – Taxa de comunicação da rede CAN em relação ao tamanho do cabeamento (SOUZA, 2013).	53
Figura 33 – Definição do protocolo CAN nas camadas do modelo padrão OSI (MARQUES; FRANCO, 2004).	54
Figura 34 – Representação dos níveis de tensão para a criação de um <i>bit</i> dominante (NATALE et al., 2012).	54
Figura 35 – Representação da arbitragem de mensagens utilizada no protocolo CAN. (HPL, 2002).	55
Figura 36 – Os formatos do <i>frame</i> de dados (HPL, 2002).	56
Figura 37 – Circuito utilizado para a implementação da rede CAN em todos os módulos.	57
Figura 38 – Placas produzidas para a leitura de sensores.	60
Figura 39 – Placa para a gravação dos dados no cartão SD.	61
Figura 40 – Placa para a utilização do <i>display</i> LCD.	61
Figura 41 – Placa para a utilização de GPS e telemetria.	62
Figura 42 – Módulo de radiofrequência Digi XBee (DIGI, 2020).	64
Figura 43 – Topologia do projeto, com dois módulos XBee em comunicação ponto-a-ponto.	65
Figura 44 – Interface do <i>software</i> X-CTU para configuração dos módulos XBee. . . .	66
Figura 45 – Configuração dos parâmetros do <i>Range Test</i> para o teste de distância com XBee.	66
Figura 46 – Local e posição das distâncias para os testes utilizando módulo XBee. .	67
Figura 47 – Resultado observado no teste a uma distância de 50m do ponto de envio.	67
Figura 48 – Resultado observado no teste a uma distância de 200m do ponto de envio.	68
Figura 49 – Desempenho do módulo XBee no <i>Range Test</i> em função da distância. .	68
Figura 50 – Desempenho do módulo XBee no teste de latência em função da distância.	69
Figura 51 – Topologia de comunicação da rede LoRaWAN em ambientes urbanos (JúNIOR, 2016).	71

Figura 52 – Capacidade da rede LoRa em função dos parâmetros da transmissão (TEIXEIRA; ALMEIDA, 2017).	73
Figura 53 – Módulo Heltec Wi-Fi LoRa 32 e sua pinagem (HELTEC, 2018).	73
Figura 54 – Local inicial para os testes utilizando o módulo LoRa.	74
Figura 55 – Local utilizado na segunda etapa do teste de distância com o módulo LoRa.	75
Figura 56 – Desempenho do módulo LoRa no teste de distância.	76
Figura 57 – Desempenho do módulo LoRa no teste de latência.	77
Figura 58 – Pista utilizada para o enduro nas competições.	79
Figura 59 – Teste de distância de envio, XBee <i>vs</i> LoRa (SF7, SF10, SF12).	79
Figura 60 – Teste de latência, XBee <i>vs</i> LoRa (SF7, SF10, SF12).	80
Figura 61 – Diagrama de fluxo dos dados no projeto.	81
Figura 62 – Interface gráfica desenvolvida para visualizar os dados coletados.	82
Figura 63 – Esquemático da PCI do módulo de leitura de sensores.	91
Figura 64 – Esquemático da PCI do módulo de gravação dos dados em cartão SD.	92
Figura 65 – Esquemático da PCI do módulo do <i>display</i> LCD.	93
Figura 66 – Esquemático da PCI do módulo de GPS e Telemetria.	94

Lista de tabelas

Tabela 1 – Matriz de decisão das opções de microcontroladores	51
Tabela 2 – Padrão definido para as mensagens enviadas pela rede.	58
Tabela 3 – Principais especificações dos módulos testados.	80

Lista de Siglas

Sigla	Descrição
<i>A/D</i>	Conversor Analógico Digital
<i>ABS</i>	<i>Anti-Lock Braking System</i>
<i>BCM</i>	Módulo de controle do chassi e outros componentes de um veículo - <i>Body Control Module</i>
<i>CAN</i>	<i>Controller Area Network</i>
<i>CSS</i>	Técnica de espalhamento espectral de sinais - <i>Chirp Spread Spectrum</i>
<i>CVT</i>	Transmissão continuamente variável - <i>Continuous Variable Transmission</i>
<i>ECM</i>	Módulo de controle das funções do motor - <i>Engine Control Module</i>
<i>ECU</i>	Unidade de controle eletrônico - <i>Electronic Control Unit</i>
<i>ESC</i>	Módulo de controle de estabilidade - <i>Electronic Stability Control</i>
<i>GPS</i>	Sistema de posicionamento global - <i>Global Positioning System</i>
<i>HTTP</i>	Protocolo <i>web</i> para visualização de páginas - <i>Hypertext Transfer Protocol</i>
<i>I2C</i>	<i>Inter-Integrated Circuit</i>
<i>IDE</i>	Ambiente de desenvolvimento integrado - <i>Integrated Development Environment</i>
<i>IEEE</i>	Instituto de Engenheiros Eletricistas e Eletrônicos - <i>Institute of Electrical and Electronics Engineers</i>
<i>IoT</i>	Internet das coisas - <i>Internet of Things</i>
<i>ISM</i>	Banda de radiofrequências não licenciada - <i>Industrial, Scientific and Medical</i>
<i>ISO</i>	Organização internacional de normalização - <i>International Organization for Standardization</i>
<i>LIN</i>	Protocolo de comunicação automotivo - <i>Local Interconnect Network</i>
<i>LoRaWAN</i>	<i>Long Range Wide Area Network</i>
<i>LPWAN</i>	<i>Low Power Wide Area Network</i>
<i>NMEA</i>	<i>National Marine Electronics Association</i>
<i>OSI</i>	<i>Open System Interconnection</i>
<i>PCI</i>	Placa de circuito impresso
<i>PWM</i>	Modulação por largura de pulso - <i>Pulse Width Modulation</i>
<i>RSSI</i>	<i>Received Signal Strength Indicator</i>
<i>RTOS</i>	<i>Real Time Operating Systems</i>
<i>SAE</i>	Sociedade dos engenheiros da mobilidade - <i>Society of Automotive Engineers</i>
<i>SF</i>	Fator de espalhamento - <i>Spread Factor</i>
<i>SNR</i>	Taxa entre sinal e ruído - <i>Signal Noise Rate</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>TCM</i>	Módulo de controle da transmissão - <i>Transmission Control Module</i>
<i>TCU</i>	Módulo de controle dos sistemas de entretenimento veicular - <i>Telematics Control Unit</i>
<i>USART</i>	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>

Sumário

1	INTRODUÇÃO	21
1.1	Objetivo Geral	23
1.2	Objetivos Específicos	23
1.3	Motivação	24
1.4	Organização da Monografia	24
2	CONCEITOS PRELIMINARES	26
2.1	Projeto Baja SAE	26
2.2	Eletrônica Embarcada Automotiva	28
2.3	Protocolos de Comunicação	32
2.4	Aquisição e Análise de Dados	34
3	ELETRÔNICA EMBARCADA NO PROJETO BAJA	36
3.1	Sistemas de Medição	36
3.1.1	Velocidade	36
3.1.2	Rotação do motor	38
3.1.3	Temperaturas da CVT e da caixa de redução	38
3.1.4	Módulo GPS	40
3.1.5	Nível de combustível	42
3.1.6	Carga da bateria	45
3.2	Processamento dos dados	46
3.2.1	PIC	46
3.2.2	Arduino	47
3.2.3	ARM	48
3.2.4	Escolha do Microcontrolador	50
3.3	Análise de dados	51
3.4	Protocolo de Comunicação	53
3.4.1	Protocolo CAN	53
3.4.2	Criação das mensagens nos módulos	58
3.5	Resultados Preliminares	59
4	SISTEMA DE TELEMETRIA	63
4.1	XBee PRO S1	63
4.1.1	Teste de envio e taxa de dados	66
4.1.2	Teste de latência	69
4.1.3	Consumo de energia	70

4.2	LoRa	70
4.2.1	Teste de envio e taxa de dados	74
4.2.2	Teste de latência	76
4.2.3	Consumo de energia	77
4.3	Comparação dos resultados	78
5	INTERFACE DE ACOMPANHAMENTO	81
6	CONSIDERAÇÕES FINAIS	83
6.1	Conclusão do projeto	83
6.1.1	Sugestões de estudos e trabalhos futuros	84
	REFERÊNCIAS	86
A	ESQUEMÁTICOS DAS PLACAS UTILIZADAS	91

1 Introdução

Sistemas embarcados para aquisição de dados estão cada vez mais presentes nos veículos utilizados atualmente. A indústria automotiva é um dos setores com maior volume financeiro do mundo, produzindo milhões de carros todo ano, gerando avanços em tecnologia para fornecer melhorias em performance, conforto, segurança e outras funcionalidades, e para que isso ocorra, tais avanços têm sido disponibilizados através de tecnologias em eletrônica e *softwares*. Desde a década de 1990, o setor de eletrônica embarcada cresce a uma taxa de 10% ao ano (NAVET; SIMONOT-LION, 2017). Atualmente, os carros são equipados com mais de 50 unidades microcontroladas (ECUs), e por conta de fatores como o aumento dos preços de combustíveis e a preocupação com questões ambientais houve uma aceleração do desenvolvimento e inovação no automobilismo (Fleming, 2011). Isso se deve à evolução da tecnologia e à criação de novos e melhores equipamentos para medição, processamento e envio dos mais diversos dados que se queira medir.

Contudo, a quantidade de informações de diversas naturezas em tais sistemas é enorme e eles devem ser robustos o suficiente para atender a demanda de novos e complexos processos em um tempo de resposta apropriado a aplicação automotiva, onde cada vez mais se torna necessária a comunicação em tempo real e com uma qualidade de informação de grande confiabilidade. Por isso, se tornou indispensável a implementação de controladores que se comuniquem entre si, processem informações em alta velocidade e que possam controlar e atuar com precisão e segurança em determinadas tarefas. Tais necessidades alavancaram o aparecimento de inovações também no campo da comunicação de dados, como a implantação de barramento de campo (*fieldbus*) e equipamentos com inteligência própria (MARQUES; FRANCO, 2004).

As redes de comunicação interna nos automóveis se tornaram a melhor solução ao atender todos os requisitos necessários, facilitando a comunicação entre diferentes controladores com alta velocidade e confiabilidade. Então, por esse motivo, a maior parte dos veículos comercializados atualmente possuem uma ou mais redes de comunicação interna, tal como o barramento CAN, LIN ou *FlexRay*. Por isso, existem diversos tipos diferentes de protocolos utilizados na indústria automotiva, projetados baseando-se em determinadas aplicações, tendo portanto diferentes focos, como o aumento da velocidade ou taxa de transmissão, ou até mesmo consumo de energia e custo de implementação. Uma comparação entre vários protocolos de comunicação automotivos em relação as suas taxas de dados e custo de implementação, é feita na figura 1.

O projeto Baja tem o intuito de promover a pesquisa e desenvolvimento de soluções para uma categoria especial de veículos e frequentemente atrai a atenção de alunos de

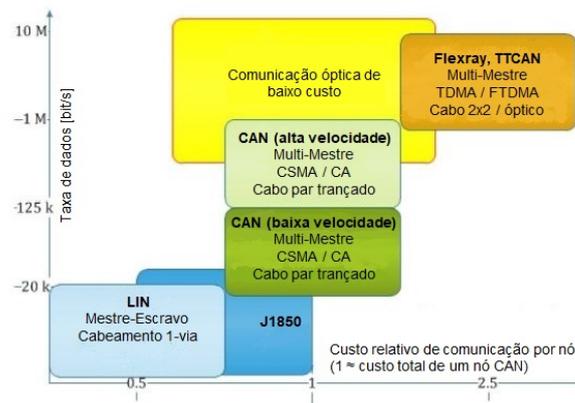


Figura 1 – Comparativo entre protocolos de comunicação por custo de nós e velocidade. Adaptado de (ARTAL; CARABALLO; DUFO, 2014).

engenharia, motivando o estudo aprofundado em temas multidisciplinares da engenharia. Ele engloba o projeto de cada subsistema mecânico (powertrain, suspensão, freios e estrutura) e permite o desenvolvimento de um sistema eletrônico para auxiliar na análise de desempenho e ser utilizado na coleta de dados utilizados na otimização do projeto.

O sistema eletrônico de um Baja geralmente é o que recebe menos investimento de todo o veículo, no entanto, em equipes de destaque internacional como é o caso da FEI (FEI, 2020), de São Bernardo do Campo, que é tetracampeã mundial de Baja, temos o exemplo de sistemas eletrônicos avançados de monitoramento e telemetria. Por isso, é evidente o potencial para projetos eletrônicos e o número de variáveis que podem ser monitoradas é muito grande, de forma que diversas equipes investem seus esforços em inovações e melhorias nessa área. Como exemplo disso, a figura 2 apresenta um diagrama do projeto de eletrônica embarcada da equipe da Universidade Federal de Viçosa, UFVBaja.

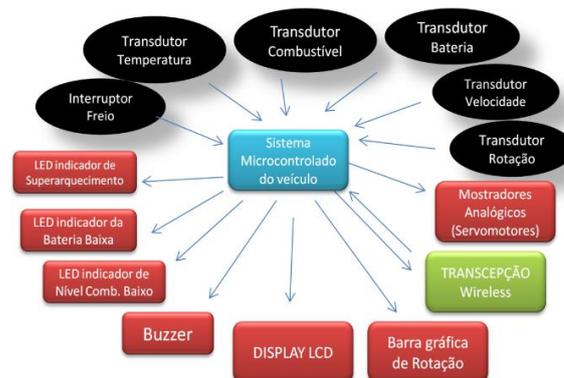


Figura 2 – Diagrama do projeto eletrônico da equipe UFVBaja (DIAS, 2010).

Estas informações coletadas são de extrema importância para a equipe, devida à necessidade de garantir ao piloto informações precisas sobre as condições do carro em testes ou até mesmo durante as provas dinâmicas da competição. Com um *feedback* como esse, há a possibilidade de uma melhor dirigibilidade, além de permitir uma maior agilidade

na ação por parte da equipe no caso de problemas em algum componente monitorado do veículo.

Além disso, nota-se que embora o diagrama mostre que são medidas e monitoradas diversas variáveis, ainda existem muitas oportunidades de inovação em aspectos de monitoramento, rede de comunicação interna no veículo e telemetria.

Dentro desse contexto, a telemetria pode exercer um papel essencial para o avanço de um projeto eletrônico para fins de análise de performance automotiva. Essa ferramenta tem seu uso relacionado ao enorme avanço em termos de engenharia em competições automobilísticas. Na fórmula 1, por exemplo, – principal referência nas competições automotivas e pioneira no uso da telemetria – logo no início dos anos 80 se utilizavam computadores para apenas algumas tarefas básicas, sendo usada somente para cronometragem e monitorando alguns parâmetros como o combustível e rotações do motor. Com o passar dos anos a evolução foi tamanha que atualmente são medidas tantas informações que é possível dizer exatamente em quais pontos da pista houveram trocas de marchas, níveis de frenagem e aceleração do piloto, monitorando dados completos do motor, chassi e dirigibilidade.

1.1 Objetivo Geral

O objetivo principal do presente trabalho foi o desenvolvimento de um sistema de aquisição de dados como um computador de bordo automotivo, de forma a possibilitar o acompanhamento e a avaliação da performance de um veículo protótipo do tipo Baja, através da inclusão de sensores e dispositivos que possam coletar as informações em tempo real e apresentar tais informações ao piloto e à equipe durante testes e competições. Neste projeto busca-se a evolução de um sistema básico de aquisição de dados automotivo através da implementação de um sistema de telemetria robusto, contando com testes de módulos de radiofrequência com tecnologias de grande potencial para o envio dos dados coletados no carro à distância, como é o caso das redes ZigBee e LoRaWAN que são abordadas em diferentes aspectos e características.

Visando a evolução contínua do projeto eletroeletrônico da equipe Baja UFSCar, a elaboração deste trabalho foi idealizada também de forma a garantir que o mesmo possa ser utilizado como uma fonte para consulta em projetos futuros, gerando uma gestão de conhecimento eficiente para a equipe.

1.2 Objetivos Específicos

- Estudos dos conceitos relacionados ao projeto de um sistema de coleta de dados;
- Desenvolvimento de diferentes sistemas de medição de forma robusta e precisa e a

escolha dos componentes utilizados em cada um;

- Implementação de um protocolo de comunicação interna para assegurar uma comunicação eficiente entre todos os subsistemas instalados no carro;
- Criação de um sistema de telemetria capaz de enviar todos os dados coletados do carro em tempo real durante testes e em provas das competições;
- Implementação de uma interface gráfica de visualização utilizando a linguagem JavaScript;
- Documentação de todo o desenvolvimento do projeto para garantir a gestão do conhecimento e a sua evolução dentro da equipe pelos próximos membros.

1.3 Motivação

A motivação deste trabalho se deu principalmente por conta da participação do autor na Equipe Baja UFSCar durante os anos de 2014 a 2017, tendo atuado como líder do projeto elétrico durante quase todo este período e, por isso, se dedicando a ele de forma ativa, desenvolvendo toda sua estrutura e buscando a sua evolução mesmo após a saída da equipe. Além disso, ainda restaram trabalhos a serem desenvolvidos e a melhoria do sistema de telemetria, necessitando de estudos e testes complementares.

Outro fator importante é que o desenvolvimento de um sistema de aquisição de dados veicular faz parte de uma linha de estudo relacionado às ECUs que compõem os carros comerciais, de forma que esta é uma área em constante crescimento e que demanda muita pesquisa e desenvolvimento, além do fato de que um projeto como esse envolve conceitos de diferentes partes da engenharia elétrica, como instrumentação, eletrônica analógica e digital, programação, análise de sinais, interfaceamento e telecomunicações. Com isso em mente, a realização deste projeto envolve a utilização de conceitos muito importantes para o desenvolvimento do autor como engenheiro, auxiliando na sua escolha de segmento do mercado de trabalho no futuro.

1.4 Organização da Monografia

Para um melhor entendimento, a forma mais adequada de apresentar seu desenvolvimento é seguindo a mesma lógica de divisão das etapas de projeto. Por isso, a estrutura do trabalho é a seguinte:

- No capítulo 2 serão primeiro abordados os temas relevantes ao projeto como uma forma de revisão bibliográfica, tangendo os temas que se relacionam com a definição do projeto Baja, eletrônica embarcada automotiva, protocolos de comunicação que

são de extrema importância para sistemas onde ocorre fluxo de dados em tempo real, e princípios da aquisição e análise de dados em competições;

- No capítulo 3 são apresentados os sistemas de medição empregados no projeto e a escolha do componente microcontrolador utilizado para que toda a informação adquirida pudesse ser processada e utilizada, além da escolha do protocolo de comunicação interna para aumentar a robustez do projeto;
- Em seguida no capítulo 4, são apresentados os testes realizados com módulos de radiofrequência para o desenvolvimento de um sistema de telemetria, abordando de forma prática os principais parâmetros considerados para a aplicação em mobilidade e buscando uma melhor solução em relação ao uso do módulo XBee que não demonstra bons resultados;
- Com isso, no capítulo 5, é discutido o uso de uma interface que apresente as informações recebidas no *box* para a equipe de forma mais clara, complementando a telemetria e facilitando seu uso em testes e competições;
- Por fim, no capítulo 6, são feitas algumas considerações e conclusões finais sobre o projeto e algumas sugestões de estudos e trabalhos futuros que possam complementar essa monografia.

2 Conceitos Preliminares

2.1 Projeto Baja SAE

O projeto Baja é um desafio criado pela SAE (traduzido do inglês, Sociedade de Engenheiros de Mobilidade) para os alunos de engenharia e tem como principal finalidade o desenvolvimento de um veículo *off road*, para que sejam aplicados na prática os conhecimentos adquiridos na graduação. Cabe aos alunos a definição e concepção de todo o projeto, bem como de sua manufatura e montagem, além de envolver algumas outras atribuições, como a gestão das equipes formadas e a busca por recursos, tornando todo o projeto muito complexo e envolvendo diversas áreas do conhecimento.

Para avaliar os projetos desenvolvidos, são promovidas anualmente algumas competições (Regionais e a Nacional) onde equipes de universidades de todo o país competem em provas que testam os carros em aspectos como velocidade, durabilidade para superar obstáculos e complexidade de projeto. A competição Nacional já ocorre há mais de 25 anos e reúne cerca de 80 equipes de todos os estados brasileiros e, para as equipes mais bem colocadas, é dada a oportunidade de competir na etapa mundial que ocorre nos Estados Unidos todos os anos (SAE, 2020a).

A competição determina alguns limites de potência e segurança, e as equipes das Universidades têm liberdade para propor soluções técnicas nas áreas de carroceria, dinâmicas longitudinal e lateral, gestão e eletrônica embarcada nos veículos. Nota-se intensa atividades de pesquisa e extensão nestas modalidades, já que alguns desafios técnicos motivam a elaboração de trabalhos de conclusão de curso, projetos de iniciação científica, e inclusive projetos de pós-graduação focados na análise e desenvolvimento de melhorias para veículos do tipo Baja.

Em cada competição são realizadas diversas provas que buscam avaliar e validar a robustez e confiabilidade do veículo construído. São avaliados também quesitos de inovação e soluções técnicas nos carros, de forma que a avaliação não fica restrita somente à performance do carro na prova final, chamada de enduro.

Na UFSCar, com a criação do curso de Engenharia Mecânica no ano de 2009, foi estruturada também a equipe de Baja. Desde então, a equipe tem crescido, contando atualmente com cerca de 40 alunos de diversos cursos, principalmente das engenharias, e vem participando das competições desde 2011, alcançando bons resultados, como o 11º lugar no Regional Sudeste de 2015 e o 30º lugar no Nacional de 2016.

Dada a complexidade de um projeto como esse e a quantidade de pessoas envolvidas, a equipe é dividida em três áreas principais, chamadas de diretorias, as quais são: projeto,



Figura 3 – Protótipo M8 da equipe Baja UFSCar durante a competição Nacional de 2020.

manufatura e administração. Cada uma delas é dividida em subáreas, chamadas de pastas, facilitando a alocação de membros e definindo melhor o escopo de cada aluno. A diretoria de projeto tem como subdivisões os sistemas do carro, sendo eles compostos por cinco pastas principais: sistema de transmissão (*powertrain*), freio, suspensão e direção, cálculo estrutural e eletrônica. O protótipo atual utilizado pela equipe, chamado M8, pode ser visto na figura 3.

O sistema eletroeletrônico desenvolvido para um protótipo Baja pode ser dividido em duas partes segundo o regulamento definido pela própria SAE para as competições, sendo a primeira parte obrigatória e consiste em um sistema de segurança para o piloto e para a equipe, e a segunda parte sendo o sistema embarcado, o qual não possui requisitos mandatórios e por isso é totalmente concebido pela própria equipe (SAE, 2020b).

O sistema de segurança do projeto eletrônico de um Baja, deve ser composto por dois itens: o botão de emergência *kill-switch* e a luz de freio. Conforme diz o regulamento, o veículo deve ser equipado com duas chaves gerais de fácil acesso, e devem desativar o motor e todo o sistema eletrônico do carro, exceto a luz de freio. Elas devem ser posicionadas dentro do habitáculo do piloto ao alcance do mesmo, e também externamente na lateral direita traseira do carro. A luz de freio deve ser composta por um circuito de LEDs acionado por interruptores de pressão hidráulica instalados em todos os circuitos hidráulicos independentes das vias de freio, de forma que a luz se acenda sempre que houver pressão em um dos canais.

O sistema embarcado, por sua vez, tem uma maior liberdade de desenvolvimento, e nele os alunos têm total controle das escolhas de cada detalhe no projeto, desde o tipo de sensores e microcontroladores, até arquitetura e tipos de tecnologias empregadas. Esse sistema é basicamente composto por todos os circuitos utilizados em medições, aquisição

de dados e telemetria. Por isso, a complexidade do projeto varia entre as equipes de forma que a análise de desempenho e a aquisição de dados pode ser melhorada quanto mais se investe nesse sistema.

2.2 Eletrônica Embarcada Automotiva

A ideia de se utilizar um sistema eletrônico para medir parâmetros em tempo real exibindo diversas informações ao usuário não é nova, na verdade é essencial que haja esse tipo de sistema em todo veículo atualmente, pois foi através dele que houve uma revolução nos projetos de automóveis, melhorando significativamente o desempenho, a confiabilidade e o conforto, sendo também um marco que impactou também na fabricação de muitos componentes e na montagem dos veículos pelas montadoras (ANJOS, 2011).

O sistema embarcado tem como principal função a medição de grandezas que servem como indicadores de performance do veículo. Para tornar isso possível, tal sistema deve ser flexível e robusto, capaz de processar todos os dados captados e devolvê-los em forma de informação ao usuário, para que ele possa controlar todo o veículo conforme suas necessidades. Apesar do seu constante crescimento, atualmente já são utilizados diversos sensores e sistemas eletrônicos em uma arquitetura de sistema embarcado, como por exemplo freios inteligentes ABS (*Anti-Lock Braking System*), controle de estabilidade e controle de tração, além de sistemas mais simples como alguns sistemas de injeção eletrônica. Na figura 4 é mostrada parte da arquitetura de um sistema embarcado e como ele está instalado pelo veículo, onde podem ser vistos diferentes módulos e componentes numerados, com cada um realizando uma tarefa específica dentro do sistema eletrônico.

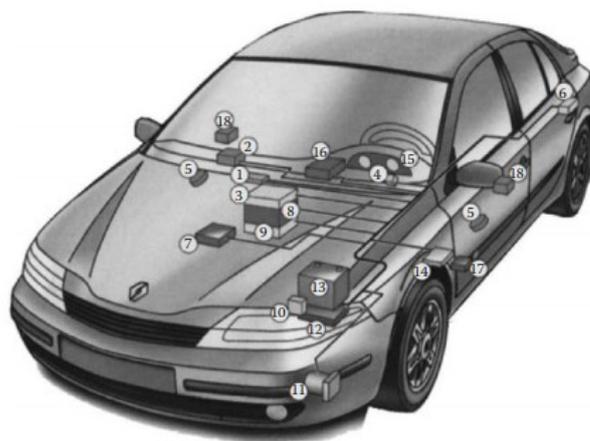


Figura 4 – Parte da arquitetura de um sistema embarcado veicular (NAVET; SIMONOT-LION, 2017).

A evolução tecnológica desse campo da engenharia, permitiu um grande aumento na capacidade de processamento dos dados, de forma que os sistemas antigos, controlados por circuitos limitados e que realizavam poucas operações, como o gerenciamento básico

das funções do motor e de ignição, passaram a se interligar com outros sistemas do carro, dependendo das centrais de processamento cada vez mais. As centrais de processamento, chamadas ECUs (*Electronic Control Unit*), como a mostrada na figura 5, foram desenvolvidas como uma forma de monitorar, controlar e permitir a realização de diagnósticos, contando com um microprocessador que utiliza todas as informações captadas pelos sensores e circuitos de aquisição de dados e faz o controle completo do sistema em que estão inseridas.



Figura 5 – Unidade de controle eletrônico Bosch para monitoramento do motor (BOSCH, 2020).

Para a realização de suas funcionalidades, as ECUs possuem programas gravados em suas memórias e dependendo dos estados de suas entradas e saídas, o *software* define as decisões que deverão ser tomadas. Esse *software* é fundamental para o funcionamento dos módulos eletrônicos e do sistema como um todo, sendo desenvolvido em três etapas durante seu processo de manufatura, iniciando pela definição do *firmware* em que o fornecedor grava as rotinas básicas do módulo, em seguida é realizada a calibração básica recebida nas montadoras como instruções mais específicas, e os parâmetros programáveis gravados no final da linha de montagem com valores determinados para o modelo do carro.

Atualmente cada veículo conta com algumas dezenas de módulos ECUs em seu sistema, de forma que os mais comuns e importantes são instalados para atuar nos seus principais componentes, como o motor (ECM) e a transmissão (TCM), sendo utilizados para o controle de estabilidade (ESC), ou até mesmo no sistema elétrico como um todo cuidando dos dispositivos secundários como ar condicionado e faróis (BCM) e das multimídias disponíveis (TCU). Esses módulos são instalados por todo o carro, geralmente próximo ao sistema em que atua, como mostrado na figura 6.

O ECM, ou *Engine Control Module*, é um dos principais módulos utilizados no sistema eletrônico, e tem o papel de atuar eletronicamente no motor e na transmissão para controlar e otimizar a quantidade de combustível injetada no motor, e os pontos de troca na transmissão durante seu funcionamento. Com as atuais leis de regulamentação de emissões de partículas lançadas na atmosfera pelos automóveis, foi também adicionada

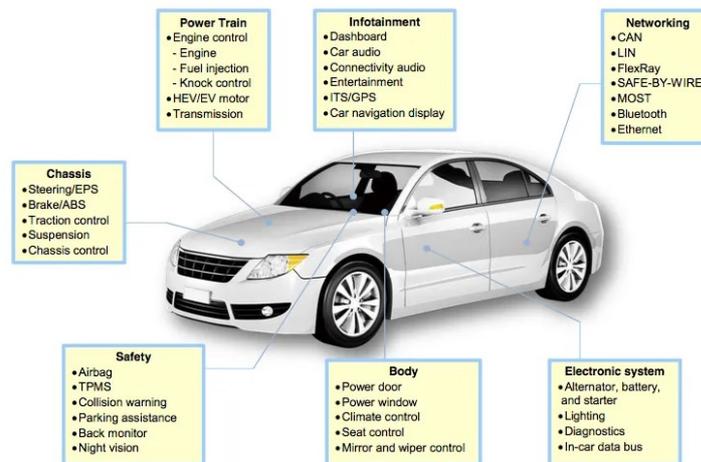


Figura 6 – Funções realizadas por ECUs em um carro (ENGINEERING, 2018).

a esse módulo a função de controlar a mistura ar/combustível de forma que o coletor catalítico seja capaz de remover uma grande parte da poluição emitida. A ECM ainda atua em outras aplicações, pois ao coletar os dados de diversos sensores, ela monitora todos os parâmetros relacionados ao motor e, além das informações de injeção de combustível e emissões, é feita a análise de temperaturas de resfriamento, mantendo total controle e o completo funcionamento do motor.

Para os veículos que contam com transmissões automáticas eletrônicas, é utilizado o módulo TCM, do inglês *Transmission Control Module*. Este utiliza diversos sensores e também informações trocadas com o ECM para calcular e definir a troca de marchas e melhorar o desempenho do carro em relação a consumo de combustível em melhores velocidades, e qualidade na troca das marchas.

O controle de estabilidade é outro item essencial nos carros e está cada vez mais presente em todos os modelos desenvolvidos. Sua função é a de auxiliar o motorista em manobras que exigem mudanças bruscas de direção do veículo, evitando acidentes ao desviar de algum obstáculo ou ao dirigir em superfícies escorregadias, fazendo com que não haja perda de controle. O *Electronic Stability Control* (ESC) é o módulo responsável por controlar a estabilidade do carro, utilizando mecanismos como o ABS e alguns outros componentes de atuação de freio eletrônico. Devido a sua precisão em manter o controle e não deixar que o veículo se perca em curvas, como mostrado na figura 7, diversos estudos indicam que esse sistema é responsável pela diminuição de acidentes fatais desde que foi implementado nos carros comerciais, e devido a isso tornou-se obrigatório seu uso juntamente com o sistema ABS em diversos países.

Para o controle da interação entre a maioria das ECUs e dos dispositivos periféricos dos carros, como sistema de ar-condicionado, faróis e travas das portas, é utilizado o módulo *Body Control Module*, BCM. Ele é também fundamental para o funcionamento geral do automóvel, pois monitora diversos sensores instalados para medir variáveis como

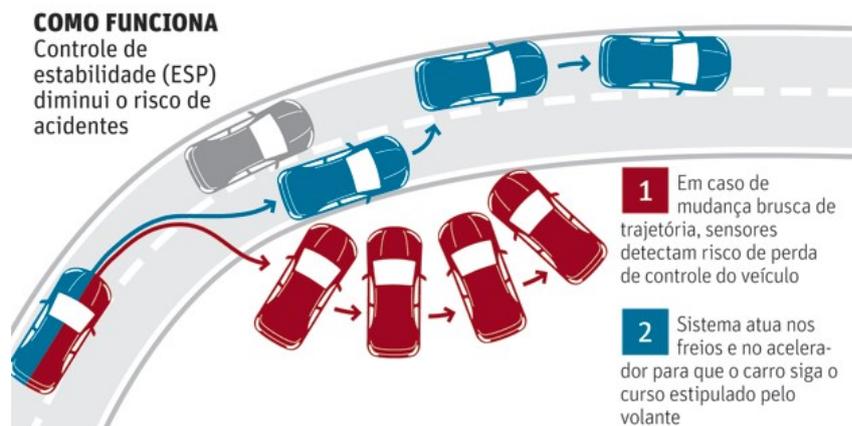


Figura 7 – Ilustração da atuação do sistema eletrônico de controle de estabilidade (PAULO, 2011).

temperaturas, velocidade ou pressões, e converte esses sinais em informações que são mostradas no painel do carro. Há um importante papel de verificação de falhas no sistema eletrônico, de forma que ao detectar algum possível erro, o BCM solicita um sinal do módulo que pode conter a falha, e caso não haja resposta, ele cria um código de erro e passa a agir para corrigir o problema.

Como uma forma de atrair mais usuários, muitos modelos de carros têm apostado cada vez mais nos sistemas de entretenimento que contam com diversos dispositivos, mas principalmente o sistema de rádio, vídeo e auxílio a navegação. Esses dispositivos são controlados pelo TCU, *Telematic Control Unit*, e ajudam a tornar longas viagens mais relaxantes e são um diferencial para a venda de veículos atualmente. É importante salientar que esses sistemas de entretenimento devem seguir as normas do Código Nacional de Trânsito, que proíbe a utilização pelo motorista de algumas funcionalidades enquanto dirige, como assistir televisão, por exemplo.

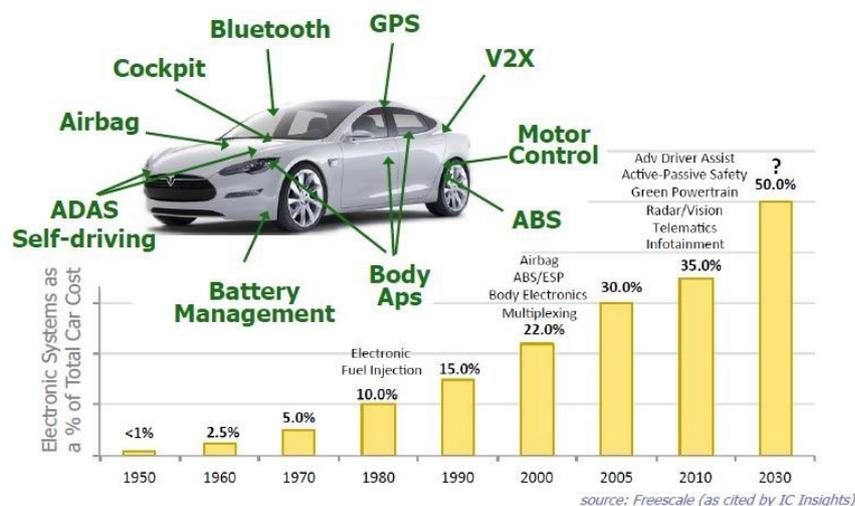


Figura 8 – Crescimento na utilização dos sistemas embarcados em carros (SOLUTIONS, 2018).

Assim, o uso de componentes eletrônicos cresce cada vez mais devido à todas as rápidas evoluções desse mercado, desde a implementação dos dispositivos de otimização de performance e de segurança, que podem ser citados como exemplo a injeção eletrônica e o freio ABS. Os sistemas eletrônicos embarcados vêm tomando cada vez mais espaço nesse setor, como mostrado na figura 8, estima-se que em 2030 os custos de utilização destes sistemas sejam equivalentes a 50% do custo de produção dos carros, dada a sua importância e ampla difusão entre as montadoras.

2.3 Protocolos de Comunicação

Em um sistema que conta com diferentes módulos operando em conjunto e trocando informações entre si, é importante que haja uma maior confiabilidade nos dados e robustez no sistema para que ele seja capaz de fazer todas as trocas de mensagens com menor quantidade de erros possível e da melhor maneira. Com essa necessidade, a ISO (Organização Mundial de Padrões) desenvolveu um modelo que padronizasse todo tipo de comunicação em equipamentos eletrônicos, o que envolve desde o tipo de meio em que essa comunicação é feita, até níveis de sinais elétricos em que a informação é enviada. O modelo criado pela ISO, chamado de *Open Systems Interconnection* (OSI), é a referência para a criação e utilização de todo tipo de protocolo de comunicação existente, e ele define até sete camadas ou níveis, como ilustrado na figura 9, onde a camada mais baixa corresponde ao *hardware* e as camadas sucessivas correspondem ao *firmware* que o utiliza. Porém não é necessário que um sistema implemente todas as sete camadas, podendo deixar camadas vazias.



Figura 9 – Camadas definidas pelo modelo OSI para protocolos de comunicação (DLTEC, 2011).

Essa divisão permite a análise mais facilitada de cada parte, a padronização de componentes que permita o desenvolvimento e suporte por parte de vários fabricantes diferentes e principalmente evitar que a modificação de uma das camadas afete as outras,

o que facilita o seu desenvolvimento (LOPES; BENEVENUTO; OLIVEIRA, 2015). Assim, as funções de cada camada são:

1. **Física:** A camada física tem como função determinar como será realizada a transmissão dos dados através de um canal de comunicação, ou seja, definindo questões físicas como nível de tensão, duração dos sinais e meio físico. Mas também trata de características mecânicas, elétricas e de operação física, como conectores, linha física, cabos, conexões de modems, sentido do fluxo de dados, entre outras coisas;
2. **Enlace:** Essa segunda camada tem como objetivo garantir a formatação correta dos dados que serão transferidos entre os módulos. Para isso, o enlace recebe os dados e adiciona informações de controle, como endereços de origem e destino, realizando um gerenciamento de erros e controlando o fluxo dos dados;
3. **Rede:** Durante o nível de rede ocorre um monitoramento do tráfego de dados pela rede externa. É ela quem define o tráfego físico dos dados conforme as condições em que se encontra a rede, a prioridade da mensagem, entre outros fatores. Por isso, esse nível tem o controle sobre a informação enviada, podendo transformar ela em pacotes menores caso isso seja delimitado pelas limitações do protocolo, e remonta-la na chegada ao destino;
4. **Transporte:** A camada de transporte faz a ligação entre os dois grupos de camadas, os níveis de aplicação que são 5 a 7, e de nível físico que são de 1 a 3. Aqui é feita a função de determinar a classe de serviço realizado, como orientado à conexão, contando com o controle de erro e um serviço de confirmação, ou sem conexão que apresenta baixa confiabilidade;
5. **Sessão:** Essa camada é responsável por estabelecer uma sessão de envio entre módulos *hosts*, definindo como será feita a transmissão e realizando marcações nos dados que serão transmitidos. Assim, caso haja alguma interrupção na rede e a sessão não tenha realizado a transmissão completa dos dados, é possível recomeçar a transmissão dos dados a partir da última marcação recebida pelo transmissor;
6. **Apresentação:** Aqui são realizados processos muito importantes para a correta transferência dos dados, pois essa camada tem a função de converter o formato da mensagem que será enviada em um formato padrão do protocolo. Além disso, pode ser feita a compressão dos dados e também a criptografia dos mesmos, o que é considerado um processo importante dentro dos protocolos, por garantir a segurança da informação contra invasores da rede ou até da quebra de pacotes para a extração de informações;
7. **Aplicação:** A camada mais alta deste modelo é responsável pela interação com o usuário. São parte desta camada os aplicativos que têm acesso a rede e que são usados

para a visualização das informações, os quais têm o papel de sintetizar os dados de forma clara e mostra-los para que possam ser usados pelo solicitante. Aplicativos padrões que fazem parte dessa camada que podem ser citados são os *browsers* que utilizam como protocolo o HTTP que faz o contato com o servidor e apresenta ao usuário tudo de forma clara, conforme programado para tal.

Com a utilização do padrão de referência OSI, cada camada é muito bem definida e adiciona algum dado à informação que se necessita enviar, e isso permite que a geração de dados que a princípio é complexa, se torne mais simples ao usuário ao mesmo tempo que tornam todo esse processo muito mais rigoroso, seguro e robusto.

2.4 Aquisição e Análise de Dados

No âmbito competitivo, a aquisição e análise de dados é uma área amplamente explorada, pois ela permite acompanhar em tempo real a performance do carro e utilizar estes dados coletados para definir melhorias de projeto, como ocorre nas diversas categorias de competições do automobilismo. Devido ao grande volume de dados, existem diversas técnicas de visualização e análise, usualmente se utilizando de gráficos ou de amostras de valores em intervalos de tempos. Um exemplo dessa visualização é mostrado na figura 10, em que é possível verificar dados de performance tanto do carro quanto do piloto, mostrando velocidade, posições do pedal do acelerador, marcha e frenagem em cada ponto da pista.

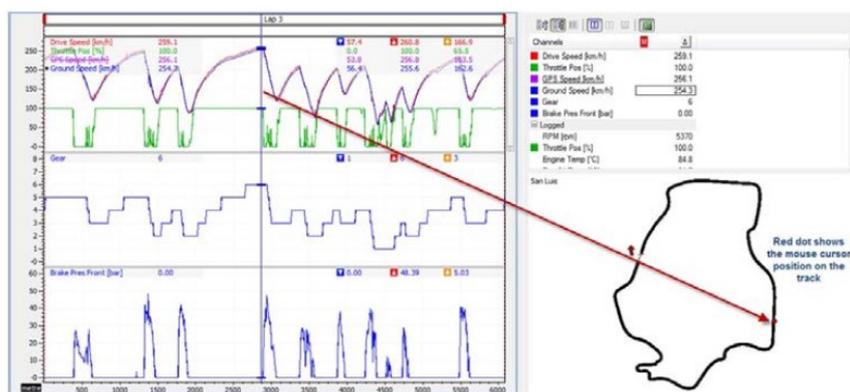


Figura 10 – Interface de um sistema de aquisição de dados de um carro de corridas (SERGERS, 2014).

E com essa grande quantidade de dados que é possível coletar, também há diferentes abordagens de análises, dependendo de qual aspecto se queira melhorar da performance. Temos então algumas categorias em que podem ser divididas, sendo elas: Análise de Performance do Veículo, Análise de Performance do Piloto, Desenvolvimento do Veículo, Confiabilidade e Segurança, Determinação de Parâmetros do Veículo e Registros de Corridas (SERGERS, 2014).

Para a análise de performance do veículo são usados dados coletados juntamente com comentários realizados pelo piloto, para que se consiga uma avaliação de como ele está se comportando. É possível a realização de análises precisas de diversos fatores do carro, nos pontos e tempos em que ocorreram na pista, para que o engenheiro possa definir alterações e melhorias que devam ser empregadas no carro.

A análise de performance do piloto é bastante útil em campeonatos para avaliar o estilo de direção e comparação entre diferentes pilotos de uma mesma equipe, além de permitir um *feedback* ao piloto para que ele possa aperfeiçoar suas habilidades em determinadas situações durante as corridas. São registradas as atividades de direção em cada volta, podendo ser medidas diversas variáveis como posições dos pedais, tempos de trocas das marchas, frenagem, entre outras coisas.

Outro fator muito importante para o uso da aquisição de dados é por ela ser utilizada como uma ótima ferramenta para a melhoria no desenvolvimento do veículo. As medições auxiliam em tomadas de decisões que definem os focos de desenvolvimento, melhorias e otimizações.

Com o objetivo de aumentar a confiabilidade e segurança do carro, são medidos parâmetros vitais, como pressão do óleo e temperaturas de diversos locais, além de medições de tensão da bateria e se algum sistema está em mal funcionamento. As falhas podem ser determinadas antes de ocorrerem ou nos primeiros momentos após, de forma que os danos são mitigados e acidentes podem ser evitados.

A criação de um modelo de simulação através da determinação de parâmetros do veículo é uma ferramenta bastante útil, permitindo que *softwares* de simulação da carros de corrida sejam desenvolvidos, sendo estes auxiliares importantes para a criação de sistemas de registros de dados. São necessários diversos dados que podem ser medidos dentro e fora das corridas e até mesmo calculados durante o projeto.

Todos os dados medidos são também gravados em registros de corridas, para que haja um histórico de performance do carro e sobre o que ocorre durante todo o tempo de uso. Com essas informações juntas com outros dados vitais, é possível determinar a vida útil das peças e dizer quando elas devem ser trocadas ou até mesmo da utilização de novas peças para o seu lugar.

As possibilidades desse tipo de sistema em um carro são diversas, sendo limitadas pelo tempo e orçamento envolvidos, por isso a aquisição de dados é usada em tantas aplicações diferentes. A configuração de medidas para um sistema inicial consiste na obtenção dos parâmetros mais básicos do veículo, permitindo a análise de chassi e de direção, sendo essas medidas: rotação do motor, velocidade das rodas, posição de câmbio, ângulo de esterçamento, e acelerações lateral e longitudinal.

3 Eletrônica Embarcada no Projeto Baja

3.1 Sistemas de Medição

O desenvolvimento do sistema de aquisição de dados é a base de todo projeto de eletrônica embarcada, por isso a sua concepção e definição de todos os parâmetros que devem ser medidos é de extrema importância na sua etapa inicial.

Devido às demandas da equipe e das provas apresentadas nas competições, foram escolhidas para serem medidas as seguintes informações do veículo: velocidade, rotação do motor, temperaturas da caixa de redução e da CVT, nível de combustível, nível da bateria e localização através de GPS. Para cada uma dessas medições foi realizado um projeto de *hardware* e *software*, além da escolha do sensor utilizado.

Cada um desses sensores é focado na análise de performance do veículo e no seu desenvolvimento, por estes serem os aspectos mais importantes observados pela equipe e que são possibilitados pelos componentes utilizados. A performance do piloto não é considerada pela rotação de membros nessa função e pela constante mudança nos obstáculos enfrentados, tanto nos testes da equipe quanto durante as competições.

3.1.1 Velocidade

A velocidade do veículo é uma das principais informações que devem ser adquiridas, pois ela auxilia no aumento da performance da direção, sendo de extrema importância para o piloto saber a velocidade do carro, de forma que ele possa se orientar durante alguma prova de velocidade ou do enduro, além de auxiliar na escolha de soluções para os projetos dos sistemas mecânicos do veículo, como por exemplo no caso do *powertrain*, onde a velocidade é um dos parâmetros utilizados para o seu projeto.

Para tal medição foram utilizados dois sensores indutivos, que se destacam pela sua robustez, alta velocidade de comutação e facilidade de uso. Este tipo de sensor é um dispositivo eletrônico capaz de reagir a proximidade de objetos metálicos e é composto de um oscilador e uma bobina que geram um campo magnético na sua extremidade. Na figura 11 há uma imagem de um sensor indutivo e uma ilustração de seu esquema de funcionamento.

Para o seu funcionamento, o sensor indutivo possui um circuito que gera um campo magnético na sua ponta, que induz pequenas correntes parasitas ao sofrer alguma interação com um material metálico. Com a condução do metal, há uma diminuição na energia do campo e, em consequência, na amplitude do sinal gerado por um oscilador interno.



Figura 11 – Sensor indutivo, em detalhe uma ilustração de seu esquema de funcionamento.

Quando este sinal se torna muito baixo, um circuito disparador percebe a mudança e muda a tensão de saída do sensor.

Estes sensores foram posicionados próximos aos discos de freio das rodas dianteiras para que a medição pudesse ser feita da melhor forma possível. Essa configuração foi adotada buscando uma maior confiabilidade nesse tipo de medição, possibilitando adotar a velocidade relativa em função das duas rodas mesmo em situações de menos precisão como por exemplo durante as curvas, onde cada roda percorre um caminho diferente, fazendo com que a velocidade medida em função da roda se altere. Na figura 12 é mostrada a montagem do sensor no carro, onde é possível verificar que o objeto escolhido para a medição foi o disco de freio, que gira junto com a roda e conta com uma geometria que possibilita tal medição.



Figura 12 – Montagem do sensor indutivo em uma das rodas do carro.

Com a leitura destes sensores, é registrado o tempo entre os pulsos enviados e então é calculada a velocidade angular das rodas, que é utilizada em seguida no cálculo da velocidade do carro.

3.1.2 Rotação do motor

A rotação do motor é uma das principais características de seu funcionamento e, conhecendo-a é possível realizar sua calibração para que sua rotação permaneça em uma faixa de valores mais adequada para uma maior eficiência.

O motor utilizado pela equipe tem um ímã em seu volante, que é a sua parte responsável por transmitir o torque obtido pelo virabrequim para a caixa de velocidades de forma que ajude a manter o equilíbrio e reduzir vibrações no motor e, por isso, ele se mantém girando com a mesma rotação. E por ele ser uma parte de fácil acesso, se torna uma boa opção para utilizar na leitura da sua rotação.

Por isso, foi escolhido para essa medição o sensor de efeito *Hall*, o qual pode interagir com o campo magnético gerado pelo ímã do volante para se obter a rotação do motor. Basicamente, o sensor de efeito *Hall* é um transdutor que, ao se aplicar um campo magnético próximo a ele, há uma resposta na forma de uma variação em sua tensão de saída. Na figura 13 ele pode ser visto, bem como sua configuração de pinagem e sua representação eletrônica.

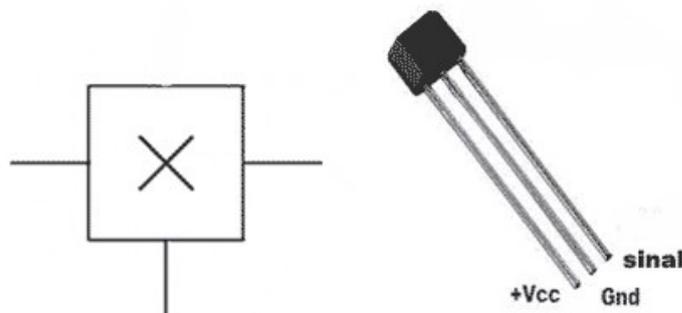


Figura 13 – Imagem do sensor de efeito Hall, indicando sua pinagem e sua representação em esquemáticos eletrônicos (MICROSYSTEMS, 2016).

Ele foi escolhido devido a sua durabilidade por ser construído em estado sólido e a sua velocidade de comutação que é muito alta, o que é satisfatório para o projeto e permite ler a rotação do motor com muita facilidade e precisão. Houve também uma grande facilidade na instalação deste sensor no motor devido ao seu tamanho reduzido.

3.1.3 Temperaturas da CVT e da caixa de redução

Em certos componentes do carro, a exposição por um longo período a temperaturas elevadas pode causar desde desgaste precoce de peças até possibilidades de incêndio. Por isso, há também uma grande necessidade de se monitorar certas temperaturas, buscando prever o funcionamento inadequado ou não otimizado de peças ou componentes que trabalhem melhor em uma certa faixa de temperaturas.

Assim, é realizado o aferimento das temperaturas do óleo da caixa de engrenagens e do ambiente de proteção da CVT. Para isso, são utilizados sensores LM35 instalados nas paredes das caixas de proteção dos componentes citados. Essa leitura é importante também para o projeto destes componentes, pois o óleo utilizado na caixa de redução é escolhido com base em alguns fatores, sendo um deles a temperatura de trabalho. Da mesma forma, a correia utilizada na CVT possui também uma temperatura ideal de trabalho, e por isso, sabendo-se a temperatura que pode chegar o ambiente interno de sua proteção, há a possibilidade de decidir se é necessário um melhor arrefecimento de tal ambiente para aumentar a vida útil da correia.

O sensor LM35 é um sensor de temperatura cuja tensão de saída é linearmente proporcional à temperatura em graus Celsius sendo, portanto, sua tensão de saída equivalente à temperatura medida naquele momento por ele. Este componente não necessita de nenhuma calibração externa ou ajuste e sua precisão é de $0,25^{\circ}\text{C}$.

Aliado a isso, este sensor tem um custo baixo, dimensões reduzidas e fácil utilização, necessitando de um conversor A/D, para a leitura do seu sinal. Sua faixa de leitura vai de -55°C a 150°C , valores que estão dentro da faixa de trabalho dos componentes monitorados. Na figura 14 pode ser vista a imagem do sensor LM35.

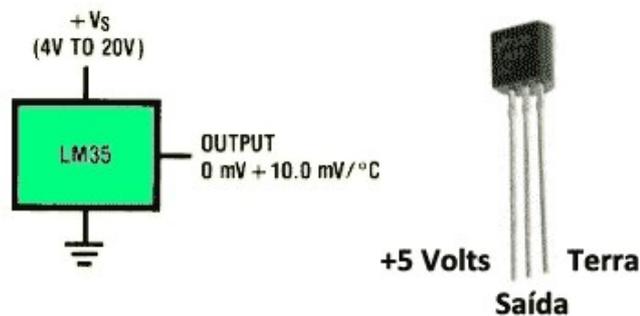


Figura 14 – Imagem do sensor LM35, indicando sua pinagem e sua representação em esquemáticos eletrônicos (INSTRUMENTS, 2017).

Nesse sistema, dois sensores são ligados a portas analógicas do microcontrolador. Como o sinal enviado pelos sensores é linear e uniforme, é lido em sua saída um valor que poderá variar entre $-0,55\text{ V}$ e $1,5\text{ V}$, sendo que cada grau Celsius de temperatura faz esse valor de tensão elétrica do sinal variar 10 mV .

Ao receber o sinal analógico, é feita uma conversão para um valor digital pelo seu conversor A/D, fazendo com que o sinal seja apresentado em *bits*, com um número equivalente à precisão deste. Por isso, para saber qual é a temperatura medida pelo sensor são necessárias duas conversões, primeiro convertendo o sinal digital entregue pelo A/D para analógico, e depois convertendo esse sinal analógico para a sua temperatura correspondente. Considerando uma tensão máxima de referência do conversor de 5 V , utiliza-se o seguinte procedimento:

$SinalDigital \rightarrow SinalAnalógico \rightarrow Temperatura$

$$Analógico = \frac{DigitalLido \times 5}{DigitalMax} \quad (3.1)$$

$$Temperatura = \frac{Analógico \times 150}{1,5} \quad (3.2)$$

Para uma maior confiabilidade desta leitura, a mesma é realizada algumas vezes, com um pequeno *delay* entre cada leitura e uma média desses valores lidos é computada. Essa abordagem é utilizada devido a possíveis imprecisões geradas pelo próprio conversor A/D interno do microcontrolador e pela velocidade reduzida em que a leitura deve ser realizada durante o funcionamento do sistema.

3.1.4 Módulo GPS

A inclusão de um módulo GPS ao projeto traz algumas vantagens, pois com as informações obtidas através dele é possível saber a rota do veículo e, com isso, definir o traçado das pistas e suas reais extensões em testes ou nas competições. Além disso, a contagem de voltas realizada pela equipe pode ser feita de forma automática e até mesmo, com o valor da velocidade adquirido por esse módulo, auxiliar na definição da velocidade instantânea do carro.

O módulo utilizado para isso foi o GPS NEO-6M, mostrado na figura 15. Com ele foi possível se obter a posição e a hora exata em que o veículo passou em cada ponto da trajetória, além disso, esse módulo fornece a altitude e a velocidade medida, permitindo a sua utilização em diversas aplicações.

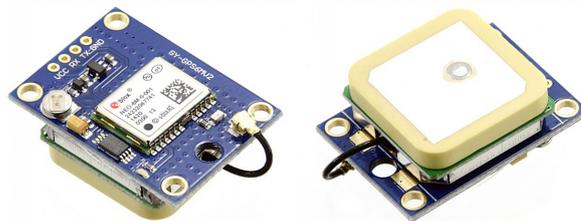


Figura 15 – Módulo GPS utilizado no projeto.

Este módulo opera com o protocolo de comunicação NMEA-183 (*National Marine Electronics Association*), o qual é utilizado como padrão de comunicação entre dispositivos náuticos. Toda mensagem nesse tipo de protocolo começa com um “\$” (cifrão), em seguida todos os dados separados por vírgula, e por fim termina com um caractere terminador de linha “\n”. Um exemplo de mensagem desse protocolo é:

\$GPGGA,104549.04,2447.2038,N,12100.4990,E,1,06,01.7,00078.8,M,0016.3,M,,*5C

Onde:

- GPGGA – GP é sigla para GPS e GGA é uma sigla para sistema de posicionamento global com dados fixos;
- 104549.04 – Tempo universal no formato HHMMSS.SS;
- 2447.2038 – Latitude, 24 graus e 47.2038 minutos;
- N – Indicador de hemisfério latitudinal, com N para norte e S para sul;
- 12100.4990 – Longitude, 121 graus e 00.4990 minutos;
- E – Indicador longitudinal, com E para leste e W para oeste;
- 1 – Indicador de correção para a qualidade da posição, sendo 0 para posição inválida, 1 para posição válida no modo SPS e 2 para posição válida no modo GPS diferencial;
- 06 – Número de satélites usados, de 0 a 12;
- 01.7 – Diluição horizontal da precisão, de 0.0 até 99.9;
- 00078.8,M – Altitude em metros em relação ao nível do mar, de -9999.9 até 17999.9;
- 0016.3,M – Altitude geoidal em metros, de -999.9 a 9999.9;
- *5C – Valor do checksum, número hexadecimal iniciado por “*”.

Em um teste inicial para verificar o funcionamento do receptor GPS, foi feita sua gravação em um cartão de memória e realizada a integração destes ao *software Google Earth*. Para isso, foi necessário converter os dados gravados no cartão SD em um arquivo texto (.txt), para a extensão NMEA. Dessa forma, pôde-se obter a rota completa do módulo e o instante de tempo em que ele passou em cada ponto, como é mostrado na figura 16.

É importante observar também, que como pode ser visto na imagem, houve momentos em que o GPS perdeu a conexão com o satélite, deixando pontos mais espaçados entre si. Isso acontece pela baixa frequência de dados fornecidos pelo módulo. Contudo, a rota não sofreu alterações drásticas e, por isso, não há grandes problemas na utilização desse sistema. Com um ajuste no software para corrigir a perda de comunicação, essa falha é corrigida e todo o traçado é lido sem maiores problemas.

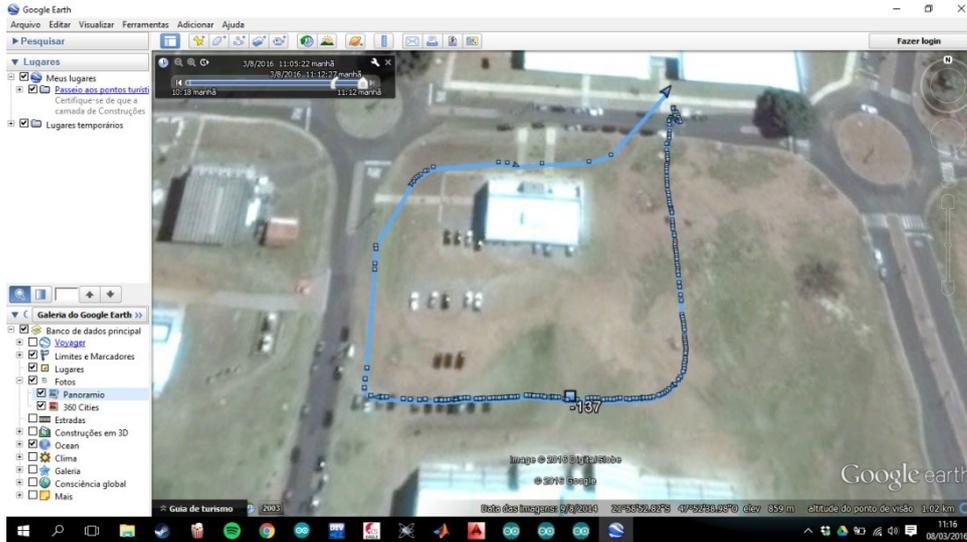


Figura 16 – Rota gerada durante os testes com o módulo GPS.

3.1.5 Nível de combustível

Conhecer o nível de combustível durante as competições é de grande importância para o projeto, principalmente durante as provas de enduro, onde o carro permanece na pista de 3 a 4 horas seguidas e só é permitido o reabastecimento no *box* e, caso o combustível acabe no meio da pista, o carro deve deixar a prova. Porém, o regulamento para esse tipo de sistema é muito restritivo por se tratar de um projeto que deve introduzir algum sensor ou sistema eletrônico próximo ou até mesmo dentro do tanque de combustível e, por isso, há a imposição de que não deve haver nenhum tipo de furo no tanque para tal medição (SAE, 2020b).

Dadas essas restrições, a solução encontrada foi a de utilizar uma célula de carga, como a vista na figura 17a, para obter a medição da massa total do tanque de forma que, sabendo a massa do mesmo vazio, é possível determinar a massa do seu conteúdo interno. Porém, o grande problema em utilizar esse tipo de medida é o fato de que o carro estará em constante movimento e a inércia do combustível dentro do tanque afeta o valor lido pela célula de carga. Dessa forma, para viabilizar a utilização deste método, foi adicionado também a leitura de um acelerômetro que permita definir momentos de menor aceleração do carro onde os dados desse sistema podem ser considerados confiáveis o suficiente para a sua utilização.

Com isso, inicialmente foram feitos testes com a célula de carga para definir sua curva de leitura. Em geral, as células de carga são montadas como uma ponte de resistores, onde cada resistor é um *strain gauge*, que varia sua resistência de saída conforme sua deformação (figura 17c). Como a saída de tensão é baixa, tipicamente é necessário o uso de um amplificador operacional com ganho de algumas centenas de vezes. O amplificador operacional escolhido foi o INA125P, pois ele atende aos requisitos de amplificação do

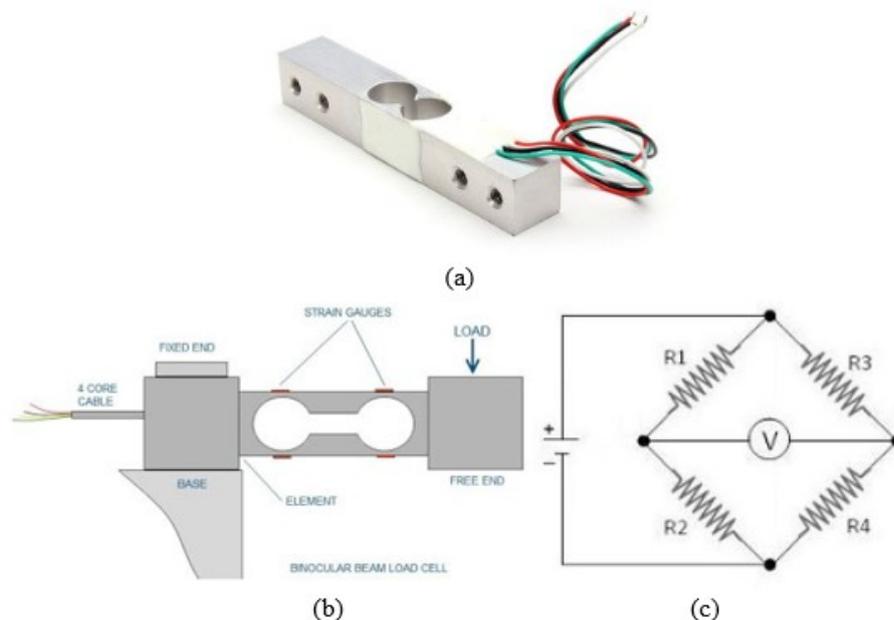


Figura 17 – Célula de carga (a), posicionamento (b) e esquema de ligação dos strain gauges que a compõem (c). Adaptado de (WEIGHING, 2010).

projeto e possui fonte de tensão de referência ajustável de alta precisão e baixo ruído.

Assim, o circuito utilizado para o teste pode ser visto na figura 18 e o resistor de ajuste de ganho (RG) escolhido foi de 120Ω , determinando um ganho de 500 vezes.

Em seguida, foram utilizadas massas de pesos conhecidos no teste, onde o valor lido era então passado para uma tabela que, em seguida, permitiu a criação de um gráfico de valor enviado pelo conversor A/D da porta analógica *versus* massa para verificar o comportamento da célula de carga. Esse gráfico foi feito utilizando o *software Excel* e pode ser visto na figura 19. Através desse gráfico gerado e com uma regressão linear, pode-se obter a curva que descreve o comportamento da célula de carga e permite fazer a conversão do sinal lido em massa.

Após esse teste, foi feito também um novo teste com um acelerômetro para que posteriormente possa ser implementada a junção de ambos para compor o sistema de medição do nível de combustível.

No projeto foi utilizada uma unidade inercial ITG/MPU GY-521, mostrada na figura 20, a qual faz uso do circuito integrado MPU-6050 da InvenSense que combina um acelerômetro de 3 eixos e um giroscópio de 3 eixos com um potente processador digital de movimentos (DMP), permitindo assim obter com alta precisão, as acelerações e os ângulos de giro em todos os eixos (rolagem, guinada e arfagem) (INVENSENSE, 2013). Além disso, o tamanho dessa unidade, é muito menor que dos acelerômetros e giroscópios convencionais, tornando seu uso extremamente facilitado.

No teste realizado com este acelerômetro foi possível medir as acelerações nos 3

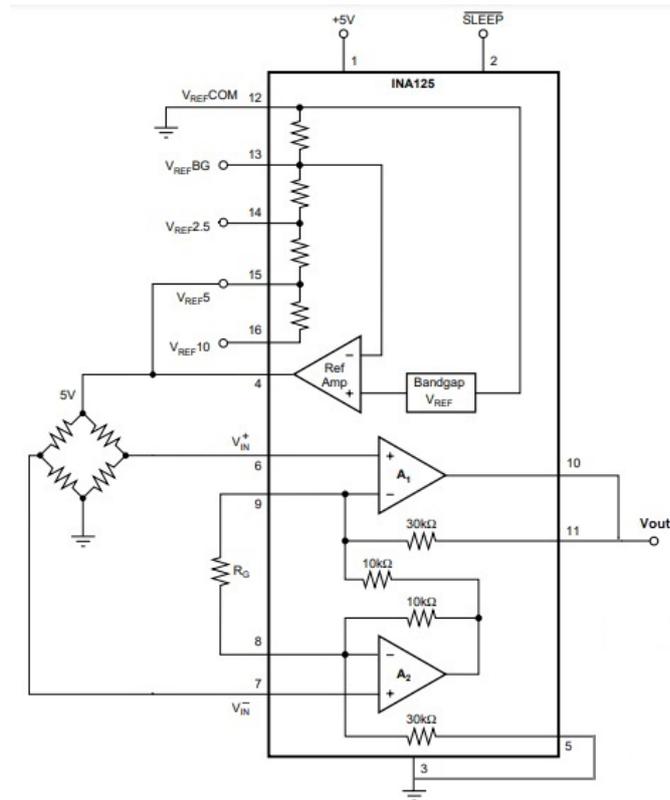


Figura 18 – Circuito utilizado para o projeto com a célula de carga (BROWN, 1998).

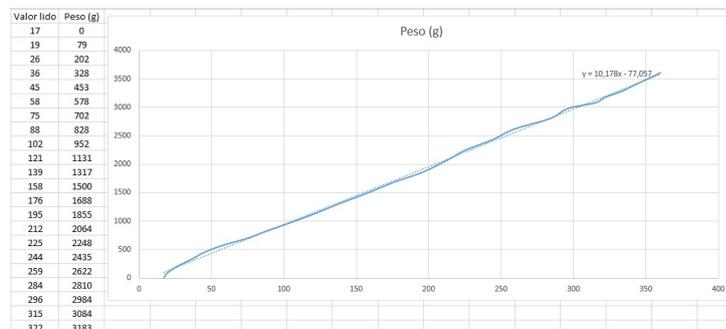


Figura 19 – Gráfico gerado da relação entre a tensão elétrica medida e massas utilizadas.

eixos, que seria o necessário para o projeto. Em seguida, foi realizado um teste com a junção de ambos, de forma que a leitura da célula de carga é feita apenas em intervalos de aceleração quando seu valor se encontra próximo de zero.

Para que ocorressem tais testes, foi produzido um módulo que pudesse ser instalado no carro com o seu circuito eletrônico e formato iguais aos das placas de leitura de sensores padrão, porém com apenas entrada para a célula de carga e com o INA125P. Essa placa pode ser vista na figura 21 e será utilizada também durante as competições para a utilização deste sistema.

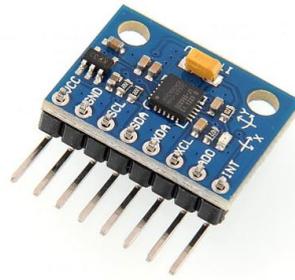


Figura 20 – Unidade inercial utilizada nos testes (ARDUINO, 2019).

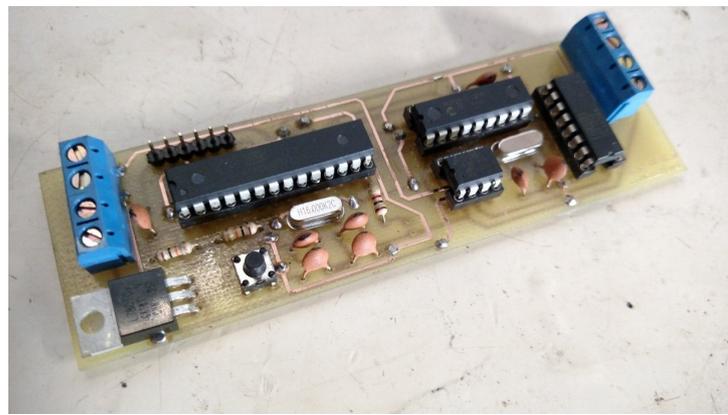


Figura 21 – Módulo construído para a leitura do nível de combustível.

3.1.6 Carga da bateria

A bateria utilizada no projeto é um item regulamentado pela SAE, tendo algumas de suas características controladas. No caso da capacidade da bateria, não é permitido que seja superior a 240Wh. Como as baterias utilizadas são por padrão de 12V de tensão, significa que a capacidade não pode ultrapassar os 20Ah. Além disso, seu funcionamento deve ser garantido para que a luz de freio que deve ser ligada a ela, esteja operante durante toda a competição.

Sendo assim, para garantir que sua carga não chegue a zero é necessário também o uso de um sistema de medição que informe o nível da bateria e permita que ela seja carregada em um intervalo entre as provas ou até mesmo trocada por outra com a carga completa, sem que isso interfira no funcionamento completo de todos os sistemas alimentados por ela no carro.

Para sanar esse problema, foi adicionado um circuito divisor de tensão simples ligado a uma das portas analógicas de um dos microcontroladores, como mostrado na figura 22, o qual é responsável pela leitura da tensão da bateria para estimar sua carga e alertar caso esta fique baixa. Essa solução não é a mais indicada, pois consome uma quantidade de energia acima do necessário, porém foi implementada pela sua facilidade e

deve ser modificada por algum outro sistema mais sofisticado e que seja mais adequado em próximas melhorias.

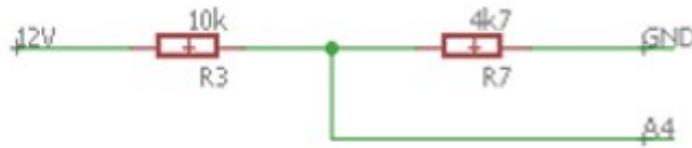


Figura 22 – Circuito divisor de tensão para leitura da carga da bateria.

3.2 Processamento dos dados

Uma escolha importante em um projeto de eletrônica embarcada é o tipo de microcontrolador que deverá ser utilizado, pois ele deve ser capaz de fazer a leitura de diferentes tipos de sensores, contar com alguns periféricos e ter uma taxa de processamento satisfatória para a aplicação proposta. Existem diversas opções no mercado, que possuem um amplo número de recursos e que já são utilizados em projetos de diferentes complexidades, porém devem ser considerados outros aspectos importantes, como o custo, tipo de linguagem de programação e sua complexidade, dificuldade na implementação e utilização.

Para este projeto foram então definidas que as principais características que seriam consideradas para a escolha do microcontrolador seria a disponibilidade de portas de entrada de sinal digital, conversor A/D interno para uso de portas analógicas, diferentes protocolos de comunicação, obrigatoriamente com Serial e I2C, no mínimo, e que aspectos como custo e disponibilidade no mercado fossem um grande diferencial.

Depois destas definições, foram avaliadas três opções de microcontroladores muito utilizados e que possuem características totalmente satisfatórias para o projeto, os quais seriam: PIC da Microchip, as placas da família Arduino, e do tipo ARM da família STM32.

3.2.1 PIC

PICs são microcontroladores populares, e utilizados tanto industrialmente quanto em projetos informais devido à sua extensa lista de benefícios, como baixo custo, grande disponibilidade e base de usuários, além da disponibilidade de ferramentas de desenvolvimento e aplicação facilitados. Mas mesmo com todos estes aspectos positivos, seu principal ponto forte é sua quantidade de modelos e famílias, como visto na figura 23, com uma variedade de diferentes características, que dá ao projetista uma liberdade de escolha entre quaisquer que sejam suas necessidades, podendo optar por várias configurações de *chip* diferentes, em detrimento da aplicação e do custo.

Em geral, para o seu uso, é necessário um dispositivo externo que permite a gravação de um código de *firmware* para que ele realize as funções desejadas. Este dispositivo,

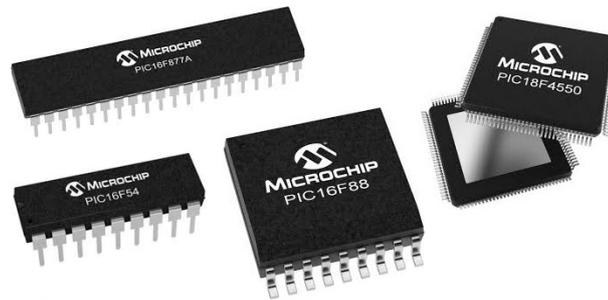


Figura 23 – Diferentes modelos de PIC disponíveis no mercado (MICROCHIP, 2020).

mostrado na figura 24 é chamado de Programador, o qual consiste em uma placa com um circuito que faz a interface entre o PIC e um computador, para a transferência do código.



Figura 24 – Modelo gravador programador de PIC (AUTOCORE, 2020).

Existem duas linguagens de programação que podem ser utilizadas para programar um PIC: Assembly e C/C++, com a segunda sendo introduzida posteriormente e contando com um nível de linguagem mais alto, muito mais amigável para aplicações mais complexas, como uso de protocolos de comunicação, como I2C (SANUSB, 2009).

Dada a aplicação, foi selecionado o modelo PIC 18F25K83 como candidato para possível uso, que tem uma frequência de operação máxima de 64 MHz, conta com 25 portas de I/O, sendo 24 canais ADC de 12 bits e comunicações UART, I2C, SPI, além dos protocolos LIN e CAN que são muito usados em aplicações automotivas (MICROCHIP, 2017).

3.2.2 Arduino

Arduino é uma plataforma criada em 2005 por alguns pesquisadores na Itália e que tinha como principais objetivos a criação de um componente que fosse barato, funcional e fácil de programar. O resultado foi uma variedade de placas, como as que são mostradas na figura 25, voltadas ao público de estudantes e projetistas amadores e que se utilizam de um conceito de *hardware* livre com um microcontrolador que pode ser programado de forma facilitada através de uma IDE própria e em uma linguagem baseada em C/C++,

sem a necessidade de se utilizar qualquer outro circuito externo ou equipamento além de um cabo USB (THOMSEN, 2014).

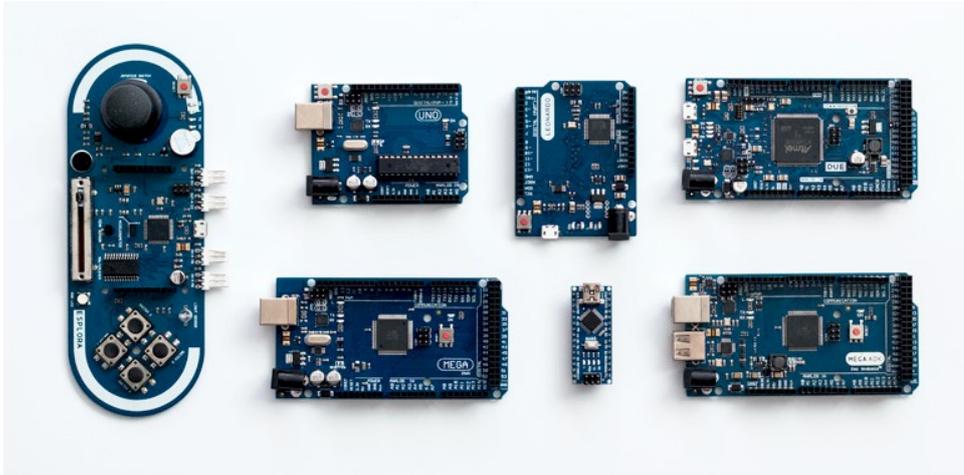


Figura 25 – Algumas das diferentes placas Arduino (THOMSEN, 2014).

Apesar de suas especificações não serem tão completas quanto as outras opções consideradas para esse projeto, o Arduino compensa com a sua facilidade de utilização, disponibilidade no mercado e variedade de periféricos externos, melhorando ainda mais sua usabilidade. Se tratando de *hardware*, seu principal microcontrolador, o ATmega328, com seu esquemático de portas mostrado na figura 26, conta com 14 portas digitais, sendo que 6 delas com configuração para saída PWM, 6 portas analógicas (que também podem ser utilizadas como digitais) com seu ADC de 10 bits, além de portas de comunicação Serial, I2C e SPI, dentre outras funções (CAMPOS, 2015).

Por ter sido projetado com o objetivo de ser um componente de fácil acesso, o Arduino reúne uma comunidade de usuários que é bastante extensa e que conta com o compartilhamento de diversos projetos e bibliotecas desenvolvidos para várias aplicações diferentes. É possível também o uso de *Shields*, que são pequenas placas criadas como módulos de *hardware*, especificamente para expandir ou até mesmo ampliar as suas capacidades.

Para essa aplicação, não foi escolhida uma placa específica da família Arduino, mas sim o controlador ATmega328, usado em alguns dos principais modelos como o UNO e o NANO. Seu uso em “*Standalone*” requer apenas a adição de alguns componentes básicos, como resistores, capacitores e um cristal de frequência, porém permite o seu uso com as mesmas capacidades de uma placa comercial, além de permitir uma maior flexibilidade no projeto.

3.2.3 ARM

A arquitetura ARM foi desenvolvida com o objetivo de diminuir o seu consumo de energia, com menor custo e com uma menor dissipação de calor em comparação

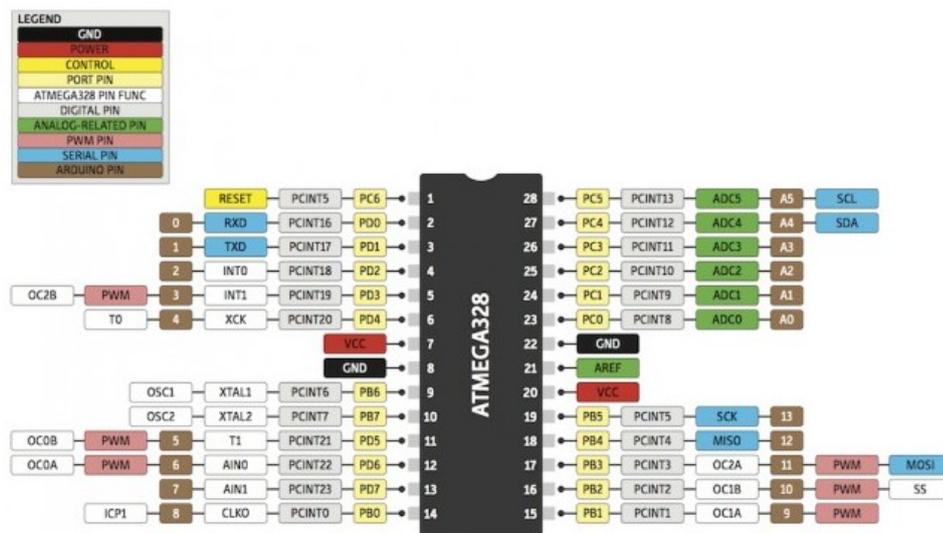


Figura 26 – Esquemático de portas do microcontrolador ATmega328 (CAMPOS, 2015).

aos microcontroladores existentes, sendo por isso, muito desejado para aplicações em dispositivos pequenos, portáteis e alimentados por bateria como os diversos sistemas embarcados. Este conceito ARM se tornou tão popular que em 2005 por volta de 95% dos celulares comercializados utilizavam pelo menos um processador ARM, alcançando até mesmo a marca de 10 bilhões de unidades produzidas, devido ao seu ótimo custo em função de seu poder de processamento (IBRAHIM, 2019).

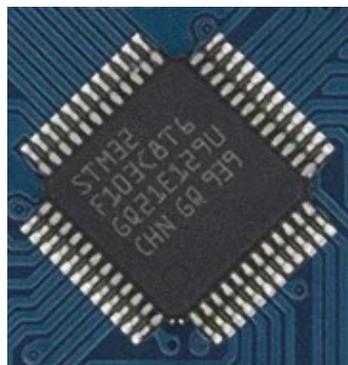


Figura 27 – Um dos modelos de ARM disponíveis. Adaptado de (MARQUES; CARRIJO; JÚNIOR, 2019).

A família de microcontroladores STM32, baseado no processador ARM Cortex-M3, visto na figura 27, conta com uma vasta estrutura para a criação de sistemas embarcados e, assim como o PIC, inclui dezenas de opções de configurações diferentes em termos de performance, tamanho de memória, periféricos e consumo de energia.

Embora o STM32 seja tão poderoso e flexível, o desenvolvimento de *software* em projetos utilizando ARM pode ser um pouco desafiador para um iniciante, por conter uma extensa documentação e bibliotecas como referências principais (BROWN, 2012). Existem, porém, alguns conjuntos de opções de ferramentas de programação que auxiliam o usuário,

além de possibilitar seu uso em projetos mais complexos como a utilização de sistemas operacionais de tempo real (RTOS), que são essenciais na construção de certos tipos de sistemas embarcados.

Como uma forma de popularizar o uso de microcontroladores ARM pela maior parte dos projetistas iniciantes e até mesmo para estudantes, foi criada a plataforma de desenvolvimento STM32 F103C8T6, baseado em Cortex-M3, e mostrada na figura 28. Apesar do seu pequeno tamanho, ela é uma placa muito potente, contando com velocidade de processamento de até 72MHz, 37 pinos de entrada/saída, baixo consumo de energia, interface de comunicações USART, I2C, SPI, CAN, 10 canais ADC com resolução de 12 *bits*, e com um custo baixo.

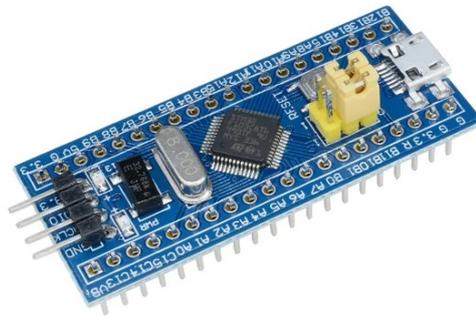


Figura 28 – Placa de desenvolvimento STM32 F103C8T6 (MARQUES; CARRIJO; JÚNIOR, 2019).

3.2.4 Escolha do Microcontrolador

Com as definições de quais seriam as opções para utilização no projeto, foi desenvolvida uma matriz de decisão a fim de selecionar a melhor opção. A tabela 1 mostra os principais parâmetros levados em consideração para a escolha e o peso de cada um destes no valor final do item.

Após a pesquisa em relação aos microcontroladores que poderiam ser utilizados, a escolha mais óbvia provavelmente seria a utilização do modelo ARM, devido a sua gama de especificações e por fornecer maior poder de processamento por um preço reduzido. Porém, devido às circunstâncias do projeto, em que se tem pouco tempo para o projeto e orçamento limitado, optou-se inicialmente por utilizar o ATmega328, processador também usado na plataforma Arduino.

Outros fatores importantes para a escolha do ATmega328 foram pela sua facilidade de implementação e utilização, programação simples e com diversas funções que auxiliam na criação de códigos. Além da opção da realização de testes preliminares utilizando alguma placa comercial, como o Arduino UNO ou Nano, por exemplo, sem a necessidade de fabricação de uma placa de circuito impresso nas fases iniciais de desenvolvimento. É preciso considerar também que durante a evolução deste projeto optou-se pela utilização de

Tabela 1 – Matriz de decisão das opções de microcontroladores

Indicadores	Peso	PIC	Arduino	ARM
Preço	4	1	4	3
Especificações técnicas	3	4	1	5
Periféricos do chip	2	4	2	4
Consumo de energia	2	2	4	3
Programação	5	2	4	1
Custo de projeto (hora-homem)	4	1	3	1
Facilidade de utilização	3	3	5	2
TOTAL		51	78	56

módulos, em que cada módulo é projetado com um microcontrolador e será utilizado para a leitura de apenas um ou dois sensores ao mesmo tempo, podendo estes serem analógicos ou digitais. Dado o número de portas que o processador contém e suas especificações, fica claro que tal escolha pode ser utilizada sem nenhum problema, permitindo até mesmo melhorias nos módulos futuramente.

3.3 Análise de dados

Com o microcontrolador escolhido e após a leitura e processamento dos dados, é necessário o uso de alguma plataforma para apresentar os dados ao piloto. Para este projeto foi escolhido um *display* LCD que servirá como interface entre o sistema de aquisição instalado e o piloto, tendo a função de mostrar algumas informações em tempo real, ao mesmo tempo que todos os parâmetros medidos são gravados em um cartão SD também instalado no sistema dentro do carro.



Figura 29 – *Display* utilizado para apresentar informações ao piloto (DWIN, 2012).

Optou-se pela utilização de um *display* de 3,5 polegadas que tem uma resolução de 320x240 *pixels*, modelo mostrado na figura 29.

As informações mostradas pelo *display* são: velocidade do carro, rotação do motor e temperaturas da caixa de redução e da CVT, além de informar também os sistemas de nível da bateria, nível do combustível e funcionamento da gravação das informações em um cartão de memória. Seu *layout* pode ser visto na figura 30.

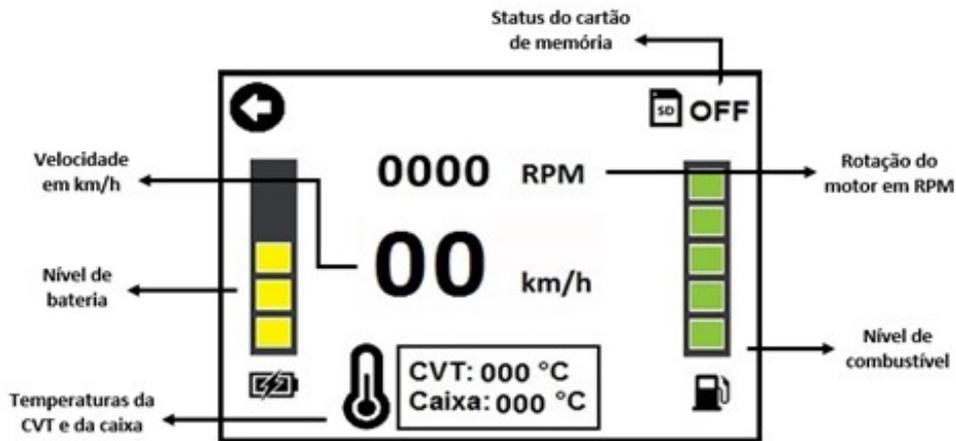


Figura 30 – *Layout* das informações mostradas para o piloto.

Com todas essas informações em tempo real, o piloto tem mais controle sobre a performance geral do carro, podendo tomar melhores decisões durante as provas da competição ou durante testes que estejam sendo realizados.

Além disso, outro módulo tem a função de armazenar todos os dados em um cartão de memória SD. São gerados arquivos contendo os dados que são gravados no formato .CSV, que permite exportar todos eles em formato de uma planilha, facilitando sua visualização, como pode ser visto na figura 31, que demonstra o padrão de gravação do sistema utilizando-se do *software Excel* para a sua posterior leitura.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
106	Genau Baja UFScar																					
107	Temperatura CVT		Temperatura Caixa			RPM Motor			Velocidade			Distancia Percorrida			Combustivel			Bateria				
108	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor
109	0.14	233.82	0.14	228.20	0.50	0.00	0.50	0.00	0.50	0.00	0.50	0.00	0.50	0.00	0.50	100.00	0.50	4.59	0.50	4.59	0.50	4.59
110	0.50	17.64	0.50	17.35	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	100.00	1.00	4.59	1.00	4.59	1.00	4.59
111	1.00	15.40	1.00	14.81	1.50	0.00	1.50	0.00	1.50	0.00	1.50	0.00	1.50	0.00	1.50	100.00	1.50	4.59	1.50	4.59	1.50	4.59
112	1.50	16.47	1.50	15.74	2.00	0.00	2.00	0.00	2.00	0.00	2.00	0.00	2.00	0.00	2.00	100.00	2.00	4.58	2.00	4.58	2.00	4.58
113	2.00	15.30	2.00	14.76	2.50	0.00	2.50	0.00	2.50	0.00	2.50	0.00	2.50	0.00	2.50	100.00	2.50	4.57	2.50	4.57	2.50	4.57
114	2.50	16.81	2.50	16.28	3.00	0.00	3.00	0.00	3.00	0.00	3.00	0.00	3.00	0.00	3.00	100.00	3.00	4.55	3.00	4.55	3.00	4.55
115	3.00	15.64	3.00	15.15	3.50	0.00	3.50	0.00	3.50	0.00	3.50	0.00	3.50	0.00	3.50	100.00	3.50	4.55	3.50	4.55	3.50	4.55
116	3.50	15.20	3.50	14.61	4.00	0.00	4.00	0.00	4.00	0.00	4.00	0.00	4.00	0.00	4.00	100.00	4.00	4.55	4.00	4.55	4.00	4.55
117	4.00	16.03	4.00	15.54	4.50	0.00	4.50	0.00	4.50	0.00	4.50	0.00	4.50	0.00	4.50	100.00	4.50	4.56	4.50	4.56	4.50	4.56
118	4.50	14.86	4.50	14.27	5.00	0.00	5.00	0.00	5.00	0.00	5.00	0.00	5.00	0.00	5.00	100.00	5.00	4.54	5.00	4.54	5.00	4.54
119	5.00	15.00	5.00	14.37	5.50	0.00	5.50	0.00	5.50	0.00	5.50	0.00	5.50	0.00	5.50	100.00	5.50	4.54	5.50	4.54	5.50	4.54
120	5.50	14.76	5.50	14.22	6.00	0.00	6.00	0.00	6.00	0.00	6.00	0.00	6.00	0.00	6.00	100.00	6.00	4.54	6.00	4.54	6.00	4.54
121	6.00	15.25	6.00	14.71	6.50	0.00	6.50	0.00	6.50	0.00	6.50	0.00	6.50	0.00	6.50	100.00	6.50	4.54	6.50	4.54	6.50	4.54
122	6.50	16.37	6.50	15.79	7.00	0.00	7.00	0.00	7.00	0.00	7.00	0.00	7.00	0.00	7.00	100.00	7.00	4.54	7.00	4.54	7.00	4.54
123	7.00	15.98	7.00	15.44	7.50	0.00	7.50	0.00	7.50	0.00	7.50	0.00	7.50	0.00	7.50	100.00	7.50	4.54	7.50	4.54	7.50	4.54
124	7.50	16.67	7.50	16.03	8.00	0.00	8.00	0.00	8.00	0.00	8.00	0.00	8.00	0.00	8.00	100.00	8.00	4.54	8.00	4.54	8.00	4.54
125	8.00	15.40	8.00	14.81	8.50	0.00	8.50	0.00	8.50	0.00	8.50	0.00	8.50	0.00	8.50	100.00	8.50	4.54	8.50	4.54	8.50	4.54
126	8.50	14.86	8.50	14.32	9.00	0.00	9.00	0.00	9.00	0.00	9.00	0.00	9.00	0.00	9.00	100.00	9.00	4.54	9.00	4.54	9.00	4.54
127	9.00	15.30	9.00	14.86	9.50	0.00	9.50	0.00	9.50	0.00	9.50	0.00	9.50	0.00	9.50	100.00	9.50	4.54	9.50	4.54	9.50	4.54
128	9.50	15.05	9.50	14.61	10.00	0.00	10.00	0.00	10.00	0.00	10.00	0.00	10.00	0.00	10.00	100.00	10.00	4.54	10.00	4.54	10.00	4.54

Figura 31 – Padrão dos arquivos com os dados gravados no cartão de memória.

3.4 Protocolo de Comunicação

Com a evolução do projeto e a adição de cada vez mais componentes e sensores, devido a algumas adversidades como o aumento de cabeamento do chicote, maior suscetibilidade a interferências, aumento da quantidade de dados e de funções a se realizar através do microcontrolador, houve a necessidade de torna-lo descentralizado e mais robusto. O próximo passo se deu então pela decisão de dividir todo o projeto em vários módulos espalhados pelo carro e utilizar um protocolo de comunicação para que eles se comunicassem entre si, de forma que fosse capaz de suprir as demandas do sistema e aumentar a sua confiabilidade.

3.4.1 Protocolo CAN

Para estabelecer uma comunicação segura e eficiente entre os módulos que fazem parte do projeto, foi escolhido como protocolo de comunicação o protocolo CAN (*Controller Area Network*), o qual é amplamente utilizado em sistemas automotivos comerciais. A escolha desse protocolo dá-se por uma gama de características importantes encontradas nele, como sua alta velocidade de comunicação, mostrada na figura 32, que pode chegar a 1Mbps a curtas distâncias (cerca de 30 a 50m) ou 20kbps a longas distâncias (cerca de 3km) (NATALE et al., 2012), impede interferências eletromagnéticas, tem mecanismos contra quebra de mensagem e perda de dados, além da possibilidade de se conectar um número grande de nós à rede, permitindo a sua melhoria e/ou expansão futuramente.

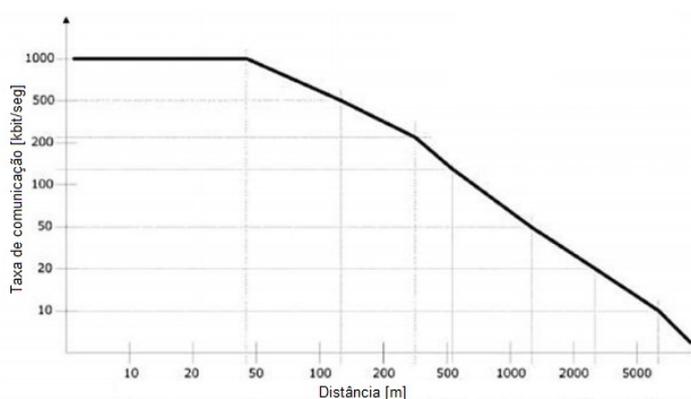


Figura 32 – Taxa de comunicação da rede CAN em relação ao tamanho do cabeamento (SOUZA, 2013).

A rede CAN define três níveis do modelo OSI incorporados em sua tecnologia: a Física (*Physical Layer*), enlace (*Data Link Layer*) e Aplicação (*Application Layer*), como mostrado na figura 33. Como já mencionado na revisão bibliográfica, aqui a camada física trata das características do sinal utilizado para a comunicação, como temporização, nível de tensão e diferenças entre o *bit* dominante do recessivo, enquanto a camada de enlace cria o pacote da mensagem com algumas informações extras que permitem definir a segurança

do protocolo e o tratamento dos dados para a detecção de erros, como quebra ou perda de informação. Por fim, a camada de aplicação tem a função de disponibilizar as informações ao usuário, tornando-os acessíveis para seu uso.

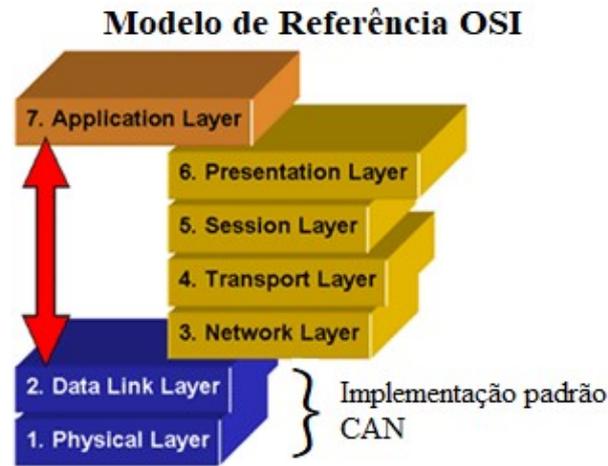


Figura 33 – Definição do protocolo CAN nas camadas do modelo padrão OSI (MARQUES; FRANCO, 2004).

Para a comunicação, o barramento utiliza 4 fios, sendo um deles o VCC, outro o GND e os dois últimos para a transmissão das mensagens, CAN_H (CAN *High*) e CAN_L (CAN *Low*). Os dados enviados pela rede devem ser interpretados pela análise da diferença de potencial entre os fios CAN_H e CAN_L. Para a correta comunicação é utilizado o conceito de dominância de *bit*, o que permite que qualquer módulo aja como mestre e envie uma mensagem e também possibilita que exista prioridade entre as mensagens mais importantes. Com isso, é definido que o *bit* recessivo possui valor lógico alto e o *bit* dominante possui valor lógico baixo, sendo que para gerar um *bit* dominante é forçado o nível elétrico do fio CAN_H para 3,5V e é forçado o fio CAN_L para 1,5V, estabelecendo uma diferença de potencial entre os dois fios de 2V, como é demonstrado na figura 34.

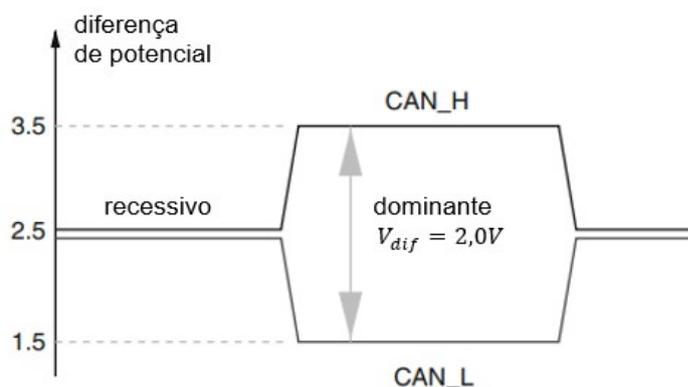


Figura 34 – Representação dos níveis de tensão para a criação de um *bit* dominante (NATALE et al., 2012).

Com relação ao formato das mensagens do protocolo CAN, a transferência delas é manifestada e controlada por quatro tipos de *frames* diferentes: o *frame* de dados que carrega a mensagem do módulo transmissor para receptores, o *frame* remoto que é utilizado para que um módulo solicite ao barramento um *frame* de dados com um identificador específico, o *frame* de erro que pode ser transmitido por qualquer módulo ao identificar um erro no barramento, e o *frame* de sobrecarga que é utilizado para que haja um intervalo extra entre dois *frames* de dados ou remotos (BOSCH, 1991).

Na versão 2.0B do protocolo, tanto o *frame* de dados quanto o remoto podem ser definidos entre dois tipos de formatos diferentes que se diferem no tamanho do campo identificador, o qual pode ser do tipo *Standard*, que contém 11 *bits*, ou do tipo *Extended*, que contém 29 *bits*. É através desse campo identificador que ocorre a prioridade de mensagens. Como pode ser visto na figura 35, que demonstra um exemplo de prioridade de mensagens do protocolo, devido ao monitoramento contínuo de cada módulo de sua própria mensagem, ambos os nós B e C enviam mensagens ao mesmo tempo, sendo que após enviar um *bit*, cada módulo analisa o barramento e verifica se outro módulo da rede o sobrescreveu, pois como visto, um *bit* dominante sempre sobrescreve um *bit* recessivo. Neste caso, o nó B interromperá imediatamente sua transmissão, ao perceber que existe outro módulo enviando uma mensagem com prioridade maior. Este módulo com prioridade maior continuará sua transmissão normalmente. Dessa forma, o protocolo consegue evitar a colisão entre mensagens, utilizando-se de uma arbitragem *bit a bit* não destrutiva (HPL, 2002).

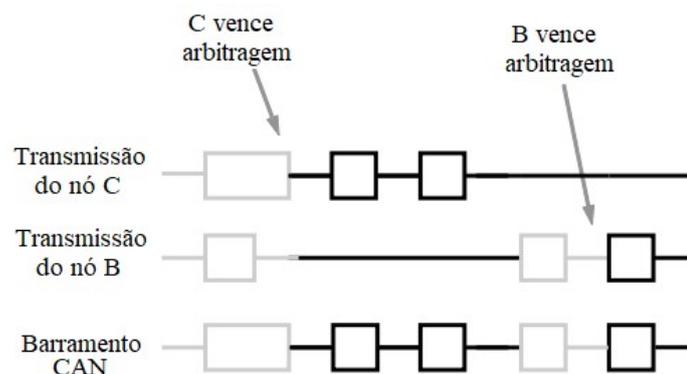


Figura 35 – Representação da arbitragem de mensagens utilizada no protocolo CAN. (HPL, 2002).

O *frame* de dados é o tipo mais importante do protocolo CAN por ele ser responsável por enviar as informações entre os módulos, por isso existe a necessidade de atribuir algumas outras instruções ao *frame* além da informação propriamente dita. Tanto o formato *standard* quanto o formato *extended* é composto por sete campos diferentes, como mostrado na figura 36.

Os campos utilizados nas mensagens são definidos como:

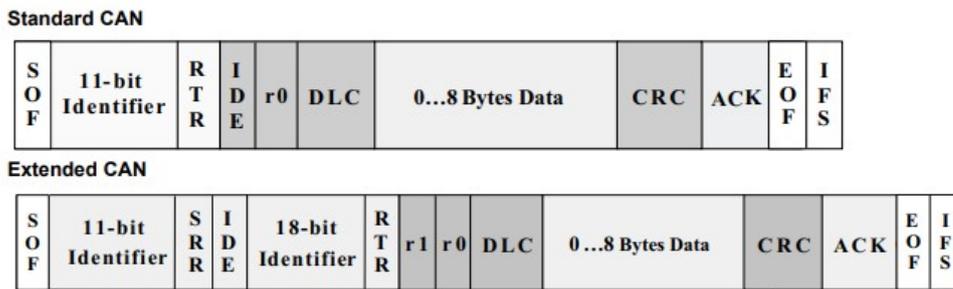


Figura 36 – Os formatos do *frame* de dados (HPL, 2002).

- **SOF - *Start of frame* (1 bit, dominante):** Indica o início do envio de um *frame* por um dos nós. Ele ocorre logo após a verificação de que o barramento está inativo e serve como uma forma de sincronização por todo o barramento e nós que fazem parte do sistema;
- ***Arbitration field* (11 bits standard, 29 bits extended):** Este campo tem formato diferente para cada tipo de *frame* da mensagem, podendo ser de 11 bits ou 29 bits dependendo do tipo escolhido. Sua função é estabelecer a prioridade da mensagem, sendo que quanto menor o valor do identificador, maior sua prioridade. O identificador é seguido pelo *bit* RTR, que define se o *frame* é de dados (*bit* dominante) ou se é um *frame* remoto (*bit* recessivo). Em seguida ocorre o *bit* IDE que define se o *frame* é do tipo padrão ou estendido, dessa forma ele permite prioridade para o formato padrão (em que esse *bit* é dominante) caso haja duas mensagens com mesmo ID. SRR é um *bit* utilizado no *frame* estendido e é sempre recessivo;
- ***Control field* (6 bits):** O campo de controle fornece o tamanho da mensagem enviada. Seus 2 primeiros bits (IDE e r1 para *frame* padrão, r1 e r0 para o estendido) serão sempre dominantes, porém aceitam receber bits recessivos e isso não altera o envio ou recebimento da mensagem. Em seguida, o número de bytes do campo de dados é definido pelo *data length code* que consiste em 4 bits que indicam esse número de forma binária. Os valores 0 a 7 consistem no seu próprio valor em bytes do tamanho da mensagem, qualquer outro valor acima desse indica que a mensagem contém 8 bytes;
- ***Data field* (0 a 64 bits):** A mensagem ou os dados que serão enviados são escritos nesse campo, podendo conter entre 0 e 8 bytes de tamanho;
- ***CRC field* (16 bits):** O CRC (*cyclic redundancy check*) é um campo com 15 bits e mais 1 bit limitador recessivo. É utilizado como *checksum*, o qual tem como definição a criação de um código que permite a verificação da existência de algum erro no recebimento da mensagem. O módulo emissor da mensagem trata a mesma como

um polinômio, realizando uma divisão pelo polinômio gerador:

$$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1 \quad (3.3)$$

O resto da divisão de módulo 2 gera a sequência CRC que é transmitida junto da mensagem, para a posterior verificação pelo módulo receptor que realiza a mesma operação após o recebimento. Se o resultado não for o mesmo, o receptor descarta o *frame* e gera um *Error frame* para solicitar uma nova transmissão;

- **Acknowledge field (2 bits):** O campo ACK é formado por apenas dois *bits*, ACK *slot* e ACK *delimiter*. Durante a transmissão, ambos os *bits* são enviados como recessivos pelo módulo transmissor e, caso o receptor tenha recebido uma mensagem válida, reporta isso para o transmissor enviando um *bit* dominante durante o ACK *slot*. Dessa forma, o transmissor verifica que no barramento existe um *bit* dominante no ACK *slot* e determina que a sua mensagem enviada está correta;
- **EOF – End of frame (7 bits):** Ao final de cada *frame* de mensagem ou remoto é enviado uma sequência de sete *bits* recessivos que informa o fim de uma mensagem e permite que não haja falha no barramento caso exista o envio de um *Error frame* por algum módulo conectado.

À nível de *hardware*, a implementação física da rede CAN necessita de alguns componentes para a criação e envio das mensagens. Além do microcontrolador, que tem a função de ler, processar e enviar as informações recebidas, são utilizados um controlador CAN e um *transceiver*. O circuito eletrônico utilizando estes três componentes foi feito utilizando o *software Eagle* e pode ser visto na figura 37.

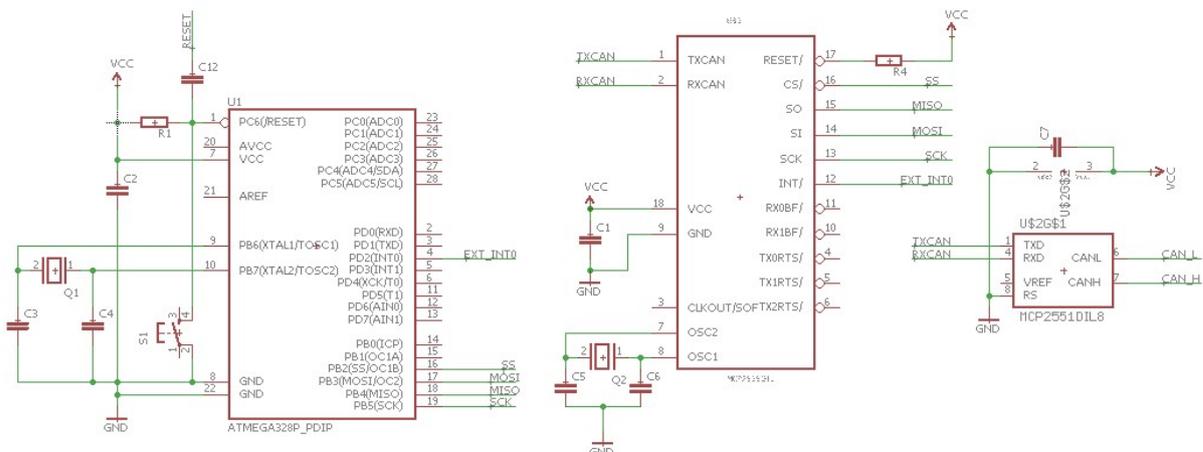


Figura 37 – Circuito utilizado para a implementação da rede CAN em todos os módulos.

O controlador CAN utilizado é o MCP2515, produzido pela Microchip (MICRO-CHIP, 2018). Ele é responsável pela implementação e controle da rede, fazendo a comunicação entre o microcontrolador e o *transceiver*, de forma que ele codifica as informações antes

delas serem transmitidas e também as decodifica ao recebê-las. Também é responsável por fazer a arbitragem da rede e a análise de erros.

O *transceiver* utilizado é o MCP2551, que também é produzido pela Microchip (MICROCHIP, 2006). Este CI é responsável por transformar as mensagens codificadas pelo controlador em pulsos elétricos e enviá-las através do barramento.

Por fim, com a utilização da rede, foi possível definir o uso de quatro tipos diferentes de módulos neste projeto: módulo de leitura de sensores, módulo central para gravação dos dados, módulo *display* e um módulo para envio dos dados por telemetria. Estes módulos foram todos produzidos no decorrer do projeto e utilizam a rede de comunicação interna para os devidos fins do projeto, cada um com sua função específica, podendo enviar ou receber mensagens de acordo com a necessidade.

3.4.2 Criação das mensagens nos módulos

Como o protocolo CAN define mensagens com 8 *bytes* de tamanho, foram definidos alguns padrões para as mensagens de cada módulo, para garantir que as mensagens enviadas e recebidas sejam tratadas de acordo com a informação que elas estão carregando.

Sendo assim, cada mensagem transmitida carrega uma ID que define a informação que ela contém, para que todos os módulos configurados para o recebimento consigam definir se aquela é uma mensagem destinada para si. A tabela 2 mostra essa divisão e identificação de cada mensagem de acordo com o módulo que a envia.

Tabela 2 – Padrão definido para as mensagens enviadas pela rede.

REDE CAN											
Modulo	Sensor	ID	Transmissão/ Recebimento	Mensagem							
				1 byte	2 byte	3 byte	4 byte	5 byte	6 byte	7 byte	8 byte
1	LM35 - Proteção da CVT	2	Transmissão	Tempo				Temp. CVT			
1	LM35 - Caixa de Redução	3	Transmissão	Tempo				Temp. Caixa			
2	Hall - Rotação do Motor	4	Transmissão	Tempo				Rotação			
3	Indutivos - Velocidade	5	Transmissão	Tempo				Velocidade			
4	Célula de Carga - Combustível	6	Transmissão	Tempo				Nivel Comb.			
5	Display		Recebimento	Velocidade	Rotação	Temp. Caixa	Temp. CVT	Nivel Comb.	Nivel Bateria	Distancia percorrida	Status Gravação
		7	Transmissão 1	Longitude				Latitude			
6	GPS / Telemetria		Recebimento	Velocidade	Rotação	Temp. Caixa	Temp. CVT	Nivel Comb.	Nivel Bateria	Distancia percorrida	Status Gravação
		8	Transmissão 2	Altitude				Velocidade	Satelites	-	Indicação de leitura
7	Central / Gravação SD	1	Transmissão	Velocidade	Rotação	Temp. Caixa	Temp. CVT	Nivel Comb.	Nivel Bateria	Distancia percorrida	Status Gravação
			Recebimento 1	Tempo				Temp. CVT			
			Recebimento 2	Tempo				Temp. Caixa			
			Recebimento 3	Tempo				Rotação			
			Recebimento 4	Tempo				Velocidade			
			Recebimento 5	Tempo				Nivel Comb.			
			Recebimento 6	Longitude				Latitude			
			Recebimento 7	Altitude				Velocidade	Satelites	-	Indicação de leitura

Como visto na tabela, os quatro módulos de sensores e mais o módulo GPS enviam as informações coletadas por eles ao módulo de cartão SD que nesse caso é tratado como uma central, tendo como função agrupar todos os dados em mensagens menores e reenviar para os módulos do *display* e para telemetria. Esse processo ocorre de forma que os módulos de sensores enviam a medição juntamente do tempo em que ela ocorreu, o módulo GPS envia em duas mensagens diferentes todas as suas informações, para que tudo seja gravado

no cartão SD. Em seguida, as informações disponibilizadas ao *display* e telemetria são reunidas em uma única mensagem onde cada *byte* é relacionado a uma medição.

A mensagem enviada pelo módulo central é a de maior prioridade, possuindo assim o ID de menor valor, e ao mesmo tempo sendo tratado com maior prioridade pelo protocolo CAN. Como 1 *byte* pode assumir qualquer valor entre 0 e 255, quase todas as informações são possíveis de serem enviadas sem nenhuma alteração, a única exceção se da pela rotação do motor que pode assumir valores de até próximo de 4000 rotações por minuto, por isso seu valor é dividido por 100, para que ao chegar no destino seja novamente multiplicado pelo mesmo fator, assumindo seu valor original.

3.5 Resultados Preliminares

Todo o projeto abordado neste capítulo, desde os sistemas embarcados e a escolha de cada componente para a coleta de dados, bem como o protocolo de comunicação CAN são discutidos aqui com um embasamento teórico inicial, porém com alguns pontos de seu desenvolvimento de forma mais resumida, focando apenas nos resultados e aspectos mais importantes para o entendimento do projeto.

Sendo assim, a parte final a ser discutida nesta seção é o projeto dos circuitos eletrônicos e as placas de circuito impresso (PCIs) que foram desenvolvidos após todos os estudos e testes apresentados. Todos os esquemáticos das placas, desenvolvidos no *software Eagle*, podem ser vistos no anexo A.

Com o uso do protocolo CAN, foi possível optar por se utilizar de um processamento descentralizado, dividindo o sistema em algumas placas de circuito que tenham seu papel bem definido, mostrado na tabela 2. Segue então que se definiu sete placas diferentes para o uso, sendo três para leitura dos sensores, uma para o nível de combustível, uma para o *display*, uma para a gravação do cartão SD e uma para leitura do módulo GPS e uso de telemetria.

O projeto do *hardware* se iniciou com as placas de sensores, nelas buscou-se utilizar um padrão para que fossem utilizados *jumpers* de seleção, permitindo selecionar se o sensor utilizado será conectado a uma porta de leitura digital ou analógica do ATmega328, além da seleção de sua alimentação, que pode ser em 5V ou 12V. Esse conceito foi utilizado buscando uma maior facilidade na manutenção do sistema, de forma que como os sensores são os componentes mais expostos às intempéries da competição, estas placas são as mais suscetíveis a danificações ou queima, necessitando de unidades reservas para possíveis emergências, como essas citadas, durante alguma prova. Na figura 38 são mostradas as placas após a manufatura e solda dos componentes.

Após a finalização dos módulos para as leituras dos sensores foram feitos os dois

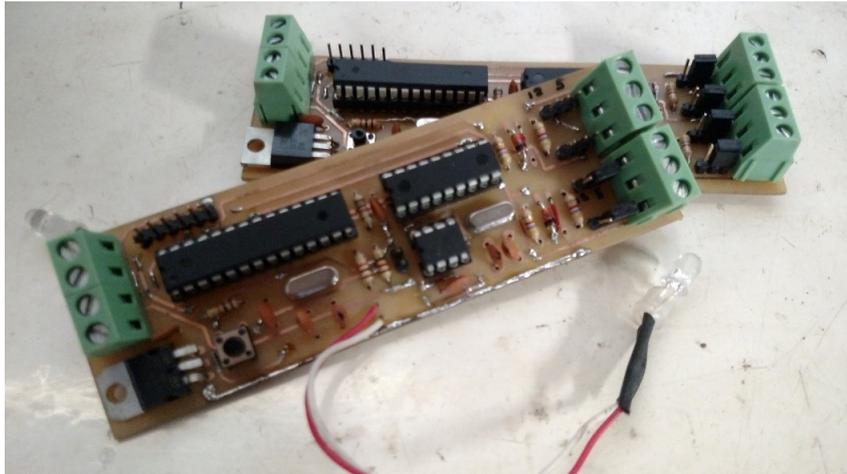


Figura 38 – Placas produzidas para a leitura de sensores.

módulos responsáveis por guardar e apresentar os dados coletados. O primeiro trata-se de uma placa usada para a gravação de todos os dados em um cartão de memória, a placa de circuito impresso é mostrada na figura 39. O módulo de cartão SD foi projetado, como pode ser visto no seu esquemático da figura 64 (Anexo A), com dois microcontroladores ATmega 328. Essa escolha foi feita devido ao fato do módulo do cartão de memória utilizar a mesma comunicação que o *chip* de interface da rede CAN, que é a SPI, e pelo ATmega só possuir uma saída para esse tipo de comunicação. Porém, a SPI só pode ser utilizada para mensagens entre o ATmega e um dispositivo por vez, impossibilitando que as mensagens que chegam pela rede CAN sejam recebidas ao mesmo tempo em que os dados são gravados no cartão de memória, o que faz com que possam ocorrer perdas de mensagens no momento em que estivessem sendo gravados os dados no cartão de memória.

Assim, cada ATmega fica responsável por uma tarefa: um recebe as mensagens pela rede CAN, e o outro grava todos os dados recebidos no cartão de memória. Com isso, após serem recebidas as mensagens de todos os módulos de leitura dos sensores, os dados são transmitidos do primeiro microcontrolador pela sua porta Serial para o seguinte para serem gravados no cartão de memória, garantido que todas as mensagens são recebidas e gravadas sem nenhum erro ou perda.

Há também algumas outras características importantes desta placa, como por exemplo que ela é responsável pela leitura da tensão da bateria para estimar sua carga e alertar caso esta fique baixa, de forma que isso é feito utilizando o circuito explicado na seção que aborda o circuito medidor da carga da bateria. Além disso, como os microcontroladores se comunicam pelas portas Serial dos ATmega, é necessário também uma forma de retirar essa ligação sempre que é necessária a gravação de um novo programa em cada um deles e, por isso, foi adicionado um sistema de *jumpers* que permite alterar o circuito sempre que é necessário fazer essa operação. Esse sistema de *jumpers* é evidenciado na figura 39, com a imagem da placa.

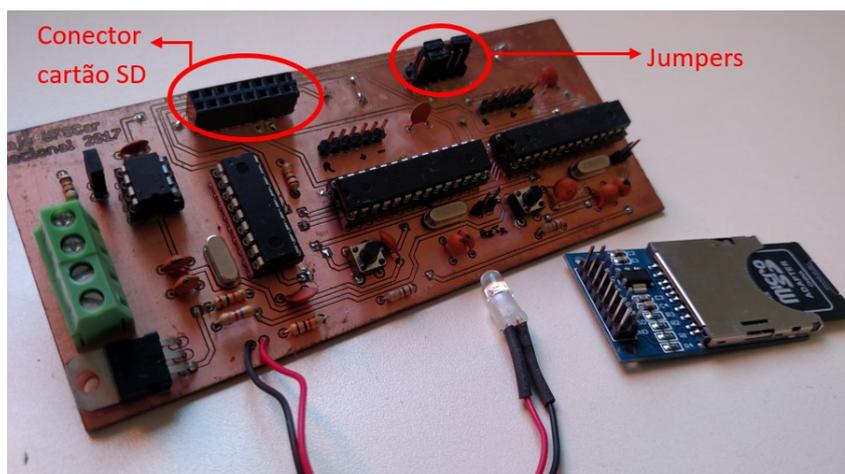


Figura 39 – Placa para a gravação dos dados no cartão SD.

A outra placa produzida, foi o módulo para apresentar parte das informações em um *display* LCD. Tal *display* que foi utilizado neste projeto, utiliza uma comunicação RS-232 e, por isso, esse módulo tem em seu circuito, além dos circuitos para a implementação da rede CAN, o circuito integrado MAX232, que é um conversor de nível e tem como função converter as mensagens enviadas pela porta Serial do ATmega de nível TTL em mensagens no nível do tipo de comunicação RS-232. Na figura 40 é mostrada a placa.

Além da comunicação com o *display*, foram adicionados também alguns LEDs para alertar ao piloto algumas situações críticas no funcionamento do veículo, como temperaturas elevadas, níveis críticos da bateria ou do tanque de combustível, e também se há a gravação correta do cartão de memória e o envio de todas as informações através de telemetria. Esses LEDs estão ligados a algumas portas digitais do ATmega desse módulo e são acionados conforme esses eventos citados ocorrem.

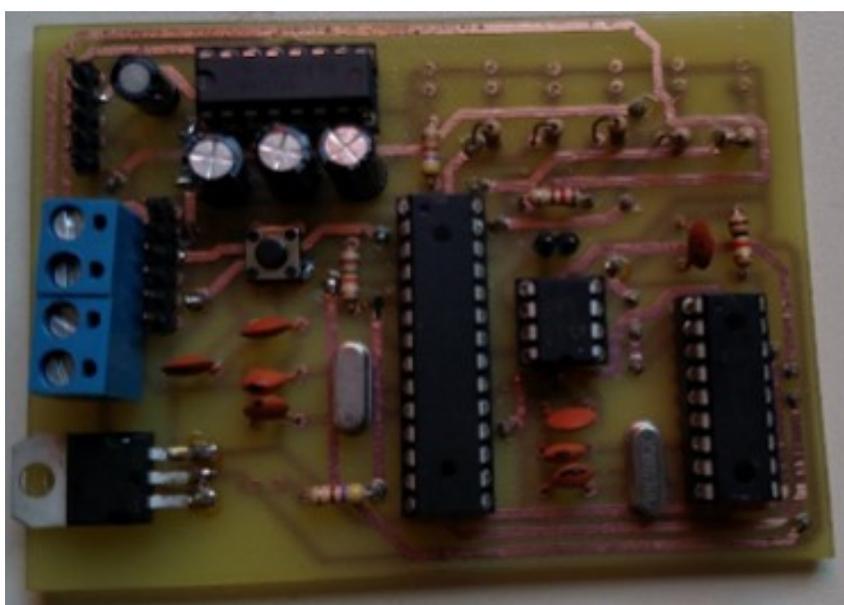


Figura 40 – Placa para a utilização do *display* LCD.

O último componente de *hardware* necessário para o funcionamento do sistema é a placa destinada a implementar a telemetria ao sistema. Tal módulo utiliza um módulo LoRa para enviar mensagens com os dados obtidos do carro via rádio para um outro que estivesse à distância (onde poderia ser o *box* da equipe durante as competições), e além disso, devido a sua posição de instalação no carro em um dos tubos que compõem a parte superior da gaiola, recebe também os dados do módulo GPS.

Porém, devido aos problemas enfrentados com a pandemia, dificultando sua manufatura a tempo para sua apresentação neste documento, apenas foi feito o esquemático deste módulo, o qual pode ser visto na figura 66 (Anexo A) junto com a placa observada no *software*, na figura 41.

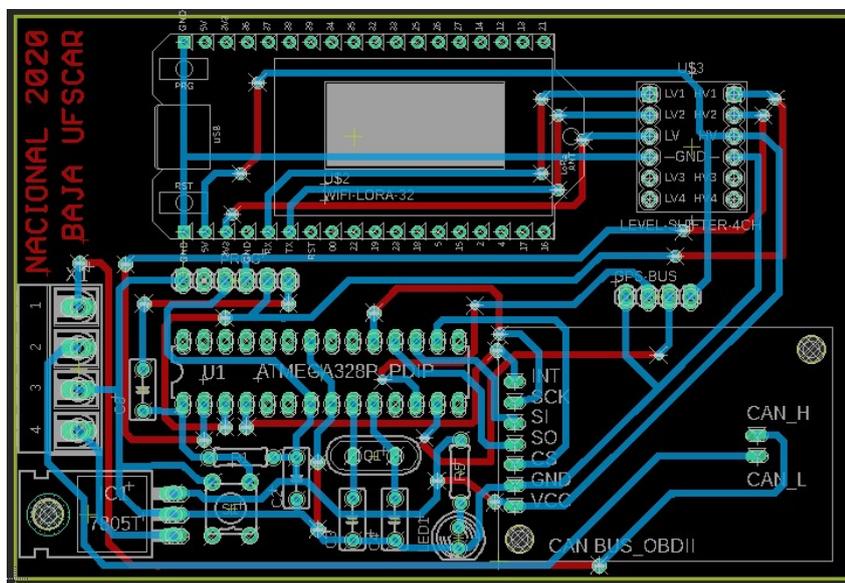


Figura 41 – Placa para a utilização de GPS e telemetria.

Com o uso do módulo de telemetria LoRa da Heltec, o qual será apresentado na seção seguinte, e para evitar que a placa tivesse dimensões muito maiores que as outras placas utilizadas, optou-se pela substituição dos componentes circuitos integrados MCP2151 e MCP2115 para uso da rede CAN, pelo módulo CAN BUS OBDII que realiza a interface do protocolo com o microcontrolador, porém permitindo o uso de uma área menor para a acomodação dos componentes e trilhas de suas ligações. Além disso, como a plataforma ESP32 trabalha com nível lógico de 3,3V, foi necessário o uso do módulo conversor de nível lógico para todos os sinais conectados entre ATmega e módulo LoRa, que é baseado nessa plataforma.

4 Sistema de Telemetria

Para a realização de um dos principais objetivos, que é o de acompanhamento em tempo real da performance do carro, é necessária a inclusão de um sistema de telemetria ao projeto que seja capaz de enviar todos os dados coletados no carro para o *box* durante as competições ou até mesmo em testes da equipe.

Esse tipo de sistema é composto basicamente por um módulo de radiofrequência que seja capaz de transmitir os dados à uma certa distância mesmo estando em movimento junto com o carro. Existem, atualmente diversos módulos de radiofrequência e em diferentes padrões que podem ser utilizados no projeto, variando em suas especificações técnicas, como taxa de transmissão, frequência de operação, área de alcance, além de diferenças em custo e operação. Neste contexto, o módulo XBee Pro, que é utilizado por diversas equipes de Baja para realizar a telemetria e muito difundido em diversos tipos de projetos não vem apresentando resultados satisfatórios situações de uso nas competições e testes.

Por esse motivo, foram iniciadas pesquisas para a utilização de um novo módulo que possa ter características que atendam melhor ao projeto e que possa ser utilizado mesmo na aplicação específica mobilidade que é o Baja. O módulo LoRa, que é um módulo que utiliza o protocolo LoRaWAN e seu *hardware* é desenvolvido na plataforma ESP32, se trata de uma tecnologia mais recente e ainda pouco utilizado em projetos de mobilidade, porém bastante promissora por alcançar grandes distâncias com um baixo consumo de energia.

Neste capítulo, são realizados alguns testes buscando a comparação entre os módulos para a aferição dos principais parâmetros de ambos os módulos na prática e definir qual deles se sai melhor na aplicação desejada. Como o XBee já foi testado pela equipe em outras ocasiões, será dado um foco maior nos estudos com o LoRa, buscando definir suas melhores configurações para o uso e o real teste de sua capacidade como peça principal de um sistema de telemetria automotivo.

4.1 XBee PRO S1

O módulo XBee, desenvolvido pelo fabricante Digi, mostrado na figura 42, é uma plataforma *hardware* de radiofrequência que utiliza comunicação no padrão ZigBee IEEE 802.15.4, que foi desenvolvido como um protocolo aberto de comunicação sem fio, baixo consumo de energia e alta segurança e que pode operar em diferentes aplicações, porém com um custo um pouco elevado. Ele conta com taxas de transferências de dados que variam de 20 a 250 kbps, consumo de até 63mW e um alcance de 90m em ambientes

fechados e até 750m em ambientes abertos com linha de visada (DIGI, 2018).



Figura 42 – Módulo de radiofrequência Digi XBee (DIGI, 2020).

Esse sistema é robusto e pode ser utilizado em uma rede com múltiplos módulos, podendo até mesmo ser expandida com a inclusão de mais módulos à rede, além de permitir seu uso em diferentes topologias de rede, onde basta configurar cada dispositivo para um dos três tipos disponíveis (coordenador, roteador ou dispositivo final) para que a informação chegue ao endereço final. Assim, cada dispositivo pode ser configurado como:

- Coordenador: se torna responsável pela distribuição de endereços, inicialização e manutenção da rede, reconhecimento de todos os nós, podendo ser uma ligação entre outras redes ZigBee;
- Roteador: é um nó comum da rede, mas que tem a possibilidade de exercer o papel de um roteador intermediário entre os nós, sem precisar de coordenador para isso, assim expandindo a rede e aumentando seu alcance;
- *End device* (dispositivo final): é o dispositivo que fica no final da rede, geralmente hospedando o componente que gera os dados e envia, ou que recebe os dados para que haja a sua atuação (sensores ou atuadores), e por isso permanece a maior parte do tempo em espera, consumindo pouca energia durante seu funcionamento.

A organização de uma rede, seu *layout* físico e lógico, definem qual a topologia usada, e essa diferença se dá para diferentes aplicações e número de dispositivos utilizados ao mesmo tempo. Em uma rede, cada dispositivo é chamado de nó e sua função depende da topologia e local em que está alocado, podendo estar se comunicando com um ou mais nós ao mesmo tempo. As principais topologias utilizadas são: ponto a ponto, barramento, árvore, estrela e malha (SÍRIO, 2018).

Devido à natureza deste projeto, onde há a comunicação entre somente dois nós, a topologia utilizada é a de ponto a ponto, conectando diretamente os dispositivos sem nenhum intermediário, como mostrado na figura 43.

Para a utilização dos módulos, é necessária a realização de uma configuração prévia em cada um deles, para que se comuniquem de forma segura e sem interferências de outras

redes de rádio que possam existir nas proximidades. A execução de todas as configurações dos módulos XBee é realizada através do software X-CTU, desenvolvido pela própria Digi e que possibilita realizar essa tarefa de maneira simples e intuitiva, além da atualização de *firmwares* ou outras funções.



Figura 43 – Topologia do projeto, com dois módulos XBee em comunicação ponto-a-ponto.

Essa ferramenta é muito importante e auxilia não só a configuração dos módulos, mas permitindo também a realização de testes para monitorar e mensurar diversos parâmetros da rede. Ele controla outros diferentes fatores também, como a potência de sinal, tempo de resposta, permite a visualização através de um monitor serial, entre outras informações.

Antes de iniciar o uso dos módulos XBee é necessário realizar a configuração dos módulos que irão se comunicar através do próprio X-CTU, devendo também definir tipo de rede e modo de operação. Há a possibilidade de uso em dois modos distintos de operação. O primeiro modo faz uso de comandos AT, é mais simples e necessita de uma configuração anterior ao envio de dados para que seja estabelecida a conexão, o que em redes com pouco volume de dados não exige muito custo operacional, porém esse modo não possui verificação de integridade e nem informa o endereço do módulo que está enviando os dados. O segundo modo é o API (*Application Programming Interface*), que define uma comunicação muito mais robusta e segura, com uma estrutura mais complexa, onde são formados pacotes com os dados que se queira enviar, juntamente com as informações de origem e destino, seu tamanho e o caractere *checksum*, usado para evitar erros ou perdas, verificando a integridade dos dados (JUNIOR et al., 2013).

A configuração dos módulos é mostrada na figura 44, sendo que para a realização dos testes propostos foram alteradas as informações de endereço de cada módulo para que haja uma paridade entre eles na rede, foram colocados em modo API, o módulo de envio configurado como Coordenador e o de recebimento como Dispositivo Final.

O alcance do módulo, é bastante influenciado pela antena utilizada, por isso foi escolhida uma antena omnidirecional e com amplificação de sinal. O modelo escolhido conta com frequência de transmissão de 2,4GHz e ganho de 9dBi, podendo operar em módulos de até 1W de potência de transmissão.

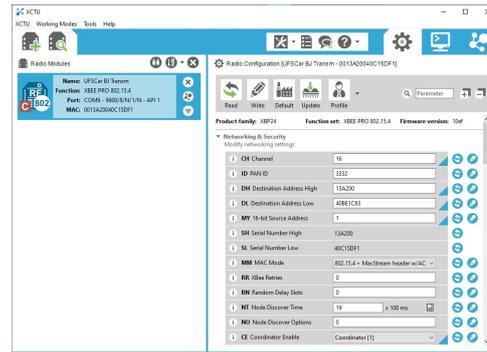


Figura 44 – Interface do *software* X-CTU para configuração dos módulos XBee.

4.1.1 Teste de envio e taxa de dados

A interface auxiliar X-CTU conta com algumas ferramentas que permitem a verificação e a medição de parâmetros da rede em uma comunicação ponto-a-ponto entre dois dispositivos. O *Range Test* é uma dessas aplicações, e ele tem como função o teste do sinal de radiofrequência e a análise da qualidade da comunicação entre dois módulos na mesma rede. Por isso, ele pode ser utilizado como um teste de distância máxima de comunicação e também da efetividade da taxa de envio e recebimento de pacotes entre os módulos. Além disso, ele realiza a apresentação dos resultados do teste graficamente e com a indicação do número de pacotes recebidos em função da quantidade enviada.

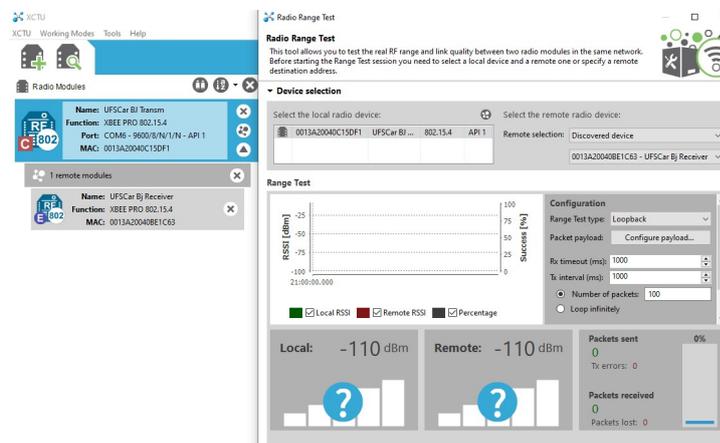


Figura 45 – Configuração dos parâmetros do *Range Test* para o teste de distância com XBee.

O teste pode ter seus parâmetros configurados para o tempo máximo de espera para a resposta do dispositivo remoto, em milissegundos (*RX Timeout*), o tempo entre as mensagens enviadas, também em milissegundos (*TX Interval*), o número de pacotes enviados durante o teste, e a janela de tempo para a visualização dos dados de RSSI (*Receiver Signal Strength Indicator*) que mede a força do sinal recebido, medidos pelo teste. A interface do teste, que mostra essas configurações e os dados medidos durante a sua realização, pode ser vista na figura 45.



Figura 46 – Local e posição das distâncias para os testes utilizando módulo XBee.

Após a configuração dos módulos, foi escolhido o local de testes, mostrado na figura 46, com diferentes distâncias de medição entre 50 e 500 metros, com linha de visada direta, sem nenhum tipo de barreira no caminho entre emissor e receptor, mas por ser um local aberto e público, contando com a passagem de alguns carros na rua.

Logo no início do teste, nas distâncias de 50 e 100 metros, foi observado uma comunicação contínua, com recebimento de 99% dos pacotes e sem nenhuma perda de sinal, como pode ser visto na figura 47 que mostra o resultado após o envio de todos os pacotes configurados no *Range Test*. O RSSI se manteve em todo o momento por volta de -80dBm, que é o valor médio devido à antena utilizada.

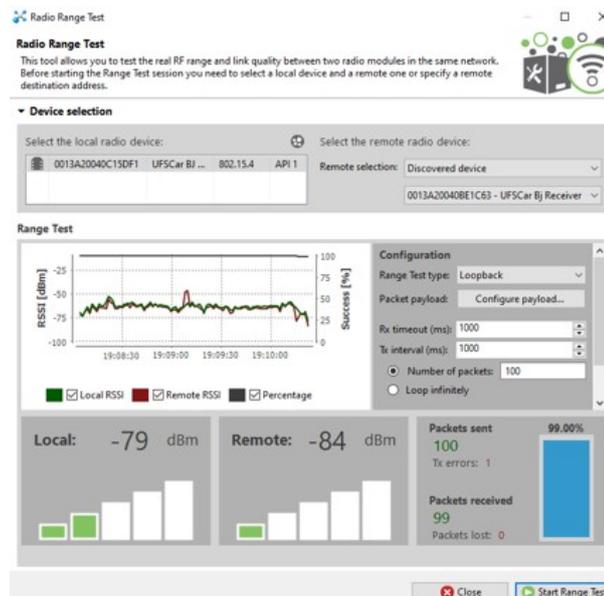


Figura 47 – Resultado observado no teste a uma distância de 50m do ponto de envio.

Porém ao se distanciar e chegar no ponto de 200 metros, pôde ser notada uma perda nos pacotes recebidos, contando com 42% de sucesso no recebimento pelo módulo, como mostrado na figura 48. Ao ir mais distante no próximo ponto em 300 metros, foram

recebidos apenas 21% dos pacotes, com uma constância no recebimento em apenas uma parte do teste.

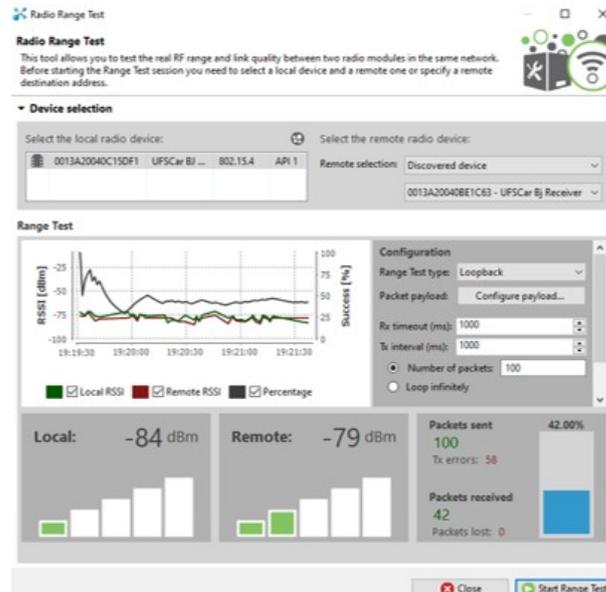


Figura 48 – Resultado observado no teste a uma distância de 200m do ponto de envio.

Por fim, nos dois pontos finais com maior distância, em 400 e em 500 metros, houve perda de conexão em quase todo o teste, contando respectivamente com apenas 12% e 1% de pacotes recebidos.

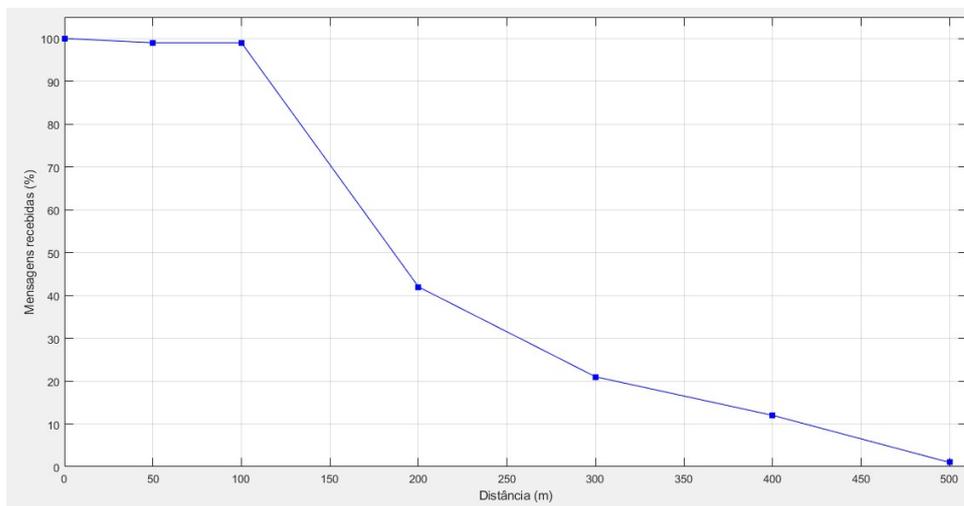


Figura 49 – Desempenho do módulo XBee no *Range Test* em função da distância.

Apesar da especificação do fabricante indicar a distância máxima de envio deste módulo como sendo de aproximadamente 750m, os testes práticos mostraram que há muita perda de pacotes em distâncias muito menores, onde próximo de 300m as perdas de sinal são substanciais e em 500m praticamente nulas, como mostrado no gráfico da figura 49 que demonstra os resultados obtidos nesse teste. Esse tipo de resultado foi também observado em outros trabalhos, como (CAMA-PINTO et al., 2017) e (RYCERSKI et al., 2017), que

demonstram também uma grande perda de sinal e em decorrência disso, perda também dos pacotes enviados, por volta de 300m de distância, mesmo realizando seus testes em diferentes condições e ambientes.

4.1.2 Teste de latência

Outro fator importante a ser testado, é a latência das mensagens. Para isso, é medida a diferença de tempo entre o envio da mensagem e o tempo do seu recebimento no dispositivo final. Nessa medição é importante que haja sincronia entre os módulos, por isso ambos são reiniciados ao mesmo tempo através da conexão entre as suas portas de reset, garantindo a precisão do teste para a aferição do tempo das mensagens no ar.

Foi utilizado o mesmo local do teste anterior, conforme figura 46, tendo diferentes pontos de envio para verificar se há alteração no tempo com o aumento da distância entre os módulos. Para a implementação de um programa que realiza as funções desejadas, foram utilizadas placas Arduino UNO, para que o *software* criado envie 30 pacotes de 12 *bytes* contendo o seu número de identificação e o tempo de envio registrado pelo módulo emissor, ao chegar no seu destino, o módulo de recebimento das mensagens também registra o tempo de chegada. Por fim, é calculada a diferença entre os tempos para que seja definida a latência da comunicação.

O gráfico da figura 50 mostra os resultados obtidos, com todos os valores medidos iniciando em 55ms com os módulos próximos durante a sincronização, até o valor de 63ms na última distância em 400m, que é o ponto onde ainda há um pouco de sinal, possibilitando essa medição.

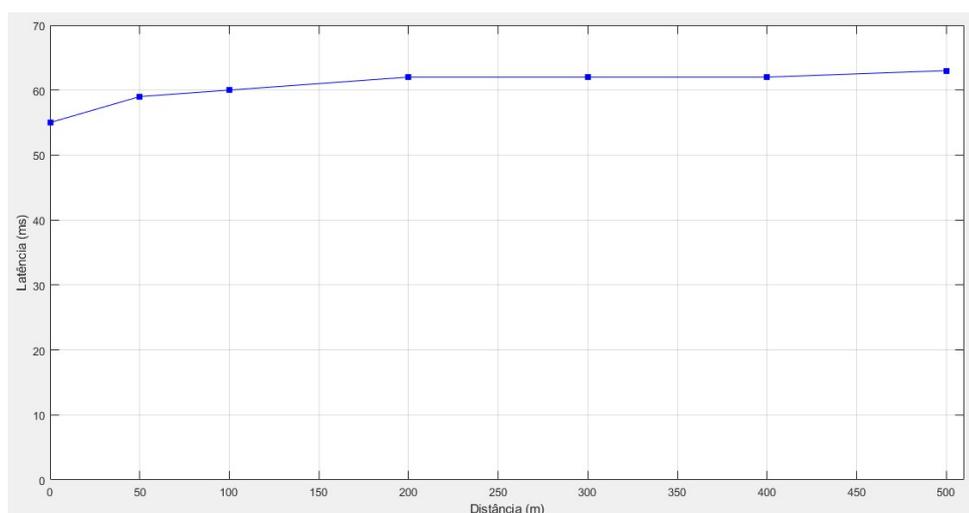


Figura 50 – Desempenho do módulo XBee no teste de latência em função da distância.

Os valores obtidos foram um pouco acima do esperado, pois o fabricante define um padrão de pouco mais de 40ms para mensagens com poucos *bytes* (DIGI, 2017). Por

isso, a latência dos módulos pode ser menor em condições mais favoráveis de teste e com configurações adicionais aos pacotes enviados.

4.1.3 Consumo de energia

O último fator a ser medido para a verificação das características da tecnologia é o consumo de energia. Essa medição deve ser feita em dois modos diferentes para o módulo transmissor, em estado ocioso e durante a transmissão. No estado ocioso o consumo é menor e define o consumo médio do módulo, pois ele permanece nessa condição a maior parte do tempo dado que o envio de mensagens é realizado em um curto período de tempo, de aproximadamente 1ms (MIRANDA; CARDIERI, 2017), gerando apenas um pulso de potência mais elevada para a realização dessa tarefa. Essa condição dificulta a medição do consumo durante a transmissão, por isso, somente para a realização da segunda parte deste teste, o módulo foi configurado para enviar mensagens continuamente, de forma que a corrente consumida durante o teste não oscile, sendo a mesma necessária para a transmissão de mensagens.

Assim, no modo ocioso foi observado o consumo médio do módulo sendo de 45mA, enquanto que durante a transmissão de mensagens o consumo medido foi em torno de 80mA.

Estes resultados estão dentro do esperado, sendo os valores do consumo médio como intermediários ao se comparar com as medições realizadas em outros trabalhos, como (HORVAT; ŠOŠTARIĆ; ŽAGAR, 2012) que considera diferentes módulos nos seus testes e em diferentes situações. Por outro lado, a corrente de transmissão medida é um pouco abaixo da especificada pelo fabricante (DIGI, 2018) que determina o valor de 140mA na versão internacional do XBee Pro S1. Essa variação também pode ser explicada por diferença nas configurações dos módulos e nas condições dos testes.

4.2 LoRa

O módulo LoRa (sigla para *Long Range*), foi desenvolvido como uma tecnologia de baixa potência e longo alcance, para possibilitar o uso de Internet das Coisas (*Internet of Things* - IoT) nas cidades, enviando seus dados por longas distâncias. O protocolo de comunicação da rede é definido pelo LoRaWAN (*Long Range Wide Area Network*) que tem por objetivo a criação de uma rede massiva de grandes áreas urbanas, sendo resistente a ruídos e a sinais com frequências próximas, para auxiliar a rede de telefonia móvel e as tecnologias sem fio de curto alcance (ORTIZ et al., 2019). Ele usa espectro radioelétrico sem licença, nas bandas Industrial, Científica e Médica (ISM) realizando essa comunicação por uma grande área e utilizando uma baixa potência entre dispositivos remotos e *gateways* conectados à rede. Esta abordagem baseada em padrões para a construção de um LPWAN

(*Low Power Wide Area Network*) permite o uso de redes IoT públicas ou privadas em qualquer lugar com todo o sistema sendo usado de forma segura, interoperável e móvel. Esse esquema é mostrado na figura 51.

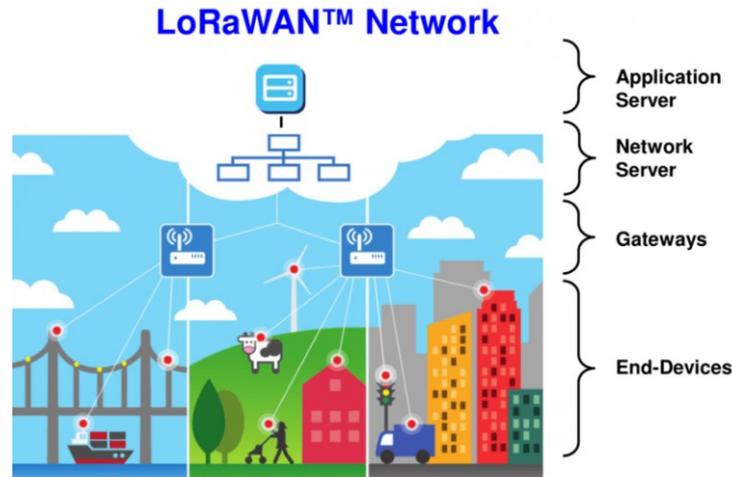


Figura 51 – Topologia de comunicação da rede LoRaWAN em ambientes urbanos (JÚNIOR, 2016).

Da forma em que o protocolo foi desenvolvido, ele utiliza sua camada física para modular o sinal em sub-bandas dentro das faixas especificadas pelo padrão da região em que se encontra. Além disso, o protocolo LoRaWAN utiliza duas camadas do modelo OSI de comunicação, a de enlace, onde é feita a detecção e correção de erros, e a camada de rede, que é criada para garantir a autenticidade de cada nó na rede, controlando o fluxo de dados para evitar congestionamentos da rede (TEIXEIRA; ALMEIDA, 2017).

A faixa de frequências regulamentada para esse protocolo no Brasil está entre 915 e 928 MHz, ou em 433MHz para redes com poucos dispositivos, assim a camada física do LoRa modula os sinais em sub-bandas de rádio nessas faixas de frequências e usa uma técnica de modulação de espalhamento espectral baseado na técnica por *chirp* (CSS), para codificar a informação.

O espalhamento espectral é uma técnica de modulação de sinal em que a largura de banda usada para a transmissão é muito maior que a banda mínima necessária para transmitir a informação. É usado um código pseudo-aleatório que define o espalhamento do sinal e que é utilizado pelo receptor para recuperar a informação, de forma que haja uma melhora na rejeição de sinais interferentes e que esse sinal não seja captado por receptores indesejados (NETTO et al., 2009). Nessa técnica, são utilizados símbolos, denominados *chirps*, onde a frequência varia linearmente com o tempo, aumentando (*up-chirp*) e diminuindo (*down-chirp*) para codificar as informações.

Para definir a capacidade máxima de transmissão de dados do canal, devido ao espalhamento espectral é possível utilizar o Teorema de Shannon-Hartley. Nesse teorema é estabelecida a capacidade de canal para uma rede de comunicação definindo a quantidade

máxima de informação livre de erros em um determinado tempo, com uma largura de banda com ruído de interferência e dada a potência do sinal limitada. Essa capacidade de transmissão é definida pela equação 4.1.

$$C = B \times \log_2 \left(1 + \frac{S}{N} \right) \quad (4.1)$$

Na equação, C é a capacidade do canal (*bits/s*), B representa a largura de banda (Hz), S é a potência média do sinal recebido (W) e N a potência média do ruído de interferência no canal (W). Assim, a taxa de sinal/ruído SNR, expressa pela razão entre S e N , e definida como uma taxa de potência média. Como na rede LoRa, para aplicação de espalhamento a taxa SNR é consideravelmente baixa, dado que o sinal de potência está frequentemente abaixo do esperado, pode-se assumir que SNR é muito menor que 1. A equação pode ser manipulada para logaritmo de base natural, conforme mostrado em 4.2 e 4.3.

$$\frac{C}{B} = 1,44 \times \left(\frac{S}{N} \right) \quad (4.2)$$

$$\frac{C}{B} \approx \frac{S}{N} \quad (4.3)$$

A partir da equação 4.3, pode se observar que para uma transmissão de uma informação livre de erros em um canal onde a relação sinal/ruído é fixa, apenas a largura de banda do sinal transmitido precisa ser alterada, compensando então a degradação desse sinal devido a taxa SNR (SEMTECH, 2015).

Com o tipo de modulação de sinal empregado, um parâmetro muito importante que pode ser alterado para mudar a característica do sinal é o espalhamento espectral, SF. O SF define a razão entre a taxa de *bits* e a taxa de *chirps*. É possível a configuração do SF em seis níveis diferentes, variando entre os valores 7 a 12, isso permite a formação de canais ortogonais, o que significa que enlaces com fatores de espalhamento diferentes criam sinais de transmissão que não interferem entre si e não colidem. Um valor mais alto desse fator aumenta a sensibilidade do limiar de recepção em relação a potência, mas faz com que o tempo de propagação no ar também aumente, diminuindo a taxa de transmissão do canal.

A partir de todos esses recursos que podem ser alterados, a aplicabilidade de uma rede que utilize o protocolo LoRaWAN possui uma capacidade bem definida e que pode ser moldada para diversas aplicações, pois com a escolha dos valores de SF, largura de banda e potência de sinal é possível, por exemplo, aumentar a taxa de dados e diminuir o consumo do sistema, ou então aumentando a distância máxima da comunicação em detrimento da latência das mensagens. A figura 52 demonstra a variação de capacidade de um canal LoRa em cada tipo de configuração dos recursos citados.

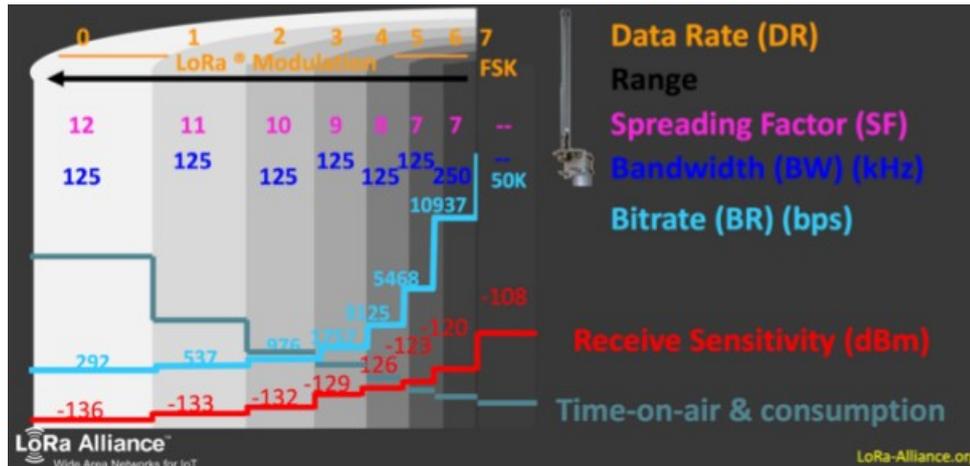


Figura 52 – Capacidade da rede LoRa em função dos parâmetros da transmissão (TEIXEIRA; ALMEIDA, 2017).

Devido ao potencial dessa tecnologia de comunicação sem fio, existem diversas plataformas de *hardware* com o suporte necessário para a sua utilização. O módulo Heltec Wi-Fi LoRa 32, mostrado na figura 53, é projetado utilizando microcontrolador ESP32, com arquitetura RISC de 32 bits, com tecnologia *Ultra Low Power* e utiliza o chip LoRa SX1276/SX1278. Conta também com conexões Bluetooth e Wi-Fi, com as respectivas antenas disponíveis, que podem auxiliar no seu uso. Sobre a plataforma há um *display* OLED de 0,96 polegadas de matriz 128 por 64 que pode ser programado para mostrar diversas informações durante seu funcionamento (PEREIRA; CRUVINEL, 2019). Além disso, seu custo é bem menor se comparado ao módulo XBee, com um valor até 80% abaixo.

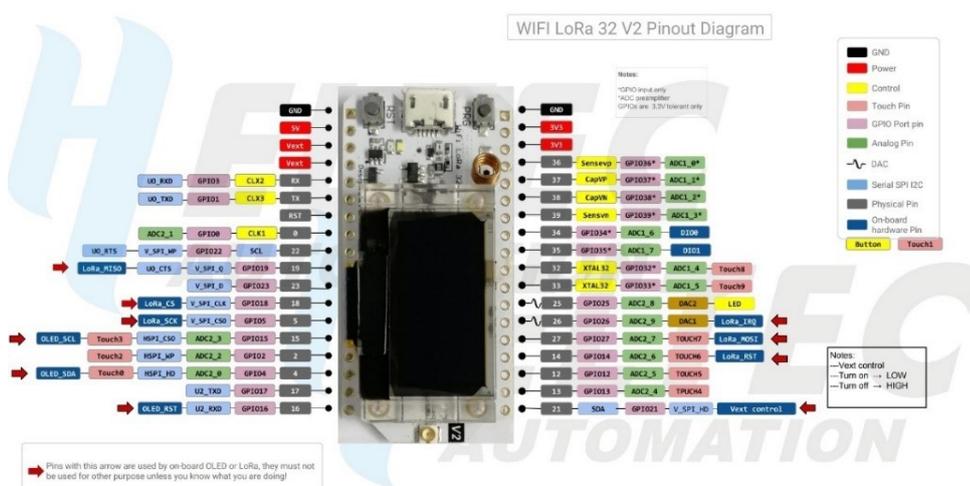


Figura 53 – Módulo Heltec Wi-Fi LoRa 32 e sua pinagem (HELTEC, 2018).

Para garantir uma maior distância de transmissão, foi utilizada junto ao módulo uma antena para a frequência de 925MHz e com ganho de sinal de 5dBi. A escolha de uma antena adequada é essencial para a comunicação a longas distâncias, e ainda mais

importante se tratando de um sistema com baixo consumo de energia para realizar essa função, pois ela direciona o sinal e o amplia para melhorar sua propagação.

Esse módulo, por ser *hardware* ESP32, conta com suporte para programação via IDE Arduino, facilitando seu uso e tendo disponíveis algumas bibliotecas que são distribuídas pela própria fabricante do módulo de forma gratuita para simplificar a sua utilização.

4.2.1 Teste de envio e taxa de dados

O primeiro teste realizado, assim como para o módulo XBee, foi feito para avaliar a distância real que os módulos conseguem alcançar e a taxa de perda de pacotes em alguns pontos desta distância. Foi desenvolvido um código que enviase 50 pacotes de 8 *bytes* a cada 1 segundo com a identificação de cada um desses pacotes, para que após todo o envio, pudesse ser verificado quantos haviam sido recebidos corretamente.

Devido ao grande alcance do módulo LoRa, esse teste foi realizado em duas etapas. A primeira etapa foi realizada em distâncias menores, iniciando em 50 metros e com o ponto mais longe de comunicação estando a 1000m de distância, sem linha de visada direta após 500m, contendo árvores e prédios entre o emissor e o receptor das mensagens, como é mostrado na figura 54.



Figura 54 – Local inicial para os testes utilizando o módulo LoRa.

Devido às características do protocolo LoRaWAN já descritos, e para se definir a melhor configuração para uso no projeto, foram utilizados três diferentes fatores de espalhamento, SF7, SF10 e SF12. Assim, é possível definir em qual valor desse parâmetro a aplicação do projeto se encaixa melhor, dado que esses valores são os limites inferior e superior e mais um valor intermediário, permitindo a verificação do comportamento da tecnologia de forma mais completa em um estudo prático.

Em todos os níveis de espalhamento testados nesta etapa, os resultados obtidos foram satisfatórios, com todos os pacotes sendo recebidos em todos os pontos de teste, mesmo na maior distância de 1km não houveram problemas entre a comunicação dos módulos. Além disso, mesmo com diversos objetos como barreiras no caminho, sejam árvores ou prédios, a comunicação foi totalmente estabelecida, demonstrando a robustez da tecnologia.

A segunda etapa foi realizada com uma distância maior, cobrindo 2km e 3km, em uma área com linha de visada, sem nenhum objeto obstruindo a direção do sinal, a área deste teste é mostrada na figura 55.



Figura 55 – Local utilizado na segunda etapa do teste de distância com o módulo LoRa.

Essa etapa foi realizada para determinação de uma distância máxima para o uso da comunicação LoRa. E mesmo com uma distância grande e o uso de uma antena pequena e com pouco ganho, e da potência máxima do sinal de envio ser baixa (100 mW segundo o fabricante), os resultados continuaram sendo muito promissores, de forma que não foi possível ainda medir a distância máxima com o uso do fator de espalhamento SF12. Como mostra o gráfico da figura 56, para SF7 a partir de 2km já houve perda de uma quantidade significativa de pacotes, onde em média 32 pacotes chegavam ao destino, e a partir de 3km a comunicação era praticamente nula, com apenas 3 pacotes sendo recebidos pelo módulo final. Para o SF10, a comunicação permaneceu constante em 2km, porém em 3km sendo observada a perda de aproximadamente 50% dos pacotes. Por fim, como já apresentado, com fator SF12 não houve problemas de comunicação, sendo recebidos em ambas as distâncias medidas quase que a totalidade das mensagens enviadas, 49 a 50 pacotes em média.

Com os ótimos resultados deste primeiro teste com o módulo LoRa da Heltec, já houve um indicativo pelo seu uso dentro do projeto, pois a transmissão de mensagens estabelecida foi muito além das distâncias que podem ser colocadas entre o veículo e a equipe durante seu uso em testes e nas competições, e mesmo com a configuração do SF7

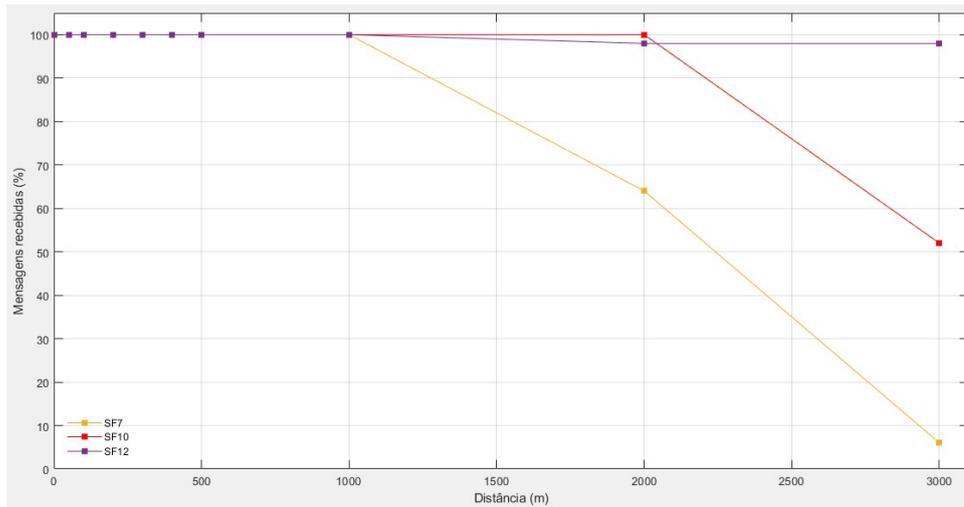


Figura 56 – Desempenho do módulo LoRa no teste de distância.

que apresentou as menores distâncias entre as medições, o resultado foi muito bom, ainda mais considerando sua taxa de transmissão maior que os demais e menor consumo de bateria.

4.2.2 Teste de latência

Para testar a latência da transmissão, ou seja, o tempo em que a mensagem trafega no ar até chegar ao seu destino, foi estabelecido um teste nos mesmos moldes do teste de distância. Foi desenvolvido um código que enviasse 30 pacotes de mensagens, de forma que cada um enviasse um número de identificação e o tempo em que ele foi enviado, para que na sua chegada o módulo que recebe essas mensagens também registrasse o tempo em que cada uma chegou. Assim como no teste de latência do XBee, é importante a sincronia entre os módulos para que os tempos registrados, tanto no envio como no recebimento, fossem precisos e a variação entre eles pudesse ser bem estabelecida, por isso, antes do início dos testes ambos eram resetados ao mesmo tempo em um circuito que interconectava seus pinos de *Reset*, de forma que ao iniciar, eles estivesse com o valor de tempo fixo para o teste e não houvesse fatores externos que alterassem a medição.

A área onde os testes ocorreram foi a mesma vista na figura 54 com distâncias entre 50 e 1000 metros, e da mesma forma também foram utilizados diferentes valores de configuração do fator de espalhamento (novamente SF7, SF10 e SF12), dado que este parâmetro também interfere na latência da comunicação.

Como esperado, foi observado uma diferença considerável de latência entre os três níveis de espalhamento espectral, porém em cada um deles não foi observada uma variação grande com o aumento da distância, na verdade a variação medida foi mínima ficando em torno de poucos milissegundos. O valor médio foi de 42ms para SF7, 291ms para SF10 e 1022ms para SF12.

Como mostrado no gráfico da figura 57 que contém os resultados do teste, o tempo de latência aproximado para cada SF já pode ser observado na mensagem de sincronia, mostrado como a distância zero. Porém, conforme os módulos se afastavam com o aumento da distância, a latência permaneceu com valores muito próximos, tendo inclusive oscilado em valores maiores no início para SF12, o que pode ser considerado uma variação natural dos valores medidos, e que pode ocorrer por características do ambiente onde foi feita a medição. Por ser um lugar urbano, conta com a passagem de carros em determinados momentos, que é um fator que pode ajudar na degradação do sinal.

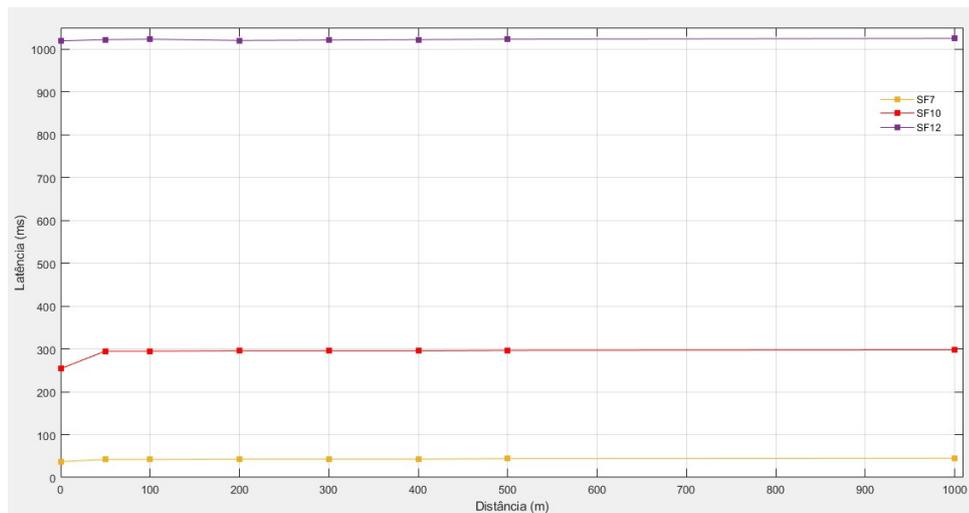


Figura 57 – Desempenho do módulo LoRa no teste de latência.

Assim como no teste de distância, esses resultados se mostraram bastante promissores e não haveria problemas em seu uso, pois mesmo com a latência observada na configuração de SF12, o volume de dados é baixo e não é feita a medição de nenhum item extremamente crítico que necessite de sinalização ou atuação rápida por parte da equipe nos *boxes*. Mas devido a diferença medida, a escolha do SF7 seria a mais apropriada para o funcionamento durante as competições ou testes.

4.2.3 Consumo de energia

O último teste realizado foi do consumo de energia no módulo LoRa e, assim como para o XBee, também foram verificadas as correntes de alimentação deste para os dois modos de funcionamento, nos períodos de ociosidade e na transmissão de dados. Ressalta-se novamente que a medição relativa ao modo ocioso indica o consumo médio do módulo, por esse se tratar do modo de funcionamento durante a maior parte do tempo, com a transmissão gerando pulsos de maior potência. Além disso, como nos outros testes, é importante verificar a diferença nas características com a alteração do fator de espalhamento, pois espera-se que haverá um aumento no consumo, entre os valores máximo e mínimo de configuração (SF7 e SF12).

A biblioteca utilizada fornece a opção de alterar a potência de sinal de transmissão para um valor máximo de 20dBm utilizando uma função *PA boost*, enquanto que a potência padrão de transmissão é de 14dBm. Sendo assim, o teste foi realizado também considerando as potências de envio com e sem a função *PA boost*.

Como esperado, o resultado observado foi que a corrente no modo ocioso permanece constante em todas as configurações de SF e mesmo com *PA boost* ativado os valores de corrente se mantiveram em torno de 50mA. Para a transmissão, com o valor padrão de potência de transmissão, foram medidos os valores de 165mA para todos os fatores de espalhamento SF, indicando que a corrente utilizada para transmitir as mensagens é em torno de 115mA. Com a potência aumentada através do *PA boost*, da mesma forma não houve alteração durante a transmissão entre os diferentes fatores de espalhamento, com valores em torno de 185mA, ou seja, 135mA para a transmissão de mensagens.

A definição da potência padrão de 14dBm para a transmissão dos dados é a melhor escolha, considerando que a distância máxima de envio nos usos padrões do sistema é bem abaixo das distâncias que o módulo alcança com essa potência, e por permitir o aumento da autonomia da bateria, que é um fator muito importante durante as competições.

Porém, tendo em vista que o volume de dados enviados pelo carro é baixo, e considerando que o consumo médio permanece o mesmo com o aumento da potência de envio para 20dBm através do *PA boost*, sua utilização pode ser uma opção também viável em casos em que se encontre interferências no sinal por parte do ambiente ou por outros sinais de radiofrequência, que podem degradar o sinal antes do seu recebimento.

4.3 Comparação dos resultados

Após diferentes testes com os módulos XBee e LoRa, ficou evidente que para a aplicação deste projeto, relacionado à mobilidade, o segundo é a melhor opção de uso, contando com uma robustez muito maior ao se estabelecer uma comunicação, além da facilidade na sua utilização através da biblioteca disponibilizada pela fabricante que permite a criação de *softwares* com IDE e linguagem de programação Arduino, além do menor consumo de energia ao se considerar o tempo ocioso (que equivale a cerca de 99,9% do tempo de funcionamento do módulo nesse projeto) e menor custo.

Com isso, o uso dos módulos XBee juntamente com a antena disponível, se mostrou pouco favorável, pois a uma curta distância, de no máximo 200 metros, já há uma grande instabilidade na comunicação entre os módulos, tendo a necessidade de algum tipo de melhoria para o envio dos dados, como redundância de mensagens e ações de verificação da sua integridade através de dígito verificador, o que significa um maior custo de processamento e a necessidade de mais potência devido ao aumento no tamanho e número de mensagens.



Figura 58 – Pista utilizada para o enduro nas competições.

Os resultados com a tecnologia LoRa foram além do esperado, demonstrando uma contínua estabilidade em grandes distâncias de acima de 1km, sem perda de dados e com um atraso muito pequeno. Para o projeto ele é uma escolha que atende todos os requisitos necessários. Nas competições, as pistas de enduro, onde o uso da telemetria é mais preponderante, têm menos de 300m em seu comprimento máximo, como pode ser visto na figura 58 que demarca a posição onde pode ser instalada a base para o recebimento das informações do carro. Isso permite o uso do LoRa configurado com SF7 para que a latência e o consumo de energia sejam menores. Outro fator importante é que a quantidade de dados enviados será baixa, com mensagens de apenas 8 *bytes* a cada 500 milissegundos, inicialmente.

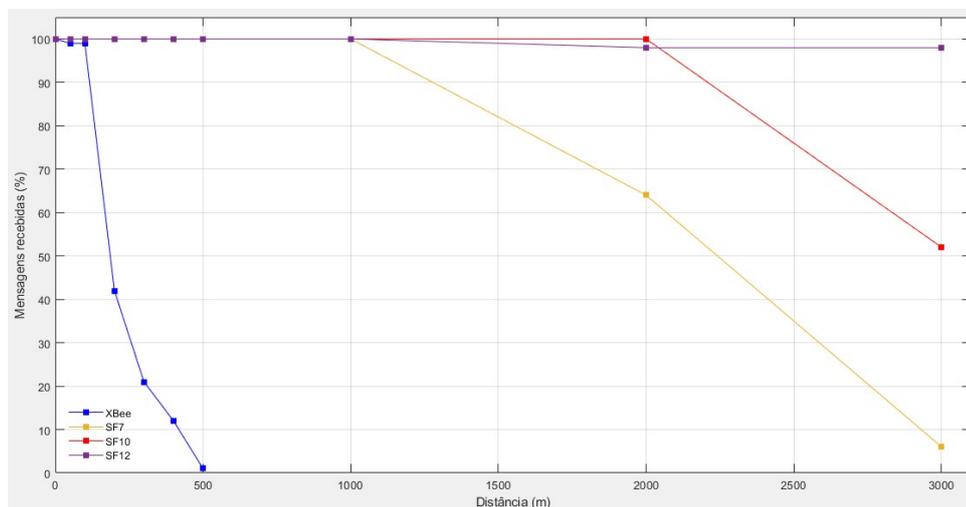


Figura 59 – Teste de distância de envio, XBee vs LoRa (SF7, SF10, SF12).

Para evidenciar a diferença de desempenho observada com os testes práticos de ambos os módulos foram gerados gráficos das curvas em função da distância máxima de transmissão e também da latência medidos, além de uma tabela que engloba todas as características medidas e algumas fornecidas pelo fornecedor em *datasheet* dos módulos.

Nas figuras 59 e 60, com todos os resultados agrupados é notada a diferença entre

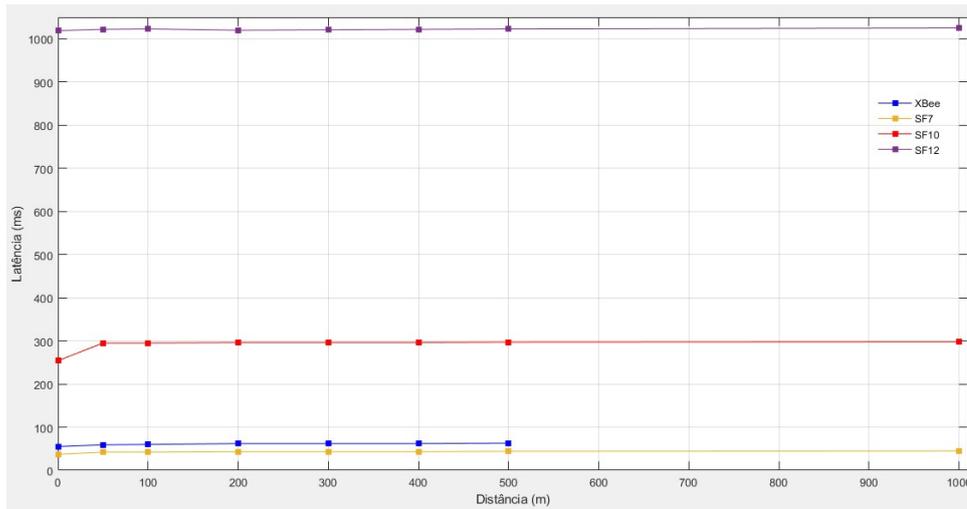


Figura 60 – Teste de latência, XBee vs LoRa (SF7, SF10, SF12).

os módulos, com o XBee chegando a distâncias muito menores que todas configurações SF do LoRa, mas tendo uma latência baixa, próxima da observada para SF7 do LoRa e abaixo das latências de SF10 e SF12. No último parâmetro medido, de consumo dos módulos, observou-se um consumo médio bem próximo ambos os módulos, entre 45 e 50mA, o que é considerada uma baixa potência nessa aplicação e ideal para este projeto.

Tabela 3 – Principais especificações dos módulos testados.

Especificações	Xbee	LoRa
Frequência	2,4 GHz	433/868/915 MHz
Potência de média/transmissão	14 dBm	14 a 20 dBm
Distância de transmissão *	≈ 300 m	> 3000 m
Latência *	60 ms	42 a 1022 ms
Taxa de dados	250 kbps	0,3 a 50 kbps
Consumo de energia *	≈ 45/80 mA	50/185 mA

* Valores medidos nos testes

Por fim, a tabela 3 reúne alguns outros dos principais parâmetros fornecidos pelos fornecedores (DIGI, 2018) (SEMTECH, 2015), além dos medidos neste trabalho, como uma forma de se obter uma visão geral de ambas as tecnologias testadas, a qual foi utilizada também como base para a escolha do módulo LoRa como a melhor opção ao projeto.

5 Interface de Acompanhamento

Com a conclusão dos testes e a escolha de todos os componentes utilizados no projeto, houve a necessidade de desenvolver uma interface que integre os dados coletados no carro com a equipe que os recebe no box. Para isso, foi desenvolvida uma interface simples que receba toda a informação e permita a sua visualização de forma facilitada pelos membros.

Foi definida a utilização da linguagem JavaScript para a criação desta interface, graças à sua facilidade de programação, e ao seu uso crescente para a criação de interfaces gráficas, como por exemplo na usada pela SpaceX no recente lançamento tripulado do seu foguete *Dragon 2* com destino à Estação Espacial Internacional, sendo o primeiro painel *touchscreen* utilizado em tal aplicação (SCHIEMANN, 2020). Essa linguagem conta também com diversos módulos que facilitam o desenvolvimento de diferentes projetos *web*, sendo utilizados como espécies de bibliotecas que expandem o uso dessa tecnologia.

O diagrama da figura 61 exemplifica a criação da interface, em que os dados são coletados no Baja com a utilização do sistema de aquisição de dados, que então os envia através da telemetria com o módulo LoRa. Em seguida, a recepção dos dados é feita por outro módulo LoRa, que transmite através da porta USB para um computador. A partir disso, é criado um servidor *web* que guarda todos os dados e os fornece em uma página *web*, que pode ser acessada por todos os membros da equipe que estiverem conectados na mesma rede, seja por outros computadores ou por celulares.

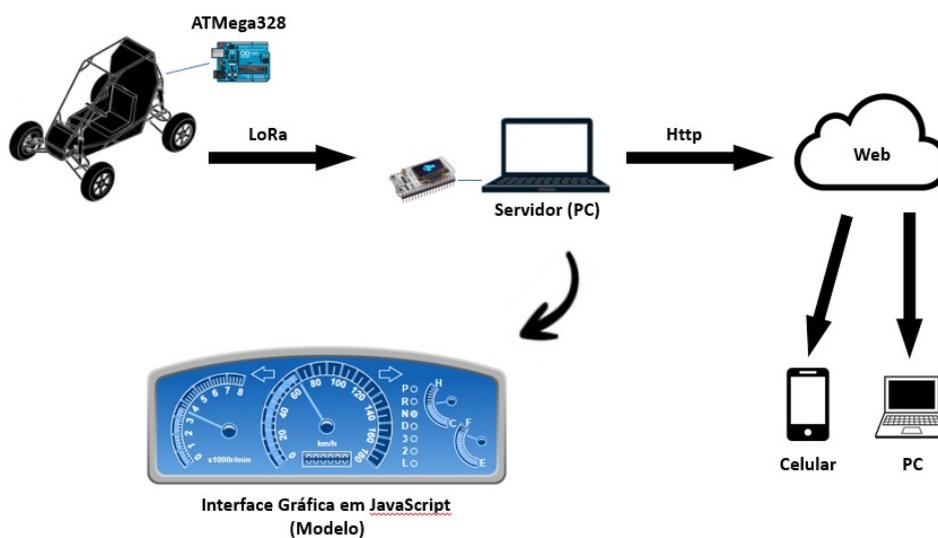


Figura 61 – Diagrama de fluxo dos dados no projeto.

O componente visual para o painel foi fornecido pela biblioteca *Perfect Widgets* (SOFTWARE, 2020), ele pode ser visto na figura 62a, que inclui diversos tipos de mostra-

dores e *sliders* que podem ser usados para diferentes aplicações de usuários. Além disso, essa ferramenta permite a criação de medidores de diferentes formatos, contando com um amplo acervo de elementos que podem ser inseridos e utilizados de forma facilitada em JavaScript.

Foi adicionado também um mapa interativo que utiliza os dados do GPS de latitude e longitude coletados para posicionar e mostrar o deslocamento do carro em tempo real durante a competição. Ele pode ser visto na figura 62b.

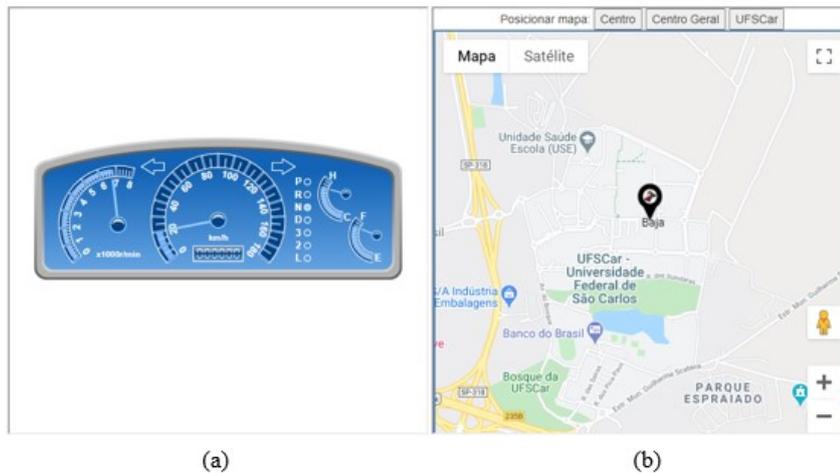


Figura 62 – Interface gráfica desenvolvida para visualizar os dados coletados.

Por estar em um estágio inicial de desenvolvimento, a interface ainda permite mostrar apenas as informações principais, como velocidade do carro, rotação do motor, nível de combustível e temperatura da CVT, porém com a flexibilidade alcançada com a linguagem JavaScript e os módulos disponíveis, esse projeto é facilmente escalável, necessitando de mais algum tempo de projeto para a sua completa implementação, utilizando-se também de gráficos e imagens para disponibilizar todas as informações que já são coletadas do carro. Seu funcionamento ocorreu como o planejado, com todos os instrumentos variando ao receber as informações.

Nota-se também, que por utilizar tecnologias abertas HTML e JavaScript, a visualização do painel, seus instrumentos e mapa, podem ser feitos a partir de qualquer dispositivo com navegador *web*, como celulares, computadores ou *tablets*. Graças a possibilidade de requisições HTTP assíncronas oferecidas pelo par HTML + JavaScript, os dados podem ser visualizados em tempo real.

Com relação ao servidor, sua implementação também é feita em JavaScript, usando a plataforma NodeJS, que permite ler dados da porta serial, e disponibiliza-las através de requisições HTTP. Assim, o servidor *web* local desenvolvido responde com páginas estáticas, que incluem figuras e páginas dinâmicas (que disponibilizam dados recebidos pela porta Serial/USB).

6 Considerações Finais

6.1 Conclusão do projeto

O projeto teve o seu desenvolvimento concluído com êxito, tendo ao final um sistema de aquisição de dados completo e funcional, que permite o acompanhamento e a análise da performance do veículo Baja em tempo real. Essa análise pode ser feita tanto pelo piloto quanto pela equipe no *box*, seja em tempo real durante as competições e testes, ou posteriormente pelo uso dos dados armazenados. Além disso, a implementação de algumas novas tecnologias e componentes que auxiliam a integração do sistema, aumentando a sua confiabilidade e robustez, proporcionou avanços e melhorias em diversos contextos do projeto, como o protocolo de comunicação CAN para o desenvolvimento do *hardware* e o módulo LoRa para a telemetria.

O uso do CAN no projeto é um fator de grande evolução ao sistema, tanto no aspecto teórico, devido ao seu intensivo uso na indústria automobilística, quanto no aspecto prático, por permitir que todos os sistemas de medição que já haviam sido implementados pudessem ser utilizados em conjunto de forma robusta e livre de perdas, além de abrir um caminho para a inclusão de ainda mais sensores ou outros componentes conforme haja a necessidade no futuro.

Apesar de ser notável que todo o projeto se encontrou funcional e com melhores resultados até mesmo em competições nas provas de apresentação de projeto, a evolução natural seria com o uso da telemetria, por essa permitir mais um avanço grande em torno do objetivo de acompanhar a performance do carro. Por isso se deu a busca por uma tecnologia de radiofrequência que estivesse ao alcance da equipe, sendo eficiente em uma aplicação de mobilidade como é o Baja, aliado a um consumo e custo-benefício atrativos. A pesquisa em torno desse tema foi extensa, pois diversos módulos se mostram boas opções, mas poucos atendem realmente aos requisitos necessários, sendo a grande maioria alternativas com custo elevado ou com uma grande complexidade no seu uso, o que dificultou a evolução dessa parte do projeto. A popularização da tecnologia LoRa possibilitou a efetivação da etapa final, se mostrando surpreendentemente eficiente, com uma transmissão de baixa potência que pôde ser estabelecida por milhares de metros mesmo contando com diversos objetos no caminho, como carros, árvores e prédios. Essa característica é bastante importante visto que durante a competição são observados diversos fatores adversos ao uso de telemetria.

A finalização do projeto se deu com a criação de uma interface em JavaScript para a visualização das informações coletadas no *box*. Essa etapa foi essencial para tornar o

sistema ainda mais útil para a equipe e por permitir uma maior interação com o piloto no carro. Seu funcionamento foi verificado e sua eficácia se mostra bastante promissora mesmo em estados iniciais de implementação e sem os testes com ela em competições. Muitas melhorias podem ser adicionadas e diversas opções de *design* podem ser usadas para aumentar a interação com a equipe e outras pessoas.

Por fim, o desenvolvimento dessa monografia cumpre o principal objetivo de criação desse projeto, de ser também uma fonte de conhecimento, se tornando um guia para outros que queiram, através do projeto de extensão universitário, expandir os conhecimentos adquiridos na graduação juntamente de uma carga maior da prática da engenharia elétrica para seu próprio crescimento profissional, e principalmente pessoal.

6.1.1 Sugestões de estudos e trabalhos futuros

Tendo em vista permitir a continuidade e evolução do projeto, além do desenvolvimento de novos estudos nessa área, seguem algumas propostas de melhorias e modificações que podem ser efetuadas a partir deste trabalho:

- Investir na melhoria dos sistemas de medição existentes através da utilização de novos tipos de transdutores, otimizações em *hardware* e *software*, ou até mesmo mudanças na metodologia de medição de algum dos parâmetros apresentados;
- Expansão do sistema com a adição de mais sensores para medir diferentes parâmetros não explorados. Como citado no texto, existem diversas opções que podem ser escolhidas, como pressão nas vias de freio, ângulo de volante, posição de pedais e de amortecedores, temperatura do óleo do motor, entre outros;
- Atualizações de *hardware* e *software*, que podem incluir a utilização de um microcontrolador mais voltado para a aplicação, tornando todo o sistema menos suscetível a falhas;
- Melhora da rotina de telemetria, buscando alterar os tempos entre as mensagens, para que haja um acompanhamento realmente em tempo real, com o envio de cada variável logo após a sua medição. Como observado, já há potencial para esse uso mesmo com as tecnologias de microcontrolador e módulo de rádio utilizados;
- A interface é um item passível de melhoria devido ao seu contato direto com os usuários, por isso é interessante a sua melhoria para incluir mais informações de performance com um *design* eficiente;
- Para aumentar ainda mais a interação com toda a equipe, uma ótima opção seria a criação de um aplicativo de celular que pode receber todas as informações lidas no *box*. Para isso seria necessário o envio de todas as informações (ou pelo menos

das mais importantes) para um servidor *web* que por sua vez faria o *link* dessas informações ao aplicativo;

- Por se tratar de um projeto que permite inovação, há uma boa possibilidade do uso de energias renováveis para o prolongamento da autonomia da bateria. Para isso, existem algumas opções para a geração de energia, como o uso de um pequeno painel solar ou transformando energia térmica – como a que pode ser coletada do escapamento, por exemplo – em energia elétrica, que pode ser usada para o carregamento da bateria, sem a necessidade de retirar ela do veículo para isso ou evitando a sua troca durante a competição.

Referências

ANJOS, E. dos. A evolução da eletrônica embarcada na indústria automobilística brasileira. Centro Universitário do Instituto Mauá de Tecnologia, 2011. Citado na página 28.

ARDUINO, P. H. *MPU-6050 Configuration with Arduino*. 2019. Disponível em: <<https://create.arduino.cc/projecthub/Raushancpr/mpu6050-configuration-with-arduino-1a3dcf>>. Acesso em: 02 Out. 2020. Citado 2 vezes nas páginas 11 e 45.

ARTAL, J. S.; CARABALLO, J.; DUFO, R. Can/lin-bus protocol. implementation of a low-cost serial communication network. *2014 XI Tecnologias Aplicadas a la Ensenanza de la Electronica (Technologies Applied to Electronics Teaching) (TAAE)*, p. 1–8, 2014. Citado 2 vezes nas páginas 11 e 22.

AUTOCORE. *Gravador USB PIC K150*. 2020. Disponível em: <<https://www.autocorerobotica.com.br/gravador-usb-pic-k150>>. Acesso em: 10 Out. 2020. Citado 2 vezes nas páginas 12 e 47.

BOSCH. Can specification version 2.0. *Robert Bosch GmbH, Postfach*, v. 50, 1991. Citado na página 55.

BOSCH. *Electronic engine control unit*. 2020. Disponível em: <<https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/powertrain-systems/gasoline-direct-injection/electronic-engine-control-unit/>>. Acesso em: 20 Oct. 2020. Citado 2 vezes nas páginas 11 e 29.

BROWN, B. *INA125 - Instrumentation Amplifier*. [S.l.], 1998. Disponível em: <<https://www.ti.com/lit/ds/sbos060/sbos060.pdf?ts=1606836643363>>. Acesso em: 20 Set. 2020. Citado 2 vezes nas páginas 11 e 44.

BROWN, G. Discovering the stm32 microcontroller. *Cortex*, Citeseer, v. 3, p. 34, 2012. Citado na página 49.

CAMA-PINTO, A. et al. Received strength signal intensity performance analysis in wireless sensor network using arduino platform and xbee wireless modules. *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, v. 13, n. 7, p. 1550147717722691, 2017. Citado na página 68.

CAMPOS, A. *Explorando a arquitetura do ATmega328: identificando a configuração geral e os fuses*. 2015. Disponível em: <<https://br-arduino.org/2015/05/atmega-arduino-fuses.html>>. Acesso em: 05 Out. 2020. Citado 3 vezes nas páginas 12, 48 e 49.

DIAS, J. E. A. Eletrônica, instrumentação e telemetria do veículo ufvbaja. *UFV, Viçosa-MG*, 2010. Citado 2 vezes nas páginas 11 e 22.

DIGI. *Sending data through an 802.15.4 network latency timing*. 2017. Disponível em: <<https://www.digi.com/support/knowledge-base/>>

- sending-data-through-an-802-15-4-network-latency-t>. Acesso em: 28 Nov. 2020. Citado na página 69.
- DIGI. *XBee/XBee PRO S1 802.15.4 (Legacy) User Guide*. [S.l.], 2018. Citado 3 vezes nas páginas 64, 70 e 80.
- DIGI. *Digi XBee 802.15.4*. 2020. Disponível em: <<https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-802-15-4>>. Acesso em: 15 Nov. 2020. Citado 2 vezes nas páginas 12 e 64.
- DLTEC. *Entendendo o modelo OSI*. 2011. Disponível em: <<http://www.dltec.com.br/blog/cisco/entendendo-o-modelo-osi-para-melhorar-sua-capacidade-de-resolver-problemas-em-uma-rede-cisco/>>. Acesso em: 19 Out. 2020. Citado 2 vezes nas páginas 11 e 32.
- DWIN, B. *DMT32240T035-02W*. [S.l.], 2012. Disponível em: <http://www.victorvision.com.br/images/Datasheets/3.5/DMT32240T035_02W_Datasheet.pdf>. Acesso em: 20 Jan. 2017. Citado 2 vezes nas páginas 12 e 51.
- ENGINEERING, S. *Embedded Flash Scaling Limits*. 2018. Disponível em: <<https://semiengineering.com/embedded-flash-scaling-limits/>>. Acesso em: 16 Out. 2020. Citado 2 vezes nas páginas 11 e 30.
- FEI, B. *A equipe*. 2020. Disponível em: <<https://www.bajafei.com/fei-baja>>. Acesso em: 30 Nov. 2020. Citado na página 22.
- Fleming, B. Microcontroller units in automobiles [automotive electronics]. *IEEE Vehicular Technology Magazine*, v. 6, n. 3, p. 4–8, 2011. Citado na página 21.
- HELTEC. *WiFi LoRa 32 Pin Out Diagram*. 2018. Disponível em: <https://resource.heltec.cn/download/WiFi_LoRa_32/WIFI_LoRa_32_V2.pdf>. Acesso em: 20 Nov. 2020. Citado 2 vezes nas páginas 13 e 73.
- HORVAT, G.; ŠOŠTARIĆ, D.; ŽAGAR, D. Power consumption and rf propagation analysis on zigbee xbee modules for atpc. In: IEEE. *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*. [S.l.], 2012. p. 222–226. Citado na página 70.
- HPL, S. C. Introduction to the controller area network (can). *Application Report SLOA101*, p. 1–17, 2002. Citado 3 vezes nas páginas 12, 55 e 56.
- IBRAHIM, D. *ARM-Based Microcontroller Projects Using Mbed*. [S.l.]: Newnes, 2019. Citado na página 49.
- INSTRUMENTS, T. *LM35 Precision Centigrade Temperature Sensors*. [S.l.], 2017. Citado 2 vezes nas páginas 11 e 39.
- INVENSENSE. *MPU-6000 and MPU-6050 Product Specification*. [S.l.], 2013. Citado na página 43.
- JUNIOR, S. L. M. et al. Stepe-sistema telemétrico de pesagem automotiva. *Revista Ilha Digital*, v. 4, p. 49–59, 2013. Citado na página 65.

- JÚNIOR, V. P. da S. *Conheça a tecnologia LoRa e o protocolo LoRaWAN*. Embarcados, 2016. Disponível em: <<https://www.embarcados.com.br/conheca-tecnologia-lora-e-o-protocolo-lorawan/>>. Acesso em: 30 Out. 2020. Citado 2 vezes nas páginas 12 e 71.
- LOPES, I. M.; BENEVENUTO, F. M.; OLIVEIRA, F. M. de. As camadas do modelo OSI: Revisitando suas funcionalidades e respectivos protocolos. *Revista Interdisciplinar Pensamento Científico*, v. 1, n. 2, 2015. Citado na página 33.
- MARQUES, A. L.; CARRIJO, R. S.; JÚNIOR, R. F. F. Aplicações didáticas em controle e automação utilizando softwares de simulação e sistemas de controle digital embarcados. 2019. Citado 3 vezes nas páginas 12, 49 e 50.
- MARQUES, M. A.; FRANCO, L. R. H. R. *CAN Automotivo Sistema de Monitoramento*. Tese (Doutorado) — Dissertação de Mestrado. Universidade Federal de Itajubá, 2004. Citado 3 vezes nas páginas 12, 21 e 54.
- MICROCHIP. *MCP2551 - High-Speed CAN Transceiver*. [S.l.], 2006. Data Sheet. Citado na página 58.
- MICROCHIP. *PIC18(L)F25/26K83 Datasheet*. [S.l.], 2017. Citado na página 47.
- MICROCHIP. *MCP2515 - Stand-alone CAN Controller with SPI Interface*. [S.l.], 2018. Data Sheet. Citado na página 57.
- MICROCHIP. *PIC MCU Families*. 2020. Disponível em: <<https://www.microchip.com/design-centers/8-bit/pic-mcus/device-selection>>. Acesso em: 05 Out. 2020. Citado 2 vezes nas páginas 11 e 47.
- MICROSYSTEMS, A. *A110x datasheet*. [S.l.], 2016. Citado 2 vezes nas páginas 11 e 38.
- MIRANDA, F. A. M.; CARDIERI, P. Current consumption in radio modules for wireless sensor networks. *Proceedings of the XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais—SBrT2017, São Paulo, Brazil*, p. 3–6, 2017. Citado na página 70.
- NATALE, M. D. et al. *Understanding and using the controller area network communication protocol: theory and practice*. [S.l.]: Springer Science & Business Media, 2012. Citado 3 vezes nas páginas 12, 53 e 54.
- NAVET, N.; SIMONOT-LION, F. *Automotive embedded systems handbook*. [S.l.]: CRC press, 2017. Citado 3 vezes nas páginas 11, 21 e 28.
- NETTO, F. S. et al. Geradores de seqüências pseudoaleatórias usando caos em sistemas de espalhamento espectral. Universidade Presbiteriana Mackenzie, 2009. Citado na página 71.
- ORTIZ, F. M. et al. Caracterização de desempenho de uma rede lora em ambientes urbanos: Simulação vs. prática. In: SBC. *Anais do III Workshop de Computação Urbana*. [S.l.], 2019. p. 167–180. Citado na página 70.
- PAULO, V. Folha de S. *Controle de estabilidade chega às picapes*. 2011. Disponível em: <<http://classificados.folha.uol.com.br/veiculos/1002964-controle-de-estabilidade-chega-as-picapes.shtml>>. Acesso em: 16 Out. 2020. Citado 2 vezes nas páginas 11 e 31.

- PEREIRA, M.; CRUVINEL, P. Desenvolvimento de um sistema de coleta automática dados agrícola baseado em rede lora e no microprocessador esp32. In: SBC. *Os Anais da X Escola Regional de Informática de Mato Grosso*. [S.l.], 2019. p. 43–48. Citado na página 73.
- RYCERSKI, P. et al. Field performance analysis of ieee 802.15. 4 xbee for open field and urban environment applications in smart districts. *Energy Procedia*, Elsevier, v. 122, p. 673–678, 2017. Citado na página 68.
- SAE. *Baja Nacional*. 2020. Disponível em: <<https://saebrasil.org.br/programas-estudantis/baja-sae-brasil/>>. Acesso em: 06 Out. 2020. Citado na página 26.
- SAE. *Regulamento Administrativo e Técnico Baja SAE Brasil*. 2020. Disponível em: <http://saebrasil.org.br/wp-content/uploads/2020/03/RATBSB_emenda_03.pdf>. Acesso em: 09 Out. 2020. Citado 2 vezes nas páginas 27 e 42.
- SANUSB, G. Aplicações práticas de eletrônica e microcontroladores em sistemas computacionais. 2009. Citado na página 47.
- SCHIEMANN, D. *JavaScript reaches the final frontier: Space*. InfoQ, 2020. Disponível em: <<https://www.infoq.com/news/2020/06/javascript-spacex-dragon/>>. Acesso em: 01 Dez. 2020. Citado na página 81.
- SEMTECH. *LoRa® Modulation Basics*. [S.l.], 2015. Disponível em: <<http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf>>. Acesso em: 02 Nov. 2020. Citado 2 vezes nas páginas 72 e 80.
- SERGERS, J. *Analysis techniques for racecar data acquisition*. [S.l.]: SAE, 2014. Citado 2 vezes nas páginas 11 e 34.
- SOFTWARE, P. *Perfect Widgets*. 2020. Disponível em: <<http://perfectwidgets.com/Main>>. Acesso em: 25 Nov. 2020. Citado na página 81.
- SOLUTIONS, T. *Semiconductor test responds to automotive ICs*. 2018. Disponível em: <<https://www.mentor.com/mentor-automotive/blog/2018-03-20-Free-webinar---semiconductor-test-responds-to-automotive-ICs>>. Acesso em: 18 Out. 2020. Citado 2 vezes nas páginas 11 e 31.
- SOUZA, P. V. de. Estudo e elaboração de uma rede can para aplicação em um sistema automotivo. Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG, 2013. Citado 2 vezes nas páginas 12 e 53.
- SÍRIO, A. M. Comunicação de uma rede mesh utilizando o padrão zigbee. Centro Federal de Educação Tecnológica de Minas Gerais, CEFET-MG, 2018. Citado na página 64.
- TEIXEIRA, G. B.; ALMEIDA, J. V. P. d. *Rede LoRa® e protocolo LoRaWAN® aplicados na agricultura de precisão no Brasil*. Dissertação (B.S. thesis), 2017. Citado 3 vezes nas páginas 13, 71 e 73.
- THOMSEN, A. *O que é Arduino?* 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/#:~:text=O%20Arduino%20foi%20criado%20em,a%20estudantes%20e%20projetistas%20amadores.>> Acesso em: 05 Out. 2020. Citado 2 vezes nas páginas 12 e 48.

WEIGHING, D. *How does a load cell work*. 2010. Disponível em: <<http://digitalweighing.blogspot.com.br/2010/08/how-does-load-cell-work.html>>. Acesso em: 20 Nov. 2020. Citado 2 vezes nas páginas 11 e 43.

A Esquemáticos das placas utilizadas

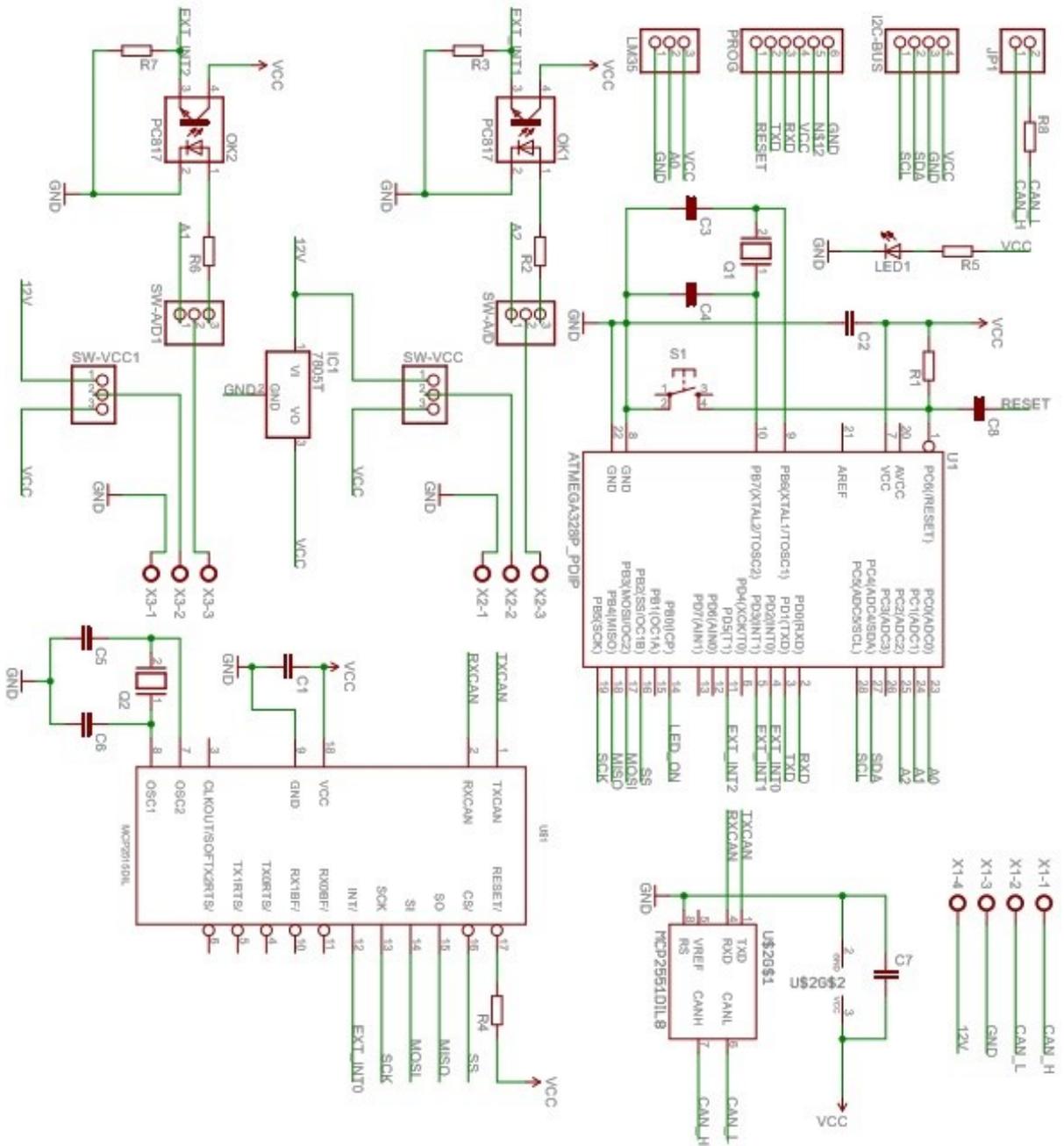


Figura 63 – Esquemático da PCI do módulo de leitura de sensores.

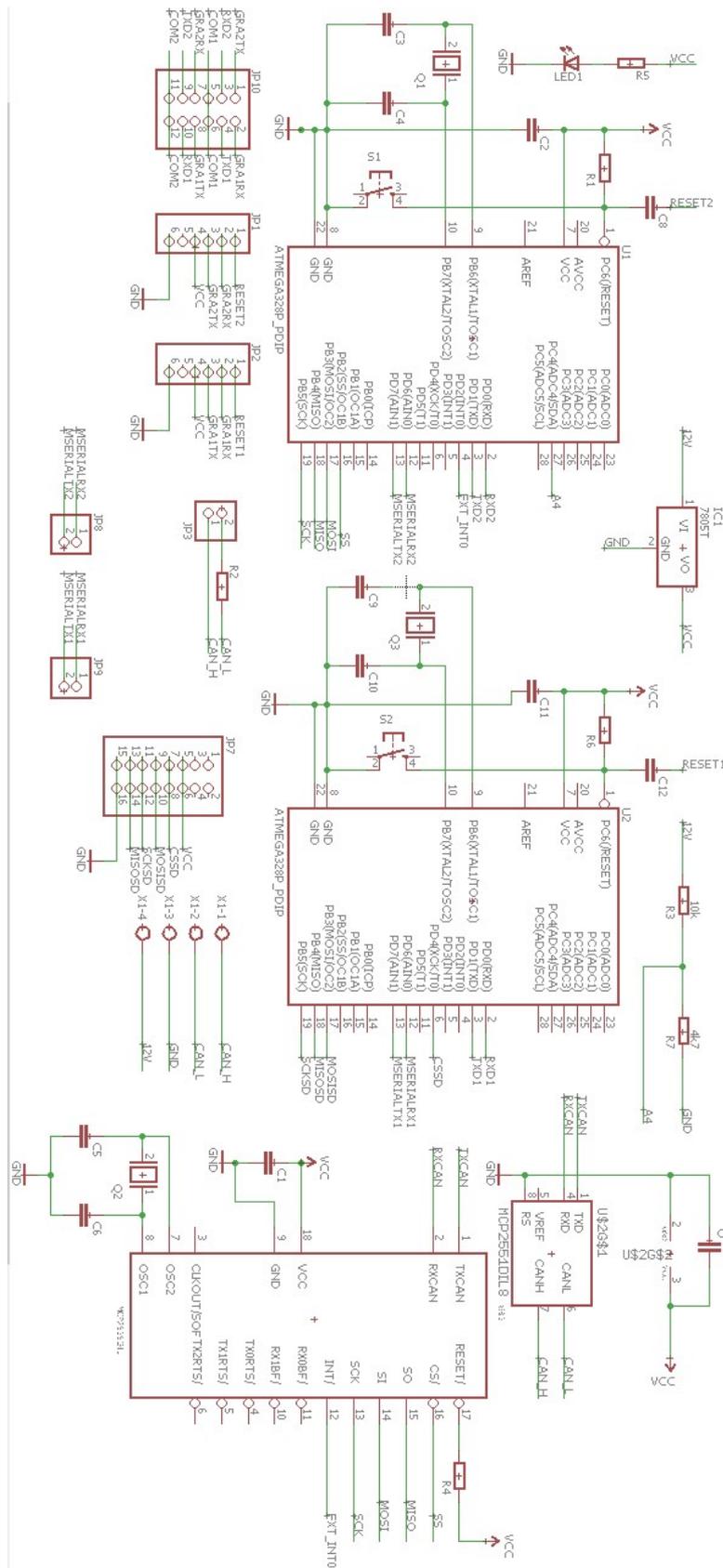


Figura 64 – Esquemático da PCI do módulo de gravação dos dados em cartão SD.

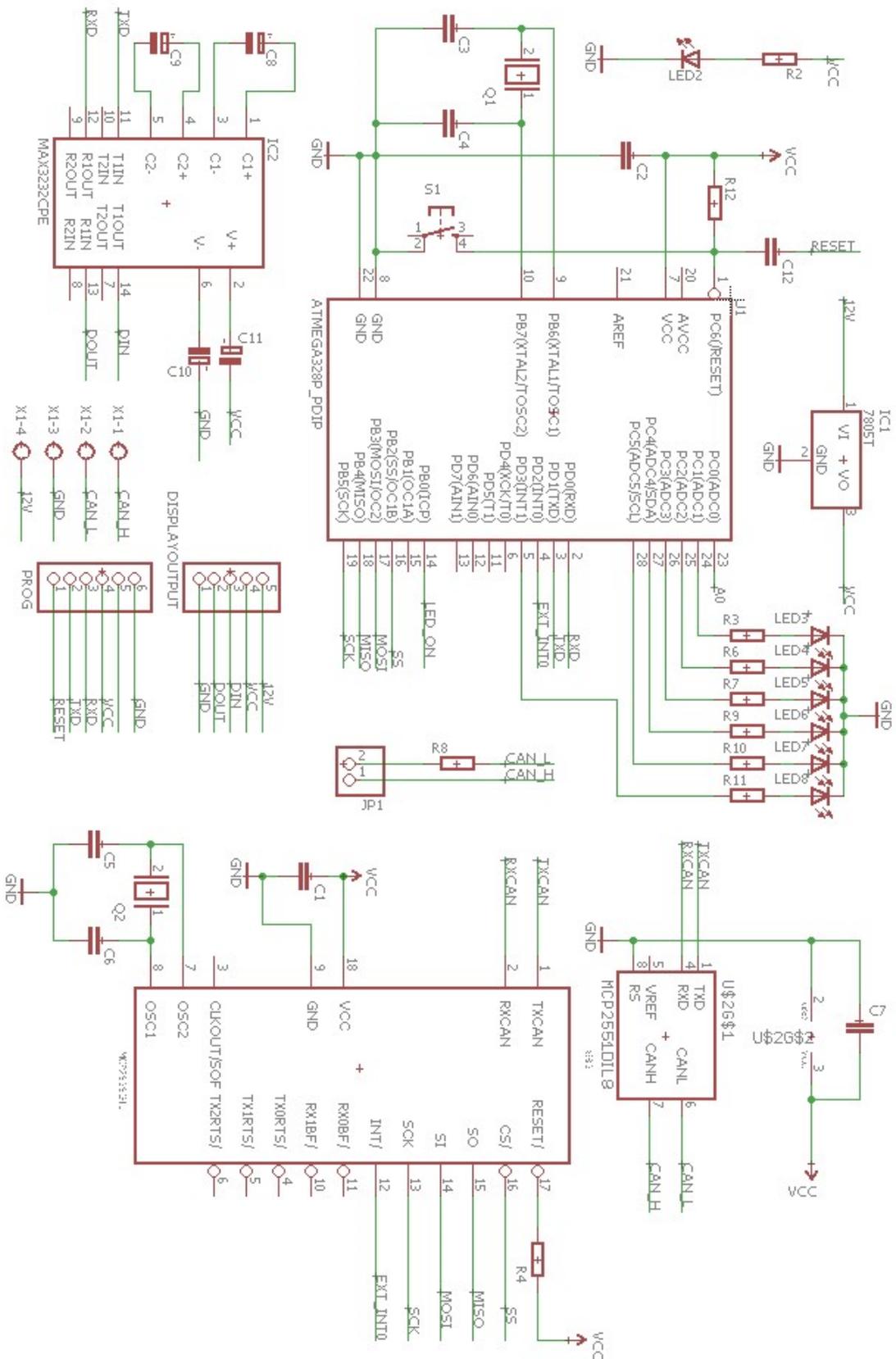


Figura 65 – Esquemático da PCI do módulo do *display* LCD.

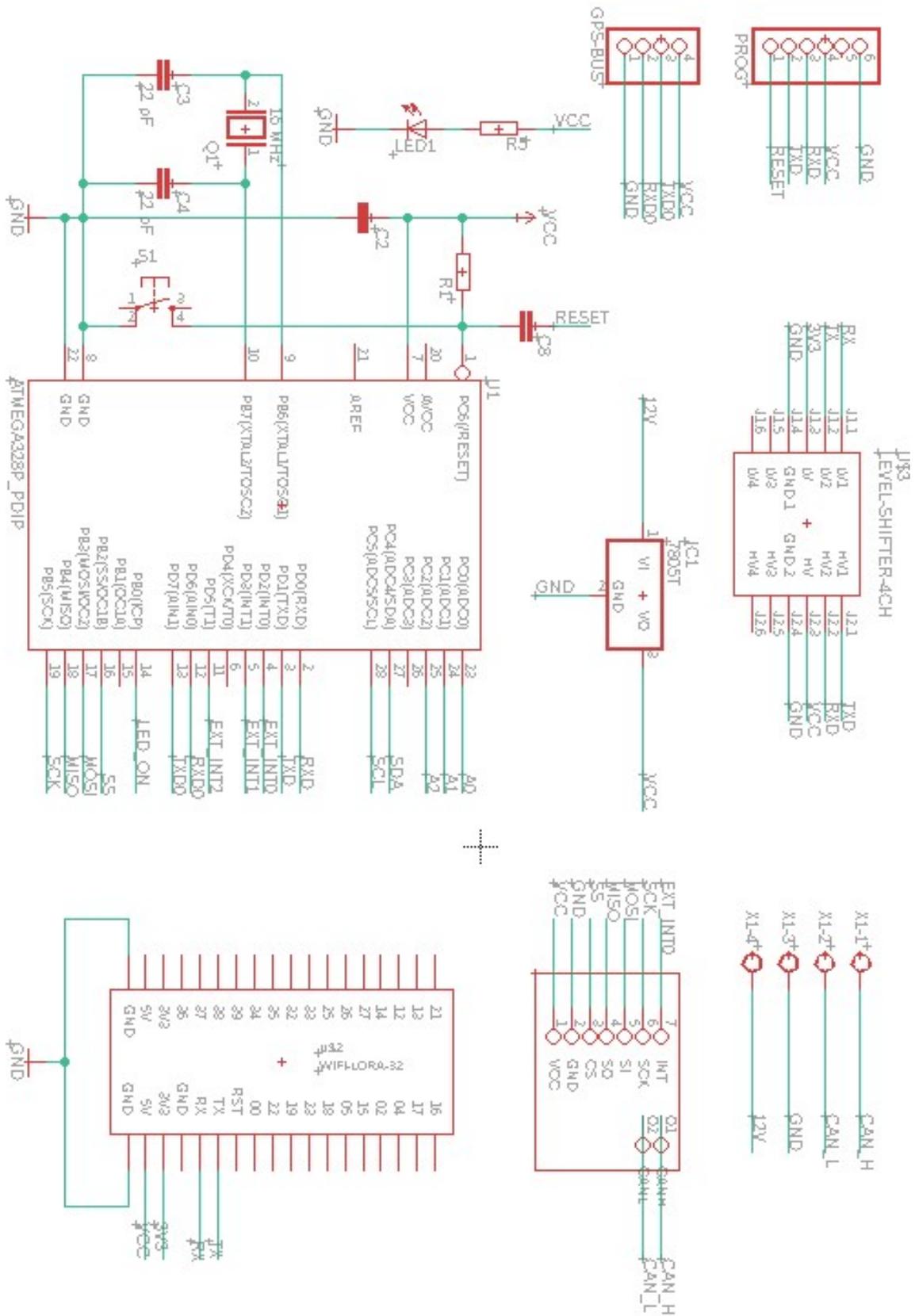


Figura 66 – Esquemático da PCI do módulo de GPS e Telemetria.